

# IT Project

## 결과보고서

프로젝트 명	서울시의 지역적 특성을 감안한 공공자전거 대여소 적정 위치에 대한 방향성 모색
팀 명	1조
문 서 제 목	결과보고서
프로젝트 유형	■ 시스템 프로그램 개발

제출일	2020-02-11
-----	------------

팀원	팀장	최세윤
	팀원	김은비
	팀원	백영서
	팀원	홍석준

## 문서 정보 / 수정 내역

Filename	결과보고서-공공자전거 대여소 적정 위치에 대한 방향성 모색.hwp
원안작성자	
수정작업자	

수정날짜	대표수정자	추가/수정 항목	내 용
2021-01-15	백영서	최초 작성	
2021-01-18	백영서	내용 수정	수정된 연구내용 추가
2021-01-19	백영서	내용 수정	향후 추진 계획 수정
2021-01-27	홍석준	내용 작성	진행된 내용 추가, 정리
2021-02-02	홍석준	내용 수정	내용추가 및 코드, 결과화면 수정
2021-02-10	김은비	내용 추가	진행된 코드, 결과화면 추가
2021-02-10	홍석준	내용 정리	추가된 화면 정리
2021-02-17	홍석준	결과 작성	최종 결론 작성

목 차

1. 프로젝트 목표 ..... 4

1.1 요약 ..... 4

1.2 서론 ..... 4

2. 수행 내용 및 본론 ..... 5

2.1 주제선정 ..... 5

2.2 수행내용 ..... 5

2.3 구현 화면 ..... 6

3. 최종보고서 결론 ..... 25

4. 자료출처 ..... 26

## 1. 프로젝트 목표

통행패턴 특성을 분석하고 친환경 교통수단 활성화에 노력하며 공공자전거 수요 예측

### 1.1 요약

- \* 공공자전거 사용률 증가와 편의성을 위해 수요를 예측하여 신규 대여소 설치 장소의 적절성을 평가
- \* 데이터셋 생성을 위해 필요한 데이터를 수집하는 크롤링 작성
- \* 사용할 데이터 가공
- \* 수집된 데이터를 R을 이용하여 시각화
- \* R을 이용한 워드 클라우드 생성
- \* 데이터를 이용하여 머신러닝 시킨 후, 신규 대여소의 수요를 예측하고 위치가 적절한지 판단

### 1.2 서론

서울시 공공자전거 따릉이의 사용률 증가와 편의성을 위해 수요를 예측하여 수요가 많을 것이라고 예상되는 장소에 신규 대여소를 설치한다.

수집된 데이터 중 필요한 데이터는 대여소별 위도, 경도와 지하철역과의 거리, 주변 유동인구, 주변 공원의 유/무, 대여 횟수 가 있다.

이 데이터들을 이용하여 기계학습을 시키고 신규 대여소 위치와 지하철역과의 거리, 주변 유동인구, 공원 유/무를 입력하여, 수요를 예측하여 적합한 장소인지 판별한다.

## 2. 수행 내용 및 본론

### 2.1 주제선정

서울시 공공자전거 신규 대여소 설치를 위한 수요 예측

### 2.2 수행내용

- \* 필요 데이터를 수집하고 원하는 형식으로 데이터로 정제
  - 자료수집을 위한 크롤링 코드 작성
  - 공원 유무 판단을 위한 코드 작성
  - API를 이용하여 지하철역과 대여소 간의 거리를 구함
  - 유동 인구수 데이터 삽입
- \* 머신러닝을 이용하여 예측 모델 생성 및 예측 페이지 생성
- \* 수집한 데이터를 R을 이용하여 시각화, 워드 클라우드 생성
- \* 데이터를 이용하여 기계학습 시킨 후, 신규 대여소의 수요를 예측하고 적합한 장소인지 판단한다.

### 2.3 구현 화면

#### 2.3.1 사용자 품

1조 신규 대여소 설치 타당성 검증을 위한 서울시 공공 자전거 수요 예측

유동 인구	<input type="text"/>
지하철 역과의 거리	<input type="text"/>
공원의 유무	<input type="radio"/> 有 <input type="radio"/> 無

수요 예측하기

예측되는 따릉이 이용 수요 :

```

from selenium import webdriver
from selenium.webdriver.common.keys import Keys
import time
import pyautogui
# >>>>>!!실행된 크롬을 전체 화면으로 설정해주세요!!<<<<<
def A():
    browser = webdriver.Chrome("./chromedriver.exe")
    browser.get('https://data.seoul.go.kr/')
    data = browser.find_element_by_xpath('//*[@id="baseNav"]/ul/li[1]/a')
    data.click()
    # 검색
    find = browser.find_element_by_xpath('//*[@id="searchKeyword"]')
    find.click()
    find.send_keys('서울시 년도별 유동인구 및 사업체 정보')
    find.send_keys(Keys.ENTER)

    click=browser.find_element_by_xpath('//*[@id="datasetVO"]/div[2]/div/section/div[2]/dl/dt/a/strong')
    click.click()

    more = browser.find_element_by_xpath('//*[@id="listDownload"]/a')
    more.click()
    # 다운로드
    down_0 = browser.find_element_by_xpath('//*[@id="fileTr_8"]/td[6]/a')
    down_0.click()
    down_1 = browser.find_element_by_xpath('//*[@id="fileTr_5"]/td[6]/a')
    down_1.click()
    time.sleep(20)
    browser.close()

def B():
    browser = webdriver.Chrome("./chromedriver.exe")
    browser.get('https://data.seoul.go.kr/')
    # 마우스 포인터 조작
    pyautogui.moveTo(900,200,1)
    pyautogui.moveTo(900,300,0.5)
    pyautogui.moveTo(1050,300,0.5)
    pyautogui.click()
    people = browser.find_element_by_xpath('//*[@ id = "seoulstats2" ] / div[1] / div / button[10]')
    people.click()

    year = browser.find_element_by_xpath('//*[@id="treeMenu"]/li[4]/a/span')
    year.click()

    # 다운로드
    down_2 = browser.find_element_by_xpath('//*[@id="subMenu19"]/div[4]/span[2]/a/img')
    down_2.click()
    time.sleep(10)
    browser.close()

```

다음페이지에 계속

```

def C():
    browser = webdriver.Chrome("./chromedriver.exe")
    browser.get('https://data.seoul.go.kr/')
    # 공공데이터 클릭
    data = browser.find_element_by_xpath('//*[@id="baseNav"]/ul/li[1]/a')
    time.sleep(2)
    data.click()

    # 검색
    find = browser.find_element_by_xpath('//*[@id="searchKeyword"]')
    time.sleep(1)
    find.click()
    find.send_keys('서울특별시 공공자전거 대여이력 정보')
    time.sleep(1)

    find.send_keys(Keys.ENTER)
    time.sleep(1)

    # 페이지 클릭
    click =
    browser.find_element_by_xpath('//*[@id="datasetVO"]/div[2]/div/section/div[2]/dl/dt/a/strong')
    click.click()
    time.sleep(1)

    # 파일 전체보기 클릭
    more = browser.find_element_by_xpath('//*[@id="listDownload"]/a')
    more.click()
    time.sleep(2)
    # 다운로드 24개
    for cnt in range(7,31):
        down_24 = browser.find_element_by_xpath('//*[@id="fileTr_'+str(cnt)+'"]/td[6]/a')
        down_24.click()
        time.sleep(0.5)



























    time.sleep(100)
    browser.close()

print('실행된 크롬을 전체화면으로 설정해주세요.')
A()
time.sleep(1.5)
B()
time.sleep(1.5)
C()
time.sleep(1.5)
print('Wn''다운로드완료')

```

<소스 1> 크롤링을 이용한 데이터 다운로드

[공공자전거 대여소 적정 위치에 대한 방향성 모색]

▼ 오늘 (26)			
 서울특별시 공공자전거 대여정보_201911_2.csv	2021-02-17 오후 3:21	Microsoft Excel ...	90,836KB
 서울특별시 공공자전거 대여정보_201911_1.csv	2021-02-17 오후 3:21	Microsoft Excel ...	106,467KB
 서울특별시 공공자전거 대여정보_201910_3.csv	2021-02-17 오후 3:21	Microsoft Excel ...	99,384KB
 서울특별시 공공자전거 대여정보_201910_2.csv	2021-02-17 오후 3:21	Microsoft Excel ...	107,305KB
 서울특별시 공공자전거 대여정보_201910_1.csv	2021-02-17 오후 3:21	Microsoft Excel ...	106,116KB
 서울특별시 공공자전거 대여정보_201909_3.csv	2021-02-17 오후 3:22	Microsoft Excel ...	54,735KB
 서울특별시 공공자전거 대여정보_201909_2.csv	2021-02-17 오후 3:22	Microsoft Excel ...	107,280KB
 서울특별시 공공자전거 대여정보_201909_1.csv	2021-02-17 오후 3:22	Microsoft Excel ...	106,306KB
 서울특별시 공공자전거 대여정보_201908_3.csv	2021-02-17 오후 3:22	Microsoft Excel ...	57,506KB
 서울특별시 공공자전거 대여정보_201908_2.csv	2021-02-17 오후 3:23	Microsoft Excel ...	107,231KB
 서울특별시 공공자전거 대여정보_201908_1.csv	2021-02-17 오후 3:22	Microsoft Excel ...	106,313KB
 서울특별시 공공자전거 대여정보_201907_3.csv	2021-02-17 오후 3:22	Microsoft Excel ...	64,699KB
 서울특별시 공공자전거 대여정보_201907_2.csv	2021-02-17 오후 3:23	Microsoft Excel ...	107,168KB
 서울특별시 공공자전거 대여정보_201907_1.csv	2021-02-17 오후 3:23	Microsoft Excel ...	106,265KB
 서울특별시 공공자전거 대여정보_201906_3.csv	2021-02-17 오후 3:23	Microsoft Excel ...	82,833KB
 서울특별시 공공자전거 대여정보_201906_2.csv	2021-02-17 오후 3:24	Microsoft Excel ...	107,125KB
 서울특별시 공공자전거 대여정보_201906_1.csv	2021-02-17 오후 3:24	Microsoft Excel ...	106,305KB
 서울특별시 공공자전거 대여정보_201905.csv	2021-02-17 오후 3:25	Microsoft Excel ...	299,293KB
 서울특별시 공공자전거 대여정보_201904.csv	2021-02-17 오후 3:25	Microsoft Excel ...	192,215KB
 서울특별시 공공자전거 대여정보_201903.csv	2021-02-17 오후 3:25	Microsoft Excel ...	127,021KB
 서울특별시 공공자전거 대여정보_201902.csv	2021-02-17 오후 3:24	Microsoft Excel ...	68,097KB
 서울특별시 공공자전거 대여정보_201901.csv	2021-02-17 오후 3:24	Microsoft Excel ...	69,060KB
 공공자전거 대여이력 정보_201912.csv	2021-02-17 오후 3:21	Microsoft Excel ...	113,146KB
 4_유동인구_조사지점정보.zip	2021-02-17 오후 3:19	압축(ZIP) 파일	10,998KB
 3_유동인구_조사지점위치정보_전제_2015.xls	2021-02-17 오후 3:19	Microsoft Excel 9...	6,387KB
 0_유동인구_유동인구기본.zip	2021-02-17 오후 3:19	압축(ZIP) 파일	41,936KB



```
import csv

def check_park():
    f = open('./info.csv', 'r', encoding='utf-8')
    rdr = csv.reader(f)
    output = ""
    for line in rdr:

        if line[1].find('공원') > 0:
            # '공원' 이라는 단어가 없으면 -1 이 출력됨
            save = line[0]+' '+line[1]+' '+line[2]+' '+line[3]+' 1Wn'
            output += save
        elif line[1].find('숲') > 0:

            save = line[0]+' '+line[1]+' '+line[2]+' '+line[3]+' 1Wn'
            output += save
        else:

            save = line[0]+' '+line[1]+' '+line[2]+' '+line[3]+' 0Wn'
            output += save
    print(output)
    with open('./check_park.csv', "w", encoding="utf-8") as f:
        f.write(output)

check_park()
```

<소스 2> 이름에 공원, 숲 유/무 판단

# 1조 IT PROJECT 결과보고서

[공공자전거 대여소 적정 위치에 대한 방향성 모색]

구현 화면

	A	B	C	D	E
1	1695	월계미룡아파트 정문	37.62342	127.0669	0
2	2301	현대고등학교 건너편	37.52407	127.0218	0
3	2302	교보타워 버스정류장(신논현역 3번	37.50558	127.0243	0
4	2303	논현역 7번출구	37.51152	127.0215	0
5	2304	신영 ROYAL PALACE 앞	37.51253	127.0358	0
6	2305	MCM 본사 직영점 앞	37.52064	127.0345	0
7	2306	압구정역 2번 출구 옆	37.52712	127.0287	0
8	2307	압구정 한양 3차 아파트	37.52861	127.0386	0
9	2308	압구정파출소 앞	37.5293	127.0356	0
10	2309	청담역(우리들병원 앞)	37.5189	127.0494	0
11	2310	청담동 맥도날드 옆(위치)	37.52361	127.0402	0
12	2311	학동로 래미안 아파트 앞	37.51777	127.043	0
13	2312	청담역 13번 출구 앞	37.52058	127.0563	0
14	2313	금원빌딩 앞	37.52512	127.0525	0
15	2314	청담나들목입구	37.52128	127.061	0
16	2315	봉은사역 5번출구 옆	37.51425	127.0611	0
17	2316	삼성역 8번출구	37.50958	127.0634	0
18	2317	삼성도로공원	37.51589	127.0662	1
19	2318	삼성중앙역4번출구(문화센터더 리	37.51358	127.055	0
20	2319	포스코4거리 서쪽(수협 삼성동 지점	37.50661	127.0547	0
2065	2548	반포1동 자율방범조소 옆	37.50848	127.0168	0
2066	1571	북서울 꿈의숲 입구	37.62673	127.0388	1
2067	3600	사근빛물펌프장 건너편	37.56119	127.0497	0
2068	3013	서울지방고용노동청 앞	37.56762	126.9872	0
2069	2659	서울친환경유통센터	37.49417	127.1045	0
2070	3588	세종대학교(영실관)	37.55237	127.0735	0
2071	1574	송중초등학교 옆	37.6167	127.0305	0
2072	2746	수명산파크4단지411동 앞	37.55135	126.8249	0
2073	2745	수명초등학교	37.55313	126.8237	0
2074	2550	양재동 꽃시장 건너편	37.46738	127.0415	0
2075	795	양천 해누리타운	37.51542	126.8638	0
2076	794	양천문화회관	37.51684	126.863	0
2077	3587	우성식품 앞	37.54247	127.0942	0
2078	1572	우이산 입구	37.66446	127.0118	0
2079	2547	포드서초서비스센터앞	37.51905	127.0123	0
2080	1573	현대성우아파트 옆	37.61666	127.0221	0
2081	1268	용존토성역 1번출구 옆	37.51729	127.1142	0
2082	1267	올림픽공원 남2문 앞	37.51424	127.1231	1
2083	583	청계천 생태교실 앞	37.56797	127.0469	0

<화면 2> 이름에 공원, 숲 유/무를 1 또는 0으로 출력

```
import pandas as pd
from haversine import haversine

# 따릉이 대여소 CSV 파일 불러오기
bike = pd.read_csv("C:/Users/Administrator/Desktop/location/bike.csv",
                  names=['num', 'ID', 'lat', 'lon'])
bike_id = bike.loc[bike["ID"].str.contains("역", na=False)]
bike_x = bike['lat']
bike_y = bike['lon']

# 전국 지하철역 CSV 파일 불러오기
station = pd.read_csv("C:/Users/Administrator/Desktop/location/station.csv")
station_id = station.loc[station['name'].str.contains("", na=False)]
station_x = station['위도']
station_y = station['경도']

# 위도, 경도 거리 구하기
# 비교x = (위도, 경도) 순으로 할 때, haversine(비교1, 비교2) -> 위도경도 거리계산
lat1 = (bike_x[0], bike_y[0])
lon1 = (station_x[0], station_y[0])

haversine(lat1, lon1, unit='km') # 거리를 km로 환산.
print(haversine)
```

<소스 3> 대여소를 기준으로 위도, 경도를 통해 거리 구하기

	A	B	C	D	E
1		월계미루이	37.623	127.07	0.65
2	2301	현대고등학	37.524	127.02	0.648
3	2302	교보타워 비	37.506	127.02	0.651
4	2303	논현역 7번	37.512	127.02	0.2
5	2304	신영 ROYA	37.513	127.04	0.529
6	2305	MCM 본사	37.521	127.03	0.7561
7	2306	압구정역 2	37.527	127.03	0.8
8	2307	압구정 한양	37.529	127.04	0.166
9	2308	압구정파출	37.529	127.04	0.439
10	2309	청담역(우리	37.519	127.05	0.69
11	2310	청담동 맥도	37.524	127.04	0.432
12	2311	학동로 래미	37.518	127.04	1.5672
13	2312	청담역 13번	37.521	127.06	0.57
14	2313	금원빌딩 모	37.525	127.05	0.696
15	2314	청담나들목	37.521	127.06	0.386
16	2315	봉은사역 5	37.514	127.06	0.2
17	2316	삼성역 8번	37.51	127.06	0.589
18	2317	삼성도로공	37.516	127.07	0.515
19	2318	삼성중앙역	37.514	127.06	0.465
20	2319	포스코4거	37.507	127.05	0.9784
2070	3588	세종대학교	37.55237	127.0735	0.955
2071	1574	송중초등학	37.6167	127.0305	0.366
2072	2746	수명산파크	37.55135	126.8249	0.928
2073	2745	수명초등학	37.55313	126.8237	0.856
2074	2550	양재동 꽃/	37.46738	127.0415	0.316
2075	795	양천 해누리	37.51542	126.8638	0.597
2076	794	양천문화호	37.51684	126.863	0.767
2077	3587	우성식품 S	37.54247	127.0942	0.977
2078	1572	우이산 입	37.66446	127.0118	0.137
2079	2547	포드서초사	37.51905	127.0123	0.968
2080	1573	현대성우O	37.61666	127.0221	0.496
2081	1268	몽촌토성역	37.51729	127.1142	0.1
2082	1267	올림픽공원	37.51424	127.1231	0.656
2083	583	청계천 생	37.56797	127.0469	0.426

<화면 3> 따릉이 대여소를 기준으로 위도, 경도를 통해 거리 출력(km기준)

```
# 파이썬을 이용해 구와 구별 거치대 수 합 뽑아내기
import pandas as pd
import numpy as np

info = pd.read_excel('bikestop.xlsx')

info['대여소ID'] = info['대여소ID'].astype(object)

info.head()

temp_info=info.copy()

temp_info_1=temp_info.groupby(['대여소_구'])['거치대수'].sum()
sum_temp_info=pd.DataFrame(temp_info_1)
sum_temp_info

sum_temp_info.to_csv('C:/Users/Administrator/Desktop/location/gu-bike.csv',
index=True, encoding='cp949')

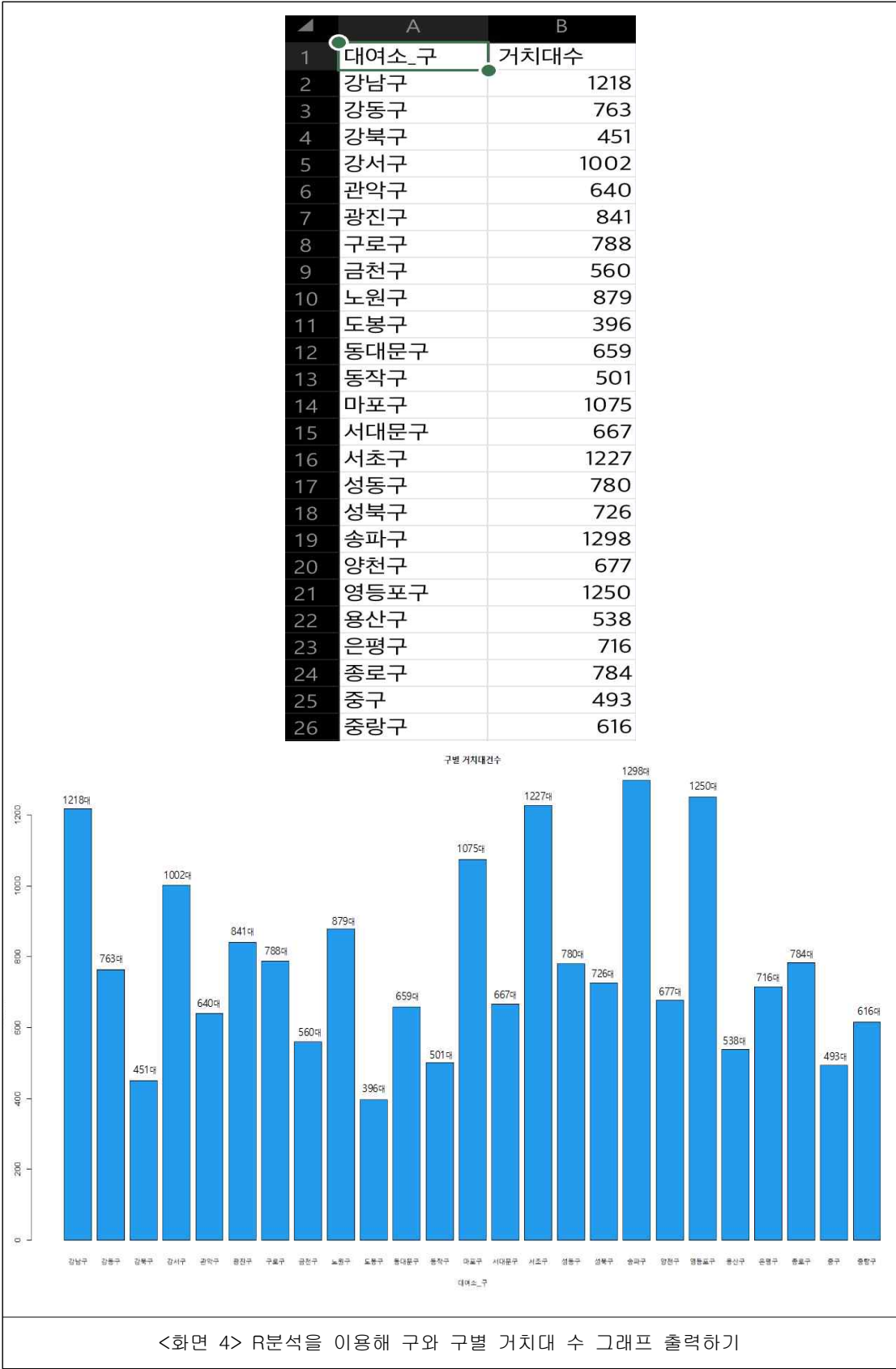
# R studio를 이용해 구별 거치대 수를 나타내는 그래프 출력
useSejongDic()
setwd("c:/r_main")

df=read.csv("구별거치대수.csv")
View(df)

bp=barplot(거치대수~대여소_구,df,col=4)

title(main="구별 거치대건수")
```

<소스 4> R를 이용해 구와 구별 거치대 수 합 그래프 그리기



```
# 파이썬을 이용해 구와 구별 대여건수 합, 반납건수 합 뽑아내기
# 구와 대여건수 합 출력
import pandas as pd
import numpy as np

info = pd.read_excel('rent18.xlsx')

info.head()

temp_info=info.copy()

temp_info_1=temp_info.groupby(['대여소 그룹'])['대여 건수'].sum()
sum_temp_info=pd.DataFrame(temp_info_1)
sum_temp_info

sum_temp_info.to_csv('C:/Users/Administrator/Desktop/location/rent_18.csv', index=True,
encoding='cp949')

# 동일한 방법으로, 구와 반납건수 합 출력
import pandas as pd
import numpy as np

info = pd.read_excel('return18.xlsx')

info.head()

temp_info=info.copy()

temp_info_1=temp_info.groupby(['대여소 그룹'])['반납 건수'].sum()
sum_temp_info=pd.DataFrame(temp_info_1)
sum_temp_info

sum_temp_info.to_csv('C:/Users/Administrator/Desktop/location/return_18.csv',
index=True, encoding='cp949')

# 구와 대여건수 합, 반납건수 합 파일, 한 파일로 합침
```

다음페이지에 계속

```
# R studio를 이용해 구별 대여, 반납 건수를 비교하는 그래프 출력
useSejongDic()
setwd("c:/r_main")

df=read.csv("구별대여반납건수.csv")
View(df)
plot(df[,2]/100,type='o',col=4,ylim=c(0,10000),ann=F,axes=F)
lines(df[,3]/100,type='o',col=3)

axis(1,at=1:25,labels =
c('강남구','강동구','강북구','강서구','관악구','광진구','구로구','금천구','노원구','도봉구','동대문구','동작구','마포구','서대문구','서초구','성동구','성북구','송파구','양천구','영등포구','용산구','은평구','종로구','중구','중랑구'))
axis(2,ylim=c(0,10000))

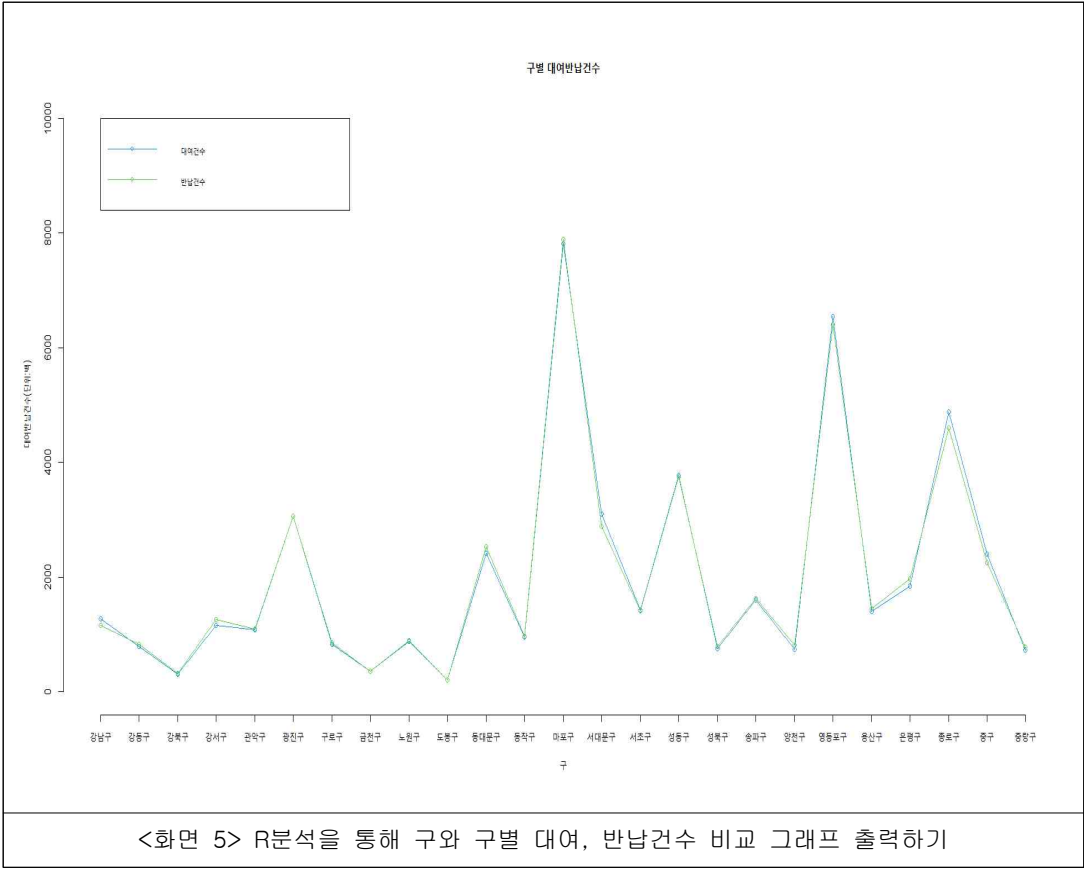
title(main='구별 대여반납건수',xlab='구',ylab='대여반납건수(단위:백)')
legend(1,10000,colnames(df)[-1],col=4:3,lty=1,cex=0.8,pch = 1)
```

<소스 5> R를 이용해 구와 구별 대여, 반납건수 그래프 그리기

	A	B	C
1	구	대여건수	반납건수
2	강남구	127593	115914
3	강동구	78419	83050
4	강북구	31028	32318
5	강서구	115786	125762
6	관악구	108005	108823
7	광진구	306407	305663
8	구로구	82621	84961
9	금천구	35803	35594
10	노원구	88320	89178
11	도봉구	19855	20673
12	동대문구	241653	252585
13	동작구	95234	97082
14	마포구	781594	788652
15	서대문구	310396	288012
16	서초구	141841	140901
17	성동구	377067	374713
18	성북구	74609	79009
19	송파구	160528	162371
20	양천구	74242	81663
21	영등포구	653789	639773
22	용산구	140318	145501
23	은평구	184674	197490
24	종로구	487342	459280
25	중구	240668	224510
26	중랑구	72474	77164

다음페이지에 계속





```

#1. 데이터에서 단어만 추출
setwd("c:/r_main")
d1=readLines("공공자전거.txt")
d1

d2=sapply(d1, extractNoun,USE.NAMES=F)
d2

#2. 단어집합 생성
d3=unlist(d2)
d3

#3. 단어 필터링
d3=Filter(function(x){
  nchar(x)<=10
},d3)

head(unlist(d3),30)

#4. 단어 핸들링
d3=gsub('WWd+', '',d3)      #숫자에 대한 것은 다 날림
d3=gsub('-', '',d3)
d3=gsub('[A-Za-z]', '',d3)
d3=gsub(" ", "",d3)
d3=gsub("(", "",d3)
d3=gsub(")", "",d3)
d3=gsub('[~!=().@#$$%^&*<, %]')
d3=Filter(function(x){nchar(x)>=2},d3)
d3

#5. txt파일로 저장하고 table로 읽어 들어서 공백제거
write((d3), "공공자전거_2.txt")
d4=read.table("공공자전거_2.txt")
d4
nrow(d4)

#6. 단어빈도수 저장
wc=table(d4)
wc

head(sort(wc,decreasing = T),30)
#7. wordcloud 출력
pal=brewer.pal(9,"Set3")
wordcloud(names(wc),freq = wc,scale = c(5,1),rot.per = 0.25,min.freq =
1,random.color = T,random.order = F,color = pal)

```

<소스 6> 워드 클라우드



```

import tensorflow as tf
import numpy as np
import csv

study = np.loadtxt("study.csv", delimiter=",")
study_x = study[:, 0:-1]
study_y = study[:, [-1]]
test1 = []
with open('test1.csv', newline='') as csvfile:
    spamreader = csv.reader(csvfile, quotechar='|')
    for row in spamreader:
        test1.append(row)
        test1 = [list(map(float, x)) for x in test1]

test2 = []
with open('test2.csv', newline='') as csvfile:
    spamreader = csv.reader(csvfile, quotechar='|')
    for row in spamreader:
        test2.append(row)
        test2 = [list(map(float, x)) for x in test2]

tf.model = tf.keras.Sequential()
tf.model.add(tf.keras.layers.Dense(input_dim=3, units=3))
tf.model.add(tf.keras.layers.Dense(input_dim=3, units=1))
tf.model.add(tf.keras.layers.Activation('softmax')) # 활성화 함수 softmax

tf.model.compile(loss='categorical_crossentropy',
optimizer=tf.keras.optimizers.SGD(lr=1e-5), metrics=['accuracy'])
# 모델 설정 loss 함수 categorical_crossentropy 최적화 경사하강법 검증 accuracy
tf.model.summary() # 요약 출력

history = tf.model.fit(study_x, study_y, epochs=2000) # 학습

print("모델 생성")
print("테스트 전용 데이터셋 값 예측 시작")
for i in range(0, 29):
    print("예측된 값: ", tf.model.predict(np.array([[test1[i][0], test1[i][1],
test1[i][2]]])), " 실제 값 : ", test1[i][3])
print("학습된 데이터셋 값 예측 시작")
for i in range(0, 29):
    print("예측된 값: ", tf.model.predict(np.array([[test2[i][0], test2[i][1],
test2[i][2]]])), " 실제 값 : ", test2[i][3])
print("모델 생성")
tf.model.save('model.h5')

```

&lt;소스 7&gt; learning

예측된 값: [[3.0]]	실제 값 : 3.0
예측된 값: [[4.0]]	실제 값 : 5.0
예측된 값: [[4.0]]	실제 값 : 5.0
예측된 값: [[3.0]]	실제 값 : 4.0
예측된 값: [[4.0]]	실제 값 : 3.0
예측된 값: [[3.0]]	실제 값 : 3.0
예측된 값: [[3.0]]	실제 값 : 4.0
예측된 값: [[4.0]]	실제 값 : 3.0
예측된 값: [[3.0]]	실제 값 : 7.0
예측된 값: [[3.0]]	실제 값 : 4.0
예측된 값: [[4.0]]	실제 값 : 3.0
예측된 값: [[3.0]]	실제 값 : 3.0
예측된 값: [[3.0]]	실제 값 : 9.0
예측된 값: [[2.0]]	실제 값 : 2.0
예측된 값: [[3.0]]	실제 값 : 2.0
예측된 값: [[4.0]]	실제 값 : 1.0
예측된 값: [[3.0]]	실제 값 : 0.0
예측된 값: [[3.0]]	실제 값 : 2.0
예측된 값: [[3.0]]	실제 값 : 1.0
예측된 값: [[3.0]]	실제 값 : 1.0
예측된 값: [[4.0]]	실제 값 : 1.0
예측된 값: [[4.0]]	실제 값 : 1.0
예측된 값: [[4.0]]	실제 값 : 2.0
예측된 값: [[4.0]]	실제 값 : 1.0
예측된 값: [[4.0]]	실제 값 : 0.0
예측된 값: [[4.0]]	실제 값 : 11.0
예측된 값: [[4.0]]	실제 값 : 5.0
예측된 값: [[3.0]]	실제 값 : 4.0
예측된 값: [[4.0]]	실제 값 : 6.0
<화면 7> learning	

```

import webbrowser
import tensorflow as tf
import numpy as np
from flask import Flask, request
app = Flask(__name__)
model = tf.keras.models.load_model('save1.h5')
style = """<style>
    .footer{background-color:#D5D5D5;
        text-align:center;
        position:absolute;
        bottom:0;
        width:98%;
        }
    table{
        text-align:center;
    }</style>"""

def mkhtml(predict=""):
    html = """<!DOCTYPE html>
<html><head>
<meta charset="utf-8">
<title>따릉이 수요 예측</title>
{0}
</head><body>
    <form action="/" method="post">
        <center>
            <h2>1조 신규 대여소 설치 타당성 검증을 위한 서울시 공공 자전거 수요
            예측</h2><br>
            <table border="2" bordercolor="white" width="500px">
                <tr height="30">
                    <td>유동 인구</td>
                    <td><input type="number" name = "move" required></td>
                </tr>
                <tr height="40">
                    <td>지하철 역과의 거리</td>
                    <td><input type='number' name = "subway_dis" required></td>
                </tr>
                <tr height="40">
                    <td>공원의 유무</td>
                    <td><input type="radio" value="1" name = "park" required>有
                    &nbsp;
                    <input type="radio" value="0" name = "park" required>無</td>
                </tr>
            </table>
        <br>
    """

```

```

        <input type="submit" value='수요 예측하기'></center> </form>

        <center>
            <br> 예측되는 따릉이 이용 수요 : {1}
        </center>

    <center>
    <footer class="footer">
    <p>광운대학교 국가기관전략훈련 빅데이터분석 및 구축을 위한 파이썬머신러닝과정</p>
    <p>최세윤, 김은비, 백영서, 홍석준</p>
    </footer></center></body></html>"".format(style, predict)

    return html

@app.route('/', methods=['GET', 'POST'])
def index():
    if request.method == 'POST':
        # 파라미터를 전달 받습니다.
        move_temp = float(request.form['move'])
        subwayDis_temp = float(request.form['subway_dis'])
        park_temp = float(request.form['park'])
        y = model.predict(np.array([[move_temp, subwayDis_temp, park_temp]]))
        html = mkhtml(str(y[0][0]))
    else:
        html = mkhtml()
    return html

if __name__ == '__main__':
    webbrowser.open_new("http://127.0.0.1:5000/")
    app.run(debug=True)
    # todo-seyun 완성시 debug=False으로 변경 필요(창이 여러개 켜짐 방지)

```

<소스 8> predict

1조 IT PROJECT 결과보고서

[공공자전거 대여소 적정 위치에 대한 방향성 모색]

1조 신규 대여소 설치 타당성 검증을 위한 서울시 공공 자전거 수요 예측

유동 인구	123456
지하철 역과의 거리	100
공원의 유무	<input checked="" type="radio"/> 有 <input type="radio"/> 無

수요 예측하기

예측되는 따릉이 이용 수요 : 21000 이상 대여 할 것으로 예측됩니다

광운대학교 국가기관전략훈련 빅데이터분석 및 구축을 위한 파이썬머신러닝과정  
최세윤, 김은비, 백영서, 홍석준

1조 신규 대여소 설치 타당성 검증을 위한 서울시 공공 자전거 수요 예측

유동 인구	1234567
지하철 역과의 거리	100
공원의 유무	<input checked="" type="radio"/> 有 <input type="radio"/> 無

수요 예측하기

예측되는 따릉이 이용 수요 : 208000 이상 대여 할 것으로 예측됩니다

광운대학교 국가기관전략훈련 빅데이터분석 및 구축을 위한 파이썬머신러닝과정  
최세윤, 김은비, 백영서, 홍석준

<화면 8> predict



### 3. 최종보고서 결론

공공 자전거의 수요는 주변의 공원과 숲의 유/무 와 지하철역과의 거리(m), 유동인구 수(1주일)의 따라 예측된다.

유동인구 수와 지하철역과의 거리를 입력하고 공원, 숲의 유/무를 체크하면 예측된 값(1년 대여+반납)이 나온다.

공공자전거를 이용하는 사람들이 매년 증가하고 있고, 그만큼 새로운 공공자전거의 대여소의 필요성이 증가하고 있다.

서울 이외에 순차적으로 인구가 많은 지역에 생길 수 있는 가능성이 있다고 생각된다.

그러므로 공공자전거 수요 예측 프로그램을 통해서 신설하려는 공공자전거 대여소의 수요 예측된 값을 제공하여 판단을 효율적으로 할 수 있도록 기대하는 바이다.

## 4. 자료출처

자료 수집 기준 논문

<https://www.dbpia.co.kr/pdf/pdfView.do?nodeId=NODE07114713&mark=0&useDate=&bookmarkCnt=1&ipRange=N&accessgl=Y&language=ko>

자료 수집

<http://data.seoul.go.kr>