

# FINAL REPORT

Team 16



이현재, 주석천, 민성재, 이상민

## Table of Contents

---

Table of Contents .....	1
1. A description of project .....	2
1.1 Purpose.....	2
1.2 Need.....	2
2. Related technology, service, opensource software.....	2
2.1 Examples of other open sources.....	2
2.2 Difference.....	3
1. QGIS.....	4
.....	4
1.1 Feature .....	4
1.2 Advantage .....	4
1.3 Disadvantage.....	4
2. KNIME .....	4
2.1 Feature .....	5
2.2 Advantage .....	6
2.3 Disadvantage.....	6
3 Difference .....	6
3. Vamap .....	7
1. Team, position.....	7
2. Architecture .....	7
3. Usage scenario .....	8
4. Use case .....	10
5. Implementation .....	15
1. Parser. ....	15
2. Dataproc.....	17
3. Scoring. ....	17
4. Roadmanager .....	18
6. Implementation after MVP .....	19
1. Hyunjae Lee. ....	19
2. Sangmin Lee .....	20
3. Seongjae Min. ....	22
4. Seokcheon Ju .....	24
4. Demo .....	26
5. Cooperation with Github .....	30
5.1. Commit, contributors .....	30
5.2. Other issues, Pull request.....	32

## 1. A description of project

---

### 1.1 Purpose

---

현실에 존재하는 수많은 종류의 데이터는 각각의 다양한 기준을 가지는데, 그 중에서 가장 보편적인 것이 **위치**이다. 많은 데이터 분석관련 연구나 어플리케이션에서 위치 기반 데이터를 분석하고 재가공하여 더 가치가 있는 데이터를 얻으려 한다. 기존의 위치 데이터 분석 프로그램들은 지도상에서 단순 특정데이터의 밀집도 혹은 분포도만 보여주는 경우가 많다.

우리의 개발목표는 사용자가 원하는 위치기반 (위도 및 경도) 데이터를 입력하면 해당 데이터와 연관성이 있는 주변 데이터 (도로, 빌딩 등) 와 함께 분석하여 각 위치기반 데이터 객체마다의 상대적 가치를 계산하고 시각화 해주는 것을 목표로 한다.

물론, 특정 상점의 위치기반 데이터가 이 프로그램을 통해 얻어지는 결과 상관분석이라고 판단이 될 수도 있지만, 이 프로그램의 또 다른 기대효과는 공공시설이나 자재 등의 위치를 통해 각 시설 및 자재 등이 얼마나 낭비되고 있는지 혹은 필요하다고 예상되는 곳에 없는지 등의 정보를 제공하는 것이다.

### 1.2 Need

---

많은 지역 기반 데이터가 있지만 입지, 특징 분석을 위해서는 사실상 사용자가 직접 분석을 해야 하는 경우가 많다. 지도의 위치만으로 보았을 때, 데이터의 분포가 어느 지역에서 어떻게 표현되는지는 단순히 데이터의 밀집도의 형태로 어느 동, 구에서 어느정도 집중되어 있다 라는 것만 알 수 있다.

하지만 우리가 데이터를 분석할 때는 단순 집중도가 중요한 것이 아니라 주변 다른 요소들과 어떻게 상호작용을 하여 밀집이 되어있는지에 대한 인사이트를 얻고자 하기 때문이다. 예를 들자면 어떠한 건물의 가치는 건물들의 밀집도만으로 알 수 있는 것이 아니라, 주변의 도로와 같은 부수적인 요소가 중요하게 작용한다. 그렇기 때문에 우리는 도로, 건물 등과 같은 주변 데이터를 사용자가 입력하게끔 하고, 이 주변 데이터로 가치를 매길 유저데이터를 사용자가 입력하게끔 하여 이 둘의 상관관계를 분석하려 한다. 데이터 간의 상관관계를 분석하여 유저가 응용할 수 있도록 시각화, 정량화를 해주는 것으로 목표로 하고 있다.

서울시 공공데이터를 살펴보면, 대부분 .xls 파일의 형태이고 그 데이터들 또한 정규적이지 못하다. 단순히 정리만 해 놓은 형태기 때문에 data set 간의 공통된 규칙이 존재하는 것이 아니기에 실 사용과는 상대적으로 거리가 있다. 그렇기에 본 프로젝트는 대표적으로 쓰일 수 있는 데이터셋을 제공하고, 개발자가 쉽게 사용할 수 있는 형태로 정규화 하는 것을 다음 목표로 한다. 즉 우리는 사용자가 어떠한 데이터의 상대적 분포와 가치를 알고 싶을 때 유용하게 사용할 수 있는 툴을 개발할 것이다.

이 프로그램은 오픈소스 커뮤니티에 기본이 되는 데이터 분석 툴을 제공하는 것이 목표이다. 기본이라 하면, 필요한 핵심기능이 있고 개발의 확장성이 높게끔 함수 설계와 재활용성을 고려하였고, 나아가 사용자의 입맛에 맞게 서비스로 활용할 수도 있으며 더 좋은 아이디어의 영감이 된다는 것을 정의로 한다.

## 2. Related technology, service, opensource software

---

### 2.1 Examples of other open sources

---

	<u>QGIS</u>	<u>WEKA</u>	<u>Rapidminer</u>	<u>Knime</u>	<u>VAMAP</u>
Data Preprocessing	O	O	O	O	O
Data Visualization	O	O	O	O	O
Provide Map	O	X	O	O	O
Open Source	O	O	O	O	O
Deal with Excel	O	O	O	O	O
User data Input	O	O	O	O	O
Data Relationship	X	X	X	O	O
Calculate Data Value with Weight	X	X	X	X	O
Data Form designation	X	X	X	X	O

그림 1 유사한 오픈소스들과의 기능 비교

- Data Preprocessing : 입력 받은 데이터를 프로그램이 원하는 형식으로 전처리 하는 기능
- Data Visualization : 입력 받은 데이터를 시각화 하여 보여주는 기능
- Provide Map : 지도를 통해 시각화를 보여주는 기능
- Open Source : 이 프로그램이 오픈소스인가?
- Deal with Excel : 입력 데이터로 엑셀파일이 다뤄지는가?
- User data Input : 사용자가 원하는 데이터를 입력할 수 있는가?
- Data Relationship : 입력한 데이터가 다른 데이터와 어떠한 관계를 가지는지를 보여주는 기능
- Calculate Data Value with Weight : 데이터에 사용자가 원하는 가중치를 줄 수 있는 기능
- Data Form designation : Setup.txt 를 통해 사용자가 입력하는 파일의 형식을 지정해주는 기능

VAMAP 의 특징인 유저의 데이터를 input 으로 하여 전처리를 하고 지도상에 시각화를 하고 데이터의 가치를 보여주는 것과 유사한 오픈소스 프로그램들을 찾고 각각의 프로그램들의 기능을 정리한 도표이다.

대부분의 오픈소스 프로젝트가 데이터 전처리, 시각화, 엑셀파일을 input 으로 쓰는 등의 기능은 제공하고 있다.

하지만 입력한 데이터가 기존 데이터와의 어떤 관계를 갖는지, 이 데이터가 어떠한 가치를 가지는지, 그리고 사용자가 입력한 엑셀과 Setup.txt 의 형식이 유효한지 체크해주는 기능은 없었다.

즉, 자신이 가진 데이터를 전처리, 시각화, 가치 환산을 하고 싶은 사용자라면 VAMAP 을 사용하는 것이 가장 좋은 선택이 될 것이다.

Vamap 과 가장 유사한 기능들을 제공하는 소프트웨어 중에서 QGIS 와 KNIME 를 직접 사용해보고 자세한 차이점들을 아래에서 살펴본다.

## 2.2 Difference

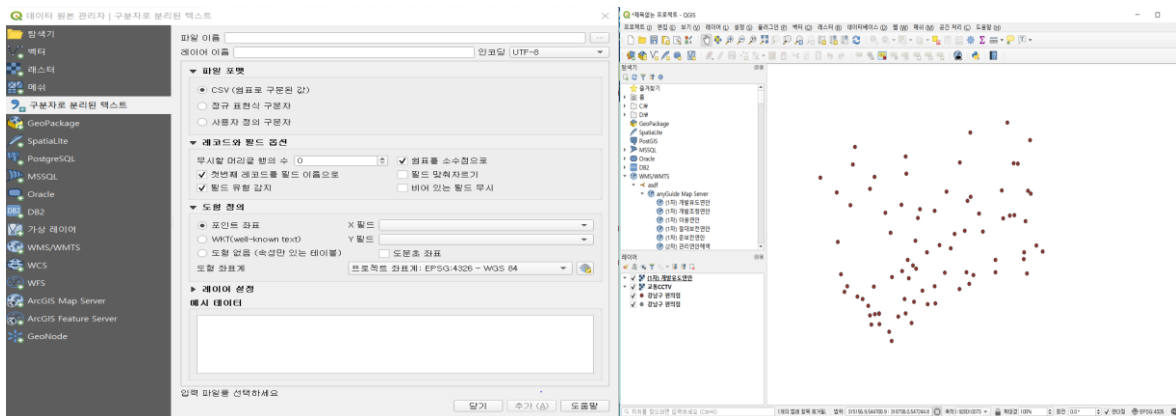
## 1. QGIS

## QGIS - Map Data Visualization



## 1.1 Feature

.csv 파일을 추가하거나 다른 데이터베이스의 테이블을 가져오는 것이 가능하다. 이 파일이나 테이블속에서 좌표를 의미하는 X 필드, Y 필드를 설정해주면 프로그램상에 각 좌표가 찍히게 된다.



다만 바로 지도위에 각 객체들이 찍히는 것이 아니고 지도를 따로 가져와야 한다. 지도를 가져오는 방법은 API 를 가져오거나 다른 사람이 등록한 플러그인에 있는 파일을 추가하는 방식이다. 또한 좌표를 기준으로 찍힌 객체, 지도, 행정구역 경계등이 이 프로그램에서는 각각 '레이어' 라고 불리는데 이 레이어들을 한꺼번에 보고싶다면 동기화하는 과정도 필요하다.

QGIS 또한 오픈소스이기 때문에 많은 유저들이 등록해 놓은 플러그인을 가져다 쓸 수 있다는 장점이 있지만 애초에 프로그램 자체에서 데이터 객체를 지도상에 띄우는 것이 굉장히 번거롭다. 또한 VAMAP의 차별성이라고 할 수 있는 시각화뿐만이 아닌 각 객체가 가지는 가치를 계산하는 기능은 QGIS에 있지 않다.

## 1.2 Advantage

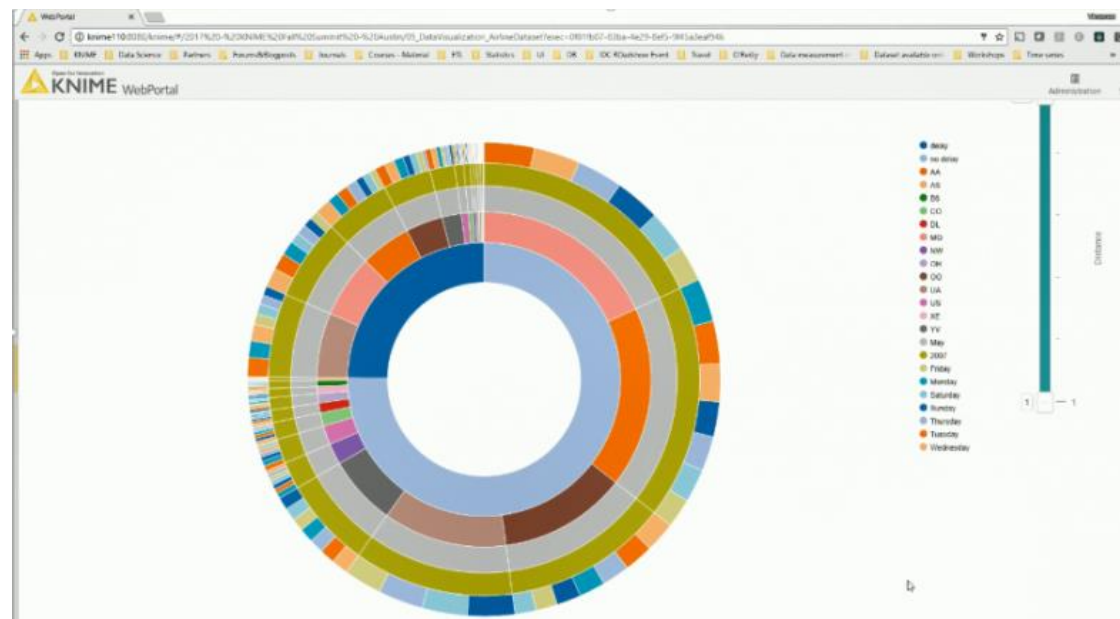
GUI가 구현된 프로그램에서 파일을 추가하는 과정이 간편하며 데이터베이스도 추가 가능한 오픈소스이기에 다른 유저들이 만든 플러그인을 사용하는 것이 가능하다.

### 1.3 Disadvantage

데이터의 객체들을 지도위에 띄우는 과정이 데이터 객체 띄우기와 지도를 띄우는 각각의 과정으로 되어있음. 또한 이를 한꺼번에 보려면 동기화하는 과정도 필요하여 VAMAP과 비교하여 굉장히 번거롭다.

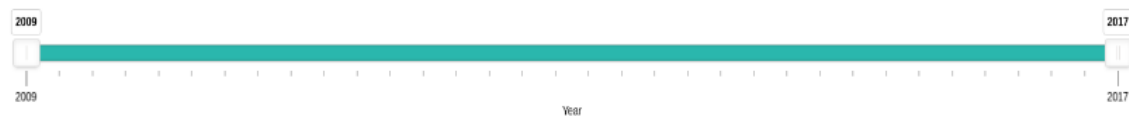
또한 여러가지 다른 기능도 제공하지만 따로 공부를 하거나 알아보지 않으면 사용하거나 활용하기가 너무 어렵다.

## 2. KNIME



A stacked area chart showing the number of pickups over time from 2009 to 2017. The y-axis is labeled 'Pickups' and ranges from 0.0 to 30M in increments of 5.0M. The x-axis is labeled 'Year' and shows dates from 2009-1 to 2017-9. The chart features three data series: Yellow (the bottom layer), Green (the middle layer), and FHV (the top layer, starting around 2014). The total pickups show a significant upward trend, particularly after 2014, with a major peak in late 2016/early 2017 reaching nearly 30M.

Year	Yellow (M)	Green (M)	FHV (M)	Total (M)
2009-1	14.0	0.0	0.0	14.0
2010-6	14.0	0.0	0.0	14.0
2012-2	14.0	0.0	0.0	14.0
2014-1	13.0	1.0	0.0	14.0
2015-6	11.0	2.0	1.0	14.0
2017-9	9.0	1.0	16.0	26.0



기본적으로 workflow 형태로 node 와 edge 를 포함하여 프로젝트를 구성한다. .csv 파일과 데이터베이스 테이블, .arff 파일 등 여러가지 파일이 load 가능하다. 시각화에 특화된 프로그램 답게 다양한 시각화 방식을 제공한다. 여기에는 맵 위에서 특정 데이터에 대한 분포 등을 시각화해주는 것도 존재하는데, 이 데이터들 간의 상관관계나 특정 데이터의 가치 등을 계산하는 방식은 존재하지 않는다.

## 2.2 Advantage

---

다양한 파일을 load 할 수 있고 다양한 시각화 방식을 확인해볼 수 있다

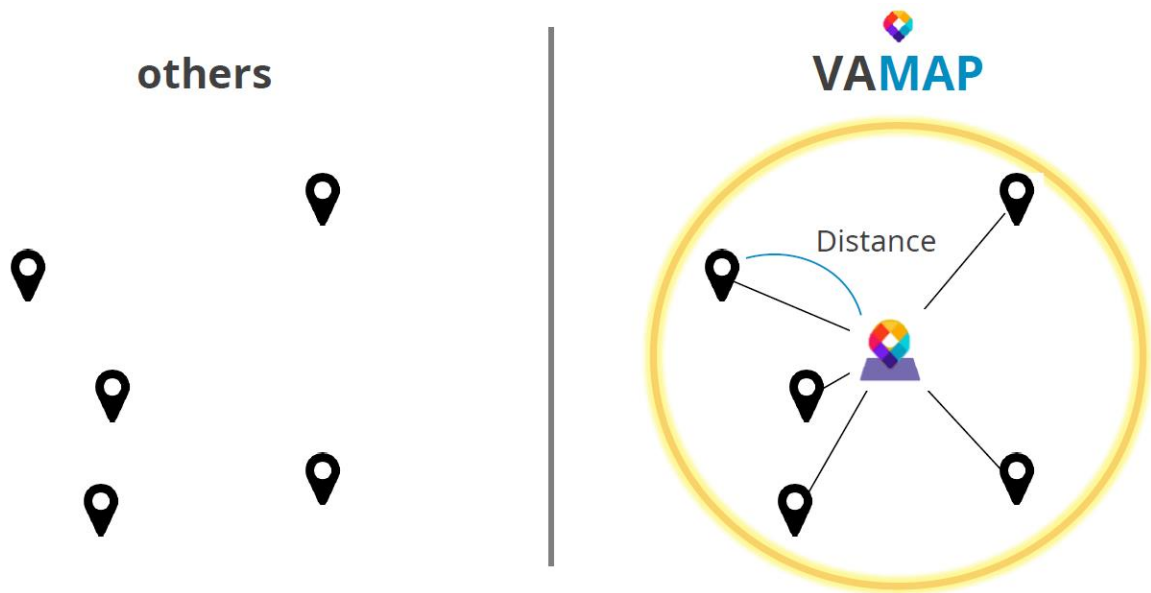
## 2.3 Disadvantage

---

역시나 숙련된 이용자가 아니라면 사용하기가 매우 어렵다. 다양한 시각화 기능을 제공하지만 데이터들의 분포를 지도상에서 확인하고 각각의 가치를 확인하고자 하는 목적이라면 큰 이점이 없다. 또한 실제로 시각화를 할 때 시간이 많이 소요된다.

## 3 Difference

---



각 유저데이터 객체가 거리 연산을 기반으로 가치를 가진다.

이 가치는 꼭 있어야할 곳에 반드시 있는지, 혹은 있지 않아도 될 곳에 무분별하게 있는지 등을 판단할 수 있는 척도가 될 수도 있고,

가치의 크기에 따라 해당 위치의 효율성도 판단할 수 있다.

QGIS, KNIME 은 기본적으로 사용자에게 일정 수준 이상의 지식을 요구한다. 예를 들어 파일을 load 하거나 지도를 불러오는 것, 지도위에 데이터를 띄우는 것이나 시각화를 하는 모든 과정이 처음 사용하는 사용자에게는 매우 어려운 일이다.

반면 VAMAP 은 먼저 Setup.txt 에서 요구하는 대로 파일의 형태를 사용자가 수정하기만 하고 터미널에서 오직 'python main.py Setup.txt' 만 실행하면 파일 내의 내용이 Setup.txt 에서 요구하는 형식인지 확인하고 엑셀파일의 주소정보를 자동으로 좌표정보로 바꿔주며 지도상에 객체의 위치와 기본데이터들의 마커가 찍힌 html 파일이 자동으로 생성된다.

또한 두 프로그램에는 존재하지 않는 유저데이터가 어떤 위치에서 얼마의 가치를 가지는지를 제공한다. 따라서 내가 가지고 있는 데이터들이 지도상에 어디에 위치하며 그 위치에서 어떤 가치를 가지는지를 알고 싶은 사용자들에게는 VAMAP 이 최선의 선택이 될 것이다.

### 3. Vamap

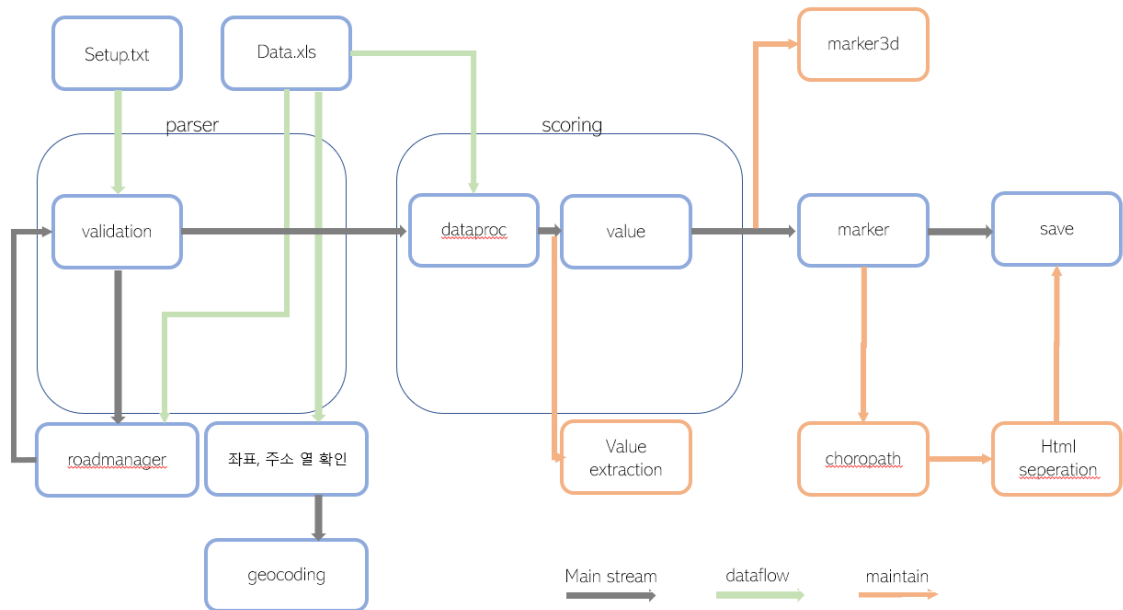
#### 1. Team, position

TEAM LEADER			
			
<b>HYUNJAE LEE</b>	<b>SUNGJAE MIN</b>	<b>SEOKCHEON JU</b>	<b>SANGMIN LEE</b>
1. README 작성 2. 도로데이터 전처리 구현 (roadmanager.py) 3. Main 함수 구성 및 기타 함수들 구현 (main.py, load.py, map3d.py, marker_func.py)	1. 사용자의 입력값들을 검증하기 위한 파싱 구현 (parser.py) 2. 데이터 수집 (choropleth)	1. Documentation 2. 입력된 엑셀 정보를 읽어서 전처리를 하는 함수 구현 (dataproc.py) 3. 데이터 수집 (user, roads, others)	1. 유저데이터와 입력데이터들간의 연산 구현 (scoring.py) 2. 데모 시연
<a href="mailto:leehj8687@gmail.com">leehj8687@gmail.com</a>	<a href="mailto:alstjdwo1601@naver.com">alstjdwo1601@naver.com</a>	<a href="mailto:smallfish06@cau.ac.kr">smallfish06@cau.ac.kr</a>	<a href="mailto:snt.oceaner@gmail.com">snt.oceaner@gmail.com</a>

그림 1 Team16 Introduction

#### 2. Architecture





**그림 2 Vamap Architecture**

Setup.txt 파일과 data 파일들이 parser 로 호출되면서 프로그램이 시작됩니다. 규칙에 맞는 Setup 일 경우에는 엑셀파일의 형식을 확인해서 올바른 경우에 연산부로 데이터를 넘깁니다. 그후 가치연산을 하여 지도위에 그 값을 표시해주게 됩니다.

### 3. Usage scenario

---

# VAMAP

A SIMPLE USER DATA-DRIVEN MAP-ANALYZING TOOL BASED ON FOLIUM

## Prerequisites

Python 3.7+ , folium 0.8.3

```
$ pip install folium
```

## Installing

1. Cloning our repo

```
$ git clone https://github.com/opensource-projectteam16/Team16_Development.git
```

2. Installing modules

```
$ pip install -r requirements.txt
```

3. Change addresses written in your excel file to location values through commands below this will return a pair of new columns that are **<column\_name>\_x** and **<column\_name>\_y** which refer latitude, longitude.

```
$ python roadmanager.py <your_roadfile.xlsx> <sheet_name> <column_name>
```

4. Set up your input variables and Run main

Write setup.txt according to setup rules (you can check them in the file) then execute command line below

```
$ python main.py setup.txt
```

5. Then you will get **MAP.html** which shows you **2 different markers**, which are **roads and others**, and **one marker** evaluated by surrounded objects that are in the given coverage

그림 3 Vamap 설치 및 실행 과정

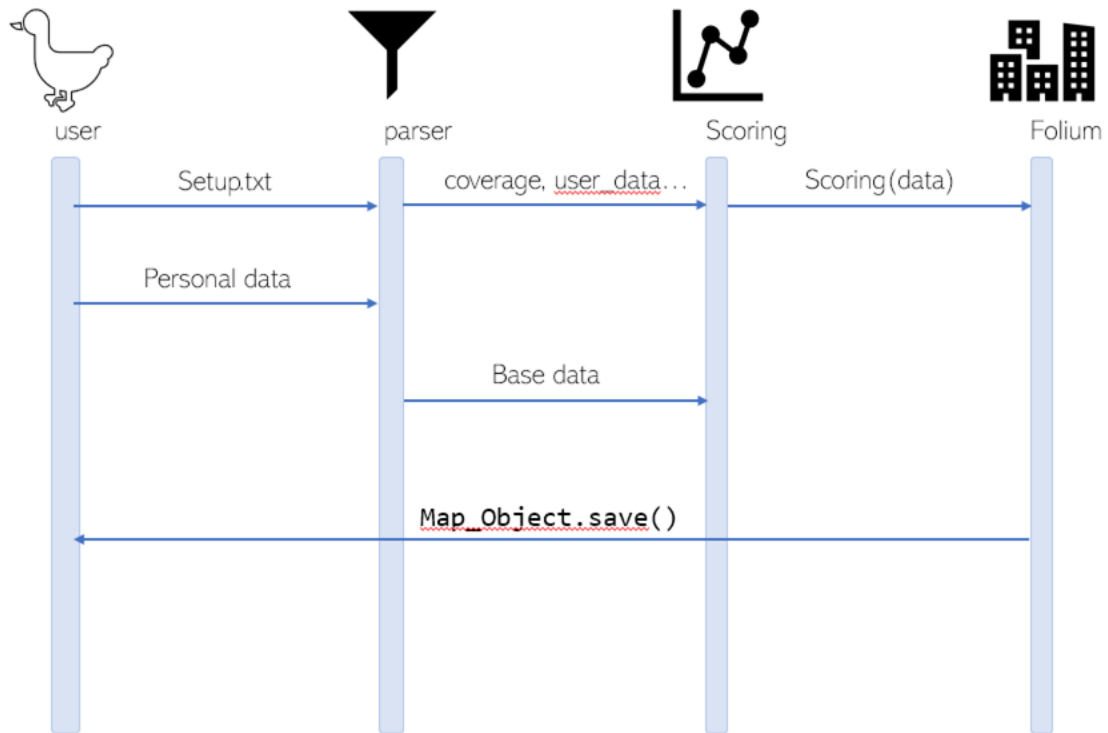


그림 4 Vamap workflow

#### 4. Use case

##### Usage Scenario

사용자는 특정 시설의 위치정보를 포함한 엑셀파일을 가지고 VAMAP 을 실행한다.  
예시 시설은 가로등 , 편의점 , 서울시 따릉이 이다.

가로등의 Coverage 는 가로등 하나가 비출 수 있는 범위, 편의점의 Coverage 는 상도,  
따릉이의 Coverage 는 역과 역 사이 거리라고 가정한다.

사용자가 알고 싶은 데이터의 가치를 계산하기 위한 기준은 다음과 같다.

- 가로등 : 길, 주거단지
- 편의점 : 길, 주거단지, 다른 편의점
- 따릉이 : 지하철, 유동인구

VAMAP 을 실행 했을 때 결과물인 html 파일을 보고 사용자는 각각의 시설이 지도상에 어떤 분포를  
가지는지 얼마만큼의 가치를 가지고 있는지를 확인할 수 있으며

편의점 점주 :

“ 이 곳에 너무 많은 편의점이 분포하여 새로운 편의점이 필요하지 않겠다. ”

가로등 건설업체 :

“ 주거 단지가 많고 길이 많아 유동 인구가 많은 지역임에도 가로등이 없어 위험하겠다. ”

따릉이 이용자 :

“ 유동인구도 많고 지하철역 바로 앞이라 이 곳에 따릉이가 있으면 정말 편할 것 같다. ”

다양한 사용자에게 분포의 시각화, 데이터의 가치를 제공함으로써 위와 같은 판단을 돕는 것이 VAMAP의 목표이자 특징이다.

## 1. Use case example – Bicycle rental

[Base]

사용자 : 자전거 대여소 관리/설치 업체 업주

[Needs]

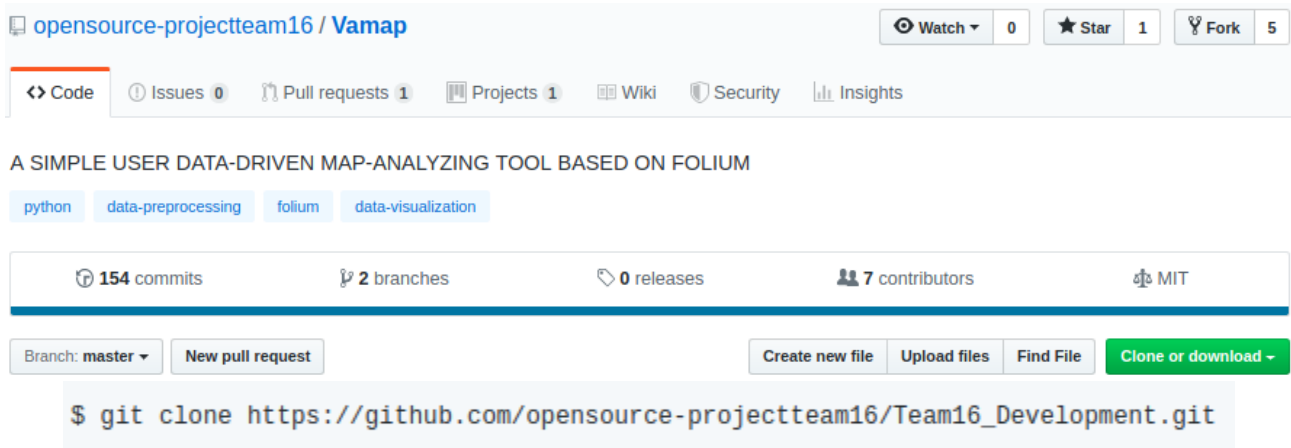
어느 위치에 자전거 대여소를 설치하는게 적합할까?

어느 자전거 대여소에 더 많은 자전거를 둘까?

현재 설치된 자전거 대여소를 계속 유지해야할까?

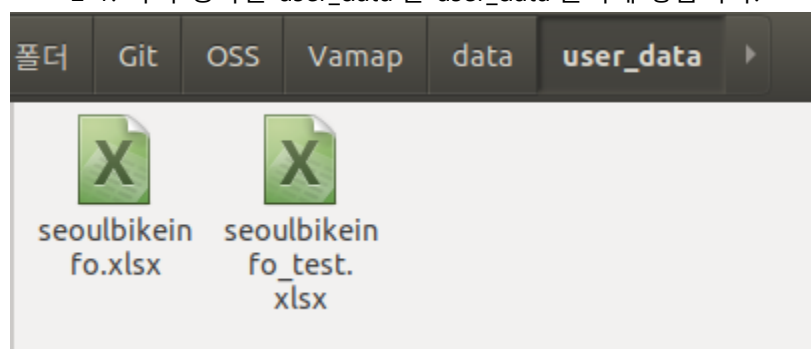
1. 사용자는 github의 다음 경로에서 Vamap 프로그램을 받습니다.

<https://github.com/opensource-projectteam16/Vamap>

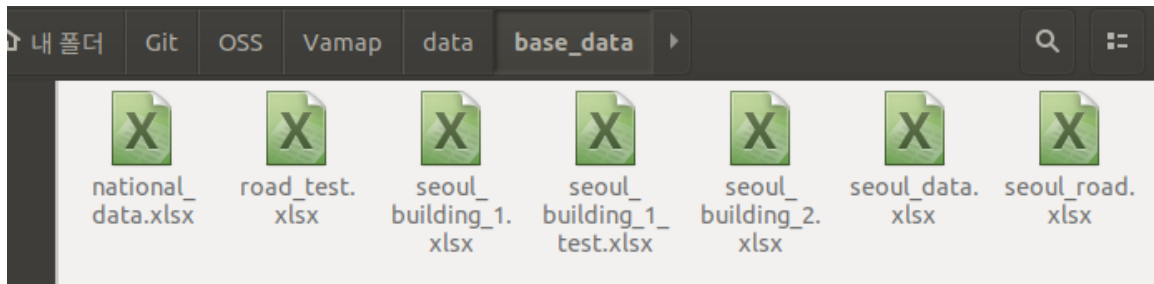


2. 사용자는 분석할 파일들을 VAMAP/data 폴더에 로드합니다.

2-1. 가치 평가할 user\_data는 user\_data 폴더에 넣습니다.



2-2. 평가에 적용될 데이터는 base\_data 폴더에 넣습니다.



2-3 (optional). 만약 데이터가 위도, 경도가 아닌 주소(도로명 등)로 표현되어 있다면 roadmanager.py 를 실행하여 이를 위도, 경도 기반 데이터로 변환할 수 있습니다.

```
$ python roadmanager.py <your_roadfile.xlsx> <sheet_name> <column_name>
```

A	B	C	D	E	F	G
name	start	end	value	length	x	y
새문안로	신문로1가 74-2	당주동 100-5	40	177	37.5708845	126.9733308
육인6가길	노상동 166-108	노상동 166-178	3	86	37.5814197	126.9651757
자하문로	적선동 8-1	홍지동 99-7	1	29	37.5869804	126.9690947
삼일대로17길	관철동 13-14	관철동 13-22	9	286	37.5691894	126.9860744
사직로1길	행촌동 171-287	행촌동 171-290	4	50	37.5736483	126.9629479
평창31길	평창동 96-4	평창동 463	8	418	37.6118628	126.9760646
창신5라길	창신동 226-32	창신동 232	2	31	37.5743516	127.0134821
동송4길	동송동 2-69	동송동 2-13	4	38	37.5840163	127.0049441
혜화로6가길	혜화동 15-167	혜화동 15-171	2	20	37.5896241	127.0009887
종로63길	송인동 72-58	송인동 72-95	1	27	37.5749909	127.0186635
이화정1나길	이화동 9-672	이화동 9-429	2	26	37.5777358	127.007768
창신5라길	창신동 226-32	창신동 226-32	2	9	37.574664	127.0136347
낙산성곽서길	종로6가 1-63	종로6가 1-47	1	15	37.5743017	127.008452
지봉로12가길	송인동 59-33	송인동 59-33	4	13	37.5764354	127.0167968
혜화로11길	명륜1가 33-93	명륜1가 33-51	2	22	37.58875	126.9986365
청경궁로26길	명륜4가 160-5	명륜4가 160-2	2	29	37.582117	126.99984

그림 5 roadmanager 실행 전

A	B	C	D	E	F	G	H	I	J	K
name	start	end	value	length	x	y	start_x	start_y	end_x	end_y
새문안로	신문로1가 74-2	당주동 100-5	40	177	37.5708845	126.9733308	37.47937	126.9269	37.47937	126.9269
육인6가길	노상동 166-108	노상동 166-178	3	86	37.5814197	126.9651757	37.44499	127.0493	37.44499	127.0493
자하문로	적선동 8-1	홍지동 99-7	1	29	37.5869804	126.9690947	37.47937	126.9269	37.47937	126.9269
삼일대로17길	관철동 13-14	관철동 13-22	9	286	37.5691894	126.9860744	37.44505	127.0599	37.67048	126.8821
사직로1길	행촌동 171-287	행촌동 171-290	4	50	37.5736483	126.9629479	37.44505	127.0599	37.44505	127.0599
평창31길	평창동 96-4	평창동 463	8	418	37.6118628	126.9760646	37.4507	126.9086	37.4507	126.9086
창신5라길	창신동 226-32	창신동 232	2	31	37.5743516	127.0134821	37.41911	127.1019	37.4507	126.9086
동송4길	동송동 2-69	동송동 2-13	4	38	37.5840163	127.0049441	37.44499	127.0493	37.44499	127.0493
혜화로6가길	혜화동 15-167	혜화동 15-171	2	20	37.5896241	127.0009887	37.4507	126.9086	37.4507	126.9086
종로63길	송인동 72-58	송인동 72-95	1	27	37.5749909	127.0186635	37.4507	126.9086	37.4507	126.9086
이화정1나길	이화동 9-672	이화동 9-429	2	26	37.5777358	127.007768	37.4507	126.9086	37.4507	126.9086
창신5라길	창신동 226-32	창신동 226-32	2	9	37.574664	127.0136347	37.4507	126.9086	37.4507	126.9086
낙산성곽서길	종로6가 1-63	종로6가 1-47	1	15	37.5743017	127.008452	37.47937	126.9269	37.47937	126.9269
지봉로12가길	송인동 59-33	송인동 59-33	4	13	37.5764354	127.0167968	37.44499	127.0493	37.44499	127.0493
혜화로11길	명륜1가 33-93	명륜1가 33-51	2	22	37.58875	126.9986365	37.47937	126.9269	37.47937	126.9269
청경궁로26길	명륜4가 160-5	명륜4가 160-2	2	29	37.582117	126.99984	37.44505	127.0599	37.67048	126.8821

그림 6 roadmanager 실행 후

3. Setup.txt 에서 프로그램 실행에 필요한(입력한) 데이터와 해당 데이터가 갖는 가중치 등을 입력합니다

```
# Coverage : this is absolute meter-wise distance centered on each user data, which will include all possible value objects. (50 ~ 1000m)
500

# User data file, sheet name, latitude, longitude, weight(-1.0 ~ 1.0)
seoulbikeinfo_test.xlsx, Excel_Import_1, 위도, 경도, -0.4

# The number of value files (this must be at least 1)
4

# The number of road file
2

# Weights (number of value files = number of weights)
0.3, 0.3, 0.2, 0.2

# Road file, sheet name, latitude, longitude, start latitude, start longitude, end latitude, end longitude (value, value-weight(-10 ~ 10 ))
# This can be empty
road_test.xlsx, road_2, x, y, start_x, start_y, end_x, end_y, value, 3
road_test.xlsx, road_3, x, y, start_x, start_y, end_x, end_y, value, 4

# Other value files, sheet name, latitude, longitude, (value, value-weight( -10 ~ 10 ))
seoul_building_1_test.xlsx, building_2, x, y, value, 3
seoul_building_1_test.xlsx, building_3, x, y, value, 4

# additional excutable file

# User select range of latitude, longitude (start , end)
0,0, 0,0
```

4. 프로그램 실행에 필요한 모듈을 설치합니다.

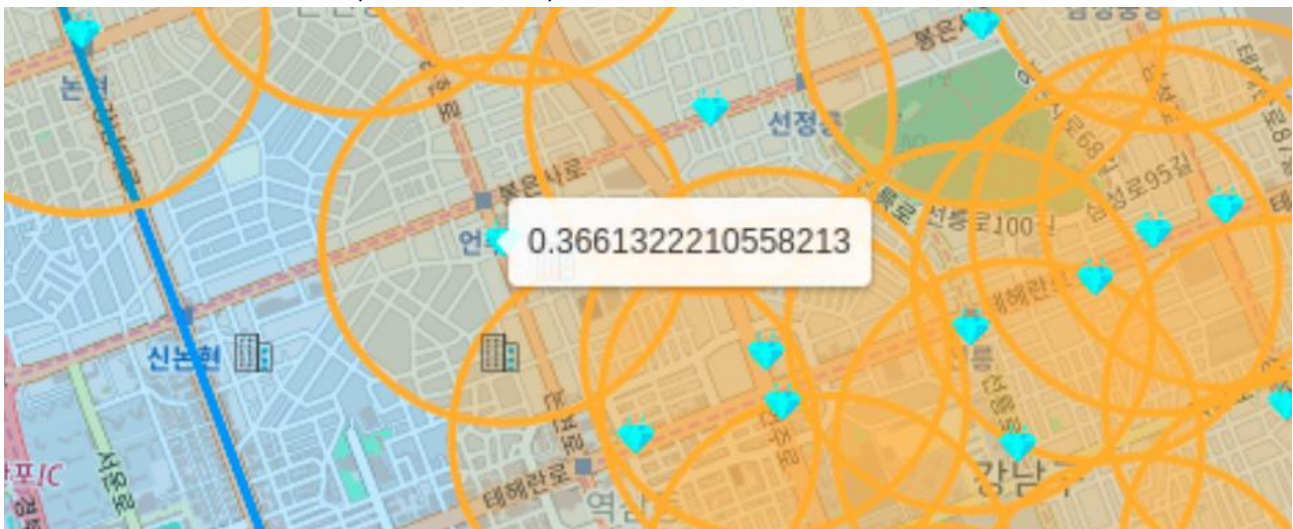
```
$ pip install folium
```

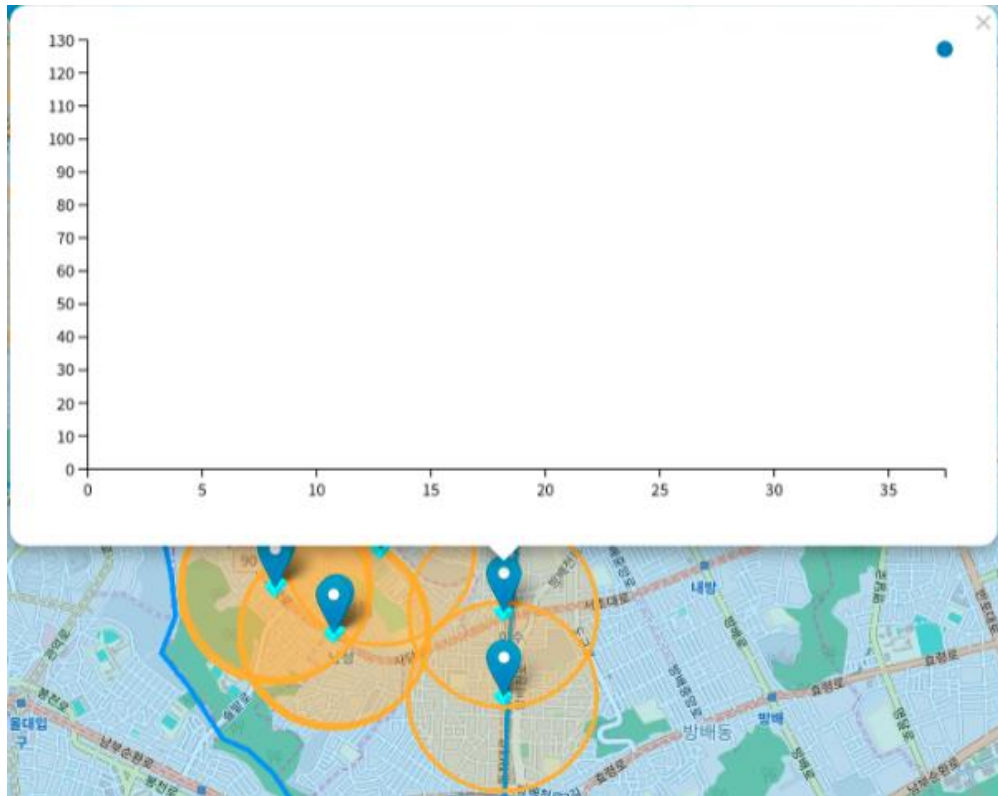
```
$ pip install -r requirements.txt
```

5. 다음 명령어를 통해 프로그램을 실행합니다.

```
$ python main.py setup.txt
```

6. Vamap 폴더 내에 Map.html 에서 결과를 확인하고 분석합니다.





[분석 예시]

사용자 : 자전거 대여소 관리/설치 업체 업주

[지점 1]

“지점 1은 해당 커버리지 내에 빌딩이 두채나 있네.

“여기에 있는 자전거 대수는 늘려야 겠다.”

[지점 2]

“지점 2는 해당 커버리지 내에 다른 자전거 대여소가 하나 있네.

여기에 있는 자전거 대수를 조금 줄여 다른 곳에 보완하고

## Scalability of Use case

본 프로젝트는 다음과 같은 확장성을 가집니다.

[경우 1] 지도가 설계도면인 경우

User : 건축업자

Needs : 건설 자재와 도구를 건설현장의 어디에 두어야 가장 효율적일까?

User data : 건설자재, 도구

base data : 건설중인 빌딩, 건설현장 통로

Satisfy : 건설 자재와 도구를 이용하기에 가장 시간효율, 공간효율적으로 각 자재와 도구를 배치할 수있다. 예를 들어 고정된 크레인을 어디에 위치시켜야 가장 공간 효율적이고 활용성이 있는 지에 대해 분석 가능하고, 이를 통해 최적의 배치를 도출할 수 있다.

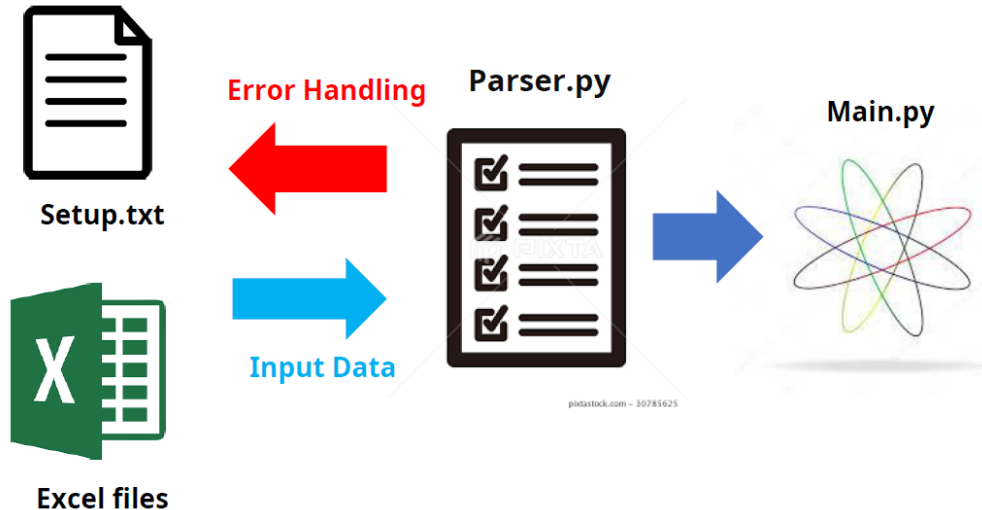
[경우 2] 쇼핑 단지를 구축하려고 하는 경우

User : 구역 활성화 담당자

Needs	: 접근성 확보를 통해 많은 유입인구를 확보해 해당 지역을 활성화하고자 한다.
User data	: 쇼핑 단지 입지 후보 구역
base data	: 도로, 주거단지, 근처 문화 공간, 근처 쇼핑 공간
Satisfy	: 가장 유입이 많은 구역이 어디인지 알 수 있다.

## 5. Implementation

### 1. Parser.



우리 팀의 프로젝트는 사용자가 Setup.txt 에 직접 데이터파일과 가중치를 지정할 수 있다.

이 과정에서 parser.py 는 Setup.txt 에 정보를 기입하는 과정에서 오류가 있는지, 입력한 데이터가 지정된 경로에 존재하는지, 그 파일 내부의 정보가 원하는 형식인지 등을 미리 사전에 파악하고  
에러가 있다면 에러 메시지를 출력해주는 사전 확인 절차를 구현한 부분입니다.

먼저 Setup.txt 에 사용자가 입력하는 정보와 고려해야 할 요소는 다음과 같습니다.

1. **Coverage** : Coverage 는 특정 데이터의 가중치를 결정하는 요소를 어디까지 포함할 것인지에 대한 경계를 의미합니다. 50m ~ 1000m 범위의 Coverage 를 10m 단위로 받길 원하기 때문에 parser.py 에서 입력 받은 Coverage 값이 10m 단위가 아니라면 10m 단위로 변환을 하며 50m ~ 1000m 범위의 값이 아니라면 에러 메시지를 띄우게 됩니다.
2. **User data** : User data 는 사용자가 원하는 엑셀 파일을 저희 프로젝트 내의 base\_data 폴더에 넣게 되는데 여기서 Setup.txt 에서는 파일의 이름, sheet 의 이름, 위도정보가 들어간 column 의 이름 , 경도정보가 들어간 column 의 이름을 입력 받습니다.

Parser.py 에서는 Setup.txt 에 입력한 정보가 위의 4가지가 맞는지 확인하고 입력하지 않은 부분이 있거나 더 많은 정보를 입력한 경우 에러 메시지를 출력합니다.

또한 이 파일의 이름이 지정된 경로에 존재하는지, 그 파일에 sheet 이름이 존재하는지, 그 sheet 내의 column 이 존재하는지를 roadmanager.py 를 호출하여 확인합니다. 여기서 에러가 생긴다면 roadmanager.py 에서 종료되며 에러 메시지를 출력합니다.

3. **Number of value files** : 이 값은 road 엑셀파일 그리고 다른 value 를 가지는 엑셀파일의 총 합을 의미하는데 무조건 하나 이상은 입력해야 하므로 1보다 작게 되면 parser.py 는 에러 메시지를 출력합니다. 또한 이 값만큼만 파일을 읽습니다.



4. **Number of road files** : 위의 value files 중에 road file 의 엑셀파일 개수를 의미합니다.

이 값 만큼만 road file 을 읽습니다.

5. **Weights** : 이 값은 뒤에 입력하는 road file, other file 에 대한 가중치를 의미합니다.

당연히 가중치의 개수는 road file, other file 의 개수만큼 입력해야 합니다.

또한 가중치를 모두 합치면 '1' 이 되어야 하며 그렇지 않다면 에러 메시지를 출력합니다.

6. **Road file argument** : Setup.txt 에서 이 부분에 사용자가 입력하는 정보는

엑셀파일의 이름, sheet 이름, 위도정보가 있는 column, 경도정보가 있는 column,

길 시작점의 위도, 경도가 있는 column, 길 끝점의 위도 경도가 있는 column

그리고 선택적으로 가중치를 주고 싶은 value 가 있는 column 과 가중치입니다.

Parser.py 에서는 입력 받을 인자가 8개 혹은 10개 이므로 인자의 개수가 8개, 10개가 아니라면 에러 메시지를 출력합니다. 인자의 개수가 맞다면 roadmanager.py 를 호출하여

파일의 이름이 지정된 경로에 존재하는지, 파일 내의 sheet 이름이 있는지, 각각의 column 들이 존재하는지를 확인합니다.

마지막으로 가중치는 -10 ~ 10 의 범위로 받기 때문에 이 범위의 값이 아니라면 에러 메시지를 출력합니다.

7. **Other file argument** : Setup.txt 에서 이 부분에 사용자가 입력하는 정보는

엑셀파일의 이름, sheet 이름, 위도정보가 있는 column, 경도정보가 있는 column

그리고 선택적으로 가중치를 주고 싶은 value 가 있는 column 과 가중치 입니다.

Parser.py 에서는 입력 받을 인자가 4개 혹은 6개 이므로 인자의 개수가 4개, 6개가 아니라면 에러 메시지를 출력합니다. 인자의 개수가 맞다면 roadmanager.py 를 호출하여

파일의 이름이 지정된 경로에 존재하는지, 파일 내의 sheet 이름이 있는지, 각각의 column 들이 존재하는지를 확인합니다.

마지막으로 가중치는 -10 ~ 10 의 범위로 받기 때문에 이 범위의 값이 아니라면 에러 메시지를 출력합니다.

#### <Setup.txt> 의 예시

```
# Coverage : this is absolute meter-wise distance centered on each user data, which will include all possible value objects. (50 ~ 1000m)
120

# User data file, sheet name, latitude, longitude
seoulbikeinfo_test.xlsx, Excel_Import_1, 위도, 경도

# The number of value files (this must be at least 1)
3

# The number of road files
1

# Weights (number of value files = number of weights)
0.3, 0.3, 0.4

# Road file, sheet name, latitude, longitude, start latitude, start longitude, end latitude, end longitude (value, value-weight(-10 ~ 10 ))
# This can be empty
road_test.xlsx, road_2, x, y, start_x, start_y, end_x, end_y, value, 4

# Other value files, sheet name, latitude, longitude, (value, value-weight(-10 ~ 10 ))
seoul_building_1.xlsx, building_2, x, y, value, 5
seoul_building_1.xlsx, building_3, x, y, value, 4
```

Setup.txt 는 main.py 에서 os.getcwd() 를 통해 현재 경로를 받아서 읽어옵니다.

이 때 경로가 Window, Linux 가 각각 다르므로 플랫폼에 따라 경로를 각각 설정해줍니다.

위의 과정에서 에러가 없었다면 Coverage , User data 를 담은 리스트, Road 엑셀파일 정보, Other 엑셀파일 정보를 main.py 로 넘깁니다.

<output>의 예시

```
120 ['seoulbikeinfo_test.xlsx', 'Excel_Import_1', '위도', '경도'] (1)0.3: ['road_test.xlsx', 'road_2', 'x', 'y', 'start_x', 'start_y', 'end_x', 'end_y', 'value', 4] (2) 0.3: ['seoul_building_1.xlsx', 'building_2', 'y', 'value', 5], '3) 0.4'
: ['seoul_building_1.xlsx', 'building_3', 'x', 'y', 'value', 4]
```

➔ Coverage , User data 의 리스트 , 엑셀파일 정보와 가중치의 dictionary

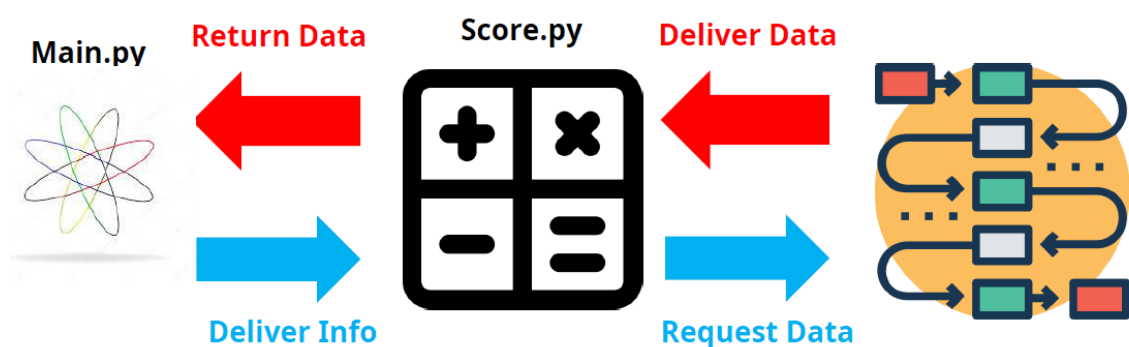
## 2. Dataproc.

parser 에서 Setup.txt 의 데이터를 읽어 필요한 데이터시트의 정보를 list 의 형태로 userdata, roaddata, basedata 인지에 대한 정보와 함께 전달받게 됩니다.

각각의 파일에서 엑셀 sheet 에서 해당하는 column 을 입력 받으면 해당 column 의 집합을 찾은 후 row 를 모아 타입에 맞게 다시 저장하게 됩니다. Excelmanager 에서는 각각의 sheet 를 process 하고 반복적인 작업을 수행합니다.

```
>>> qu.getdata()
[array([[23., 15., 10.]),
      [(14., 15., 10.)),
      [(24., 12., 10.)),
      [( 4., 24., 10.)),
      [( 3., 24., 10.)),
      [( 2., 24., 10.)),
      [( 1., 24., 10.))],
      dtype=[('x', '<f8'), ('y', '<f8'), ('value', '<f8')]), array([[23., 15.,
10.)),
      [(14., 15., 10.)),
      [(24., 12., 10.)),
      [( 4., 24., 10.)),
      [( 3., 24., 10.)),
      [( 2., 24., 10.)),
      [( 1., 24., 10.))],
      dtype=[('x', '<f8'), ('y', '<f8'), ('value', '<f8')]), array([[23., 15.,
10.)),
      [(14., 15., 10.)),
      [(24., 12., 10.)),
      [( 4., 24., 10.)),
      [( 3., 24., 10.)),
      [( 2., 24., 10.)),
      [( 1., 24., 10.))],
      dtype=[('x', '<f8'), ('y', '<f8'), ('value', '<f8')])]
```

## 3. Scoring.



Scoring 은 Dataproc 과 parser 을 통해 로드한 데이터를 연산하여 가치를 도출하는 모듈입니다.

기준이 되는 유저 데이터로부터 커버리지 내에 해당 연산은 위도, 경도로 표현된 데이터를 바탕으로 M 단위의 거리를 도출하고 이에 가중치를 곱하여 도출된 각 값의 총합이다.

연산이 완료되면 Main.py 로 연산에 적용된 user\_data (가치값을 포함), road\_data, building\_data 리스트를 전달합니다.

연산은 다음과 같은 식 입니다.

1. Input : 두 데이터의 위도 경도 좌표

A\_data 위치 : [A\_lat, A\_lon]  
B\_data 위치 : [B\_lat, B\_lon]

두 데이터 간의 위,경도 차  
D\_lat = A\_lat - B\_lat  
D\_lon = A\_lon - B\_lon

2. 위도 경도로 표현된 데이터 간의 거리 도출 식 (KM 단위)

```
km_distant = 6367 * 2* asin((sin(D_lat / 2) ** 2
+ cos(A_lat) * cos(A_lat) * sin(D_lon / 2) ** 2)^1/2)
```

(M 단위)

```
m_distant = km_distant * 1000
```

3. 가치값

Value = sum ( 사용자 입력 가중치 \* 거리 )

## 4. Roadmanager

name	start	end	value	length
새정1길	신원동 2-3	신원동 33-2	7	60
청계산로	원지동 528-8	원지동 4-54	3	700
새정2길	신원동 2-3	신원동 2-3	6	53
새정길	신원동 7-4	신원동 10-5	8	56
새정길	신원동 7-4	신원동 10-9	4	20
시흥대로12길	시흥동 962	시흥동 957-11	6	71
시흥대로17길	시흥동 983-13	시흥동 982	6	198
원터길	원지동 380-4	원지동 520-6	7	916
시흥대로26길	시흥동 954-16	시흥동 947-10	8	24
시흥대로26길	시흥동 954-16	시흥동 947-10	6	31
시흥대로34길	시흥동 942	시흥동 942-5	6	126

start_x	start_y	end_x	end_y
37.4794	126.927	37.4794	126.927
37.445	127.049	37.445	127.049
37.4794	126.927	37.4794	126.927
37.4451	127.06	37.6705	126.882
37.4451	127.06	37.4451	127.06
37.4507	126.909	37.4507	126.909
37.4191	127.102	37.4507	126.909
37.445	127.049	37.445	127.049
37.4507	126.909	37.4507	126.909
37.4507	126.909	37.4507	126.909
37.4507	126.909	37.4507	126.909

그림 2 Roadmanager 동작결과

Roadmanger 는 사용자가 주소 값을 가진 엑셀 파일을 쉽게 변환할 수 있도록 도와주는 모듈입니다. 위의 그림1 처럼 'start' 와 'end'에 해당하는 컬럼에 있는 주소 값들을 roadmanager 가 위도와 경도의 값들로 바꾸어 엑셀 파일에 저장합니다.

2. Change addresses written in your excel file to location values through commands below this will return a pair of new columns that are `<column_name>_x` and `<column_name>_y` which refer latitude, longitude.


```
$ python roadmanager.py <your_roadfile.xlsx> <sheet_name> <column_name>
```

그림 3 Roadmanager Usage

## 6. Implementation after MVP

### 1. Hyunjae Lee.

## 3D Map creating as output #5

 Closed

hyunjae-lee opened this issue 22 days ago · 2 comments



hyunjae-lee commented 22 days ago

Member

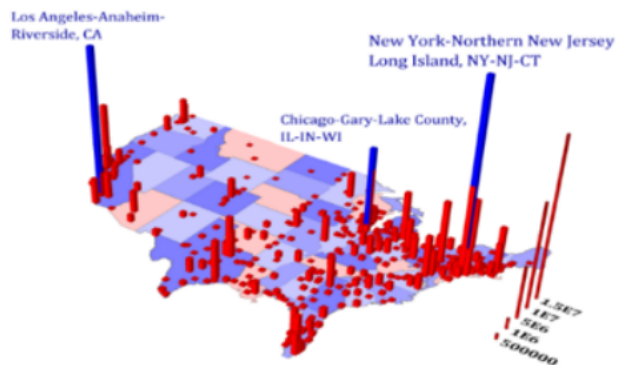
+  ...

Is your feature request related to a problem? Please describe.

현재는 folium을 통해 html 파일을 생성하고, 해당 html 파일은 2D map을 보여준다. 하지만 직관적인 시각화를 돕기 위해 3D map 을 출력값으로 보여주는 것을 고안하였다.

Describe the solution you'd like

### US Metropolitan Population Distribution



위와 같은 결과물을 예상한다.

그림 4 3차원 맵 구현 제시안 (Github Issue)



hyunjae-lee commented 22 days ago

Author

Member

+  ...

관련 자료 :

[https://nbviewer.jupyter.org/github/ocefpaf/folium\\_notebooks/blob/a68a89ae28587e6a9fbb21e7a1bd6042183a11bf/test\\_3D.ipynb](https://nbviewer.jupyter.org/github/ocefpaf/folium_notebooks/blob/a68a89ae28587e6a9fbb21e7a1bd6042183a11bf/test_3D.ipynb)

실제 테스트 결과 :

작동안됨.

대체안 :

<https://basemaptutorial.readthedocs.io/en/latest/basemap3d.html>

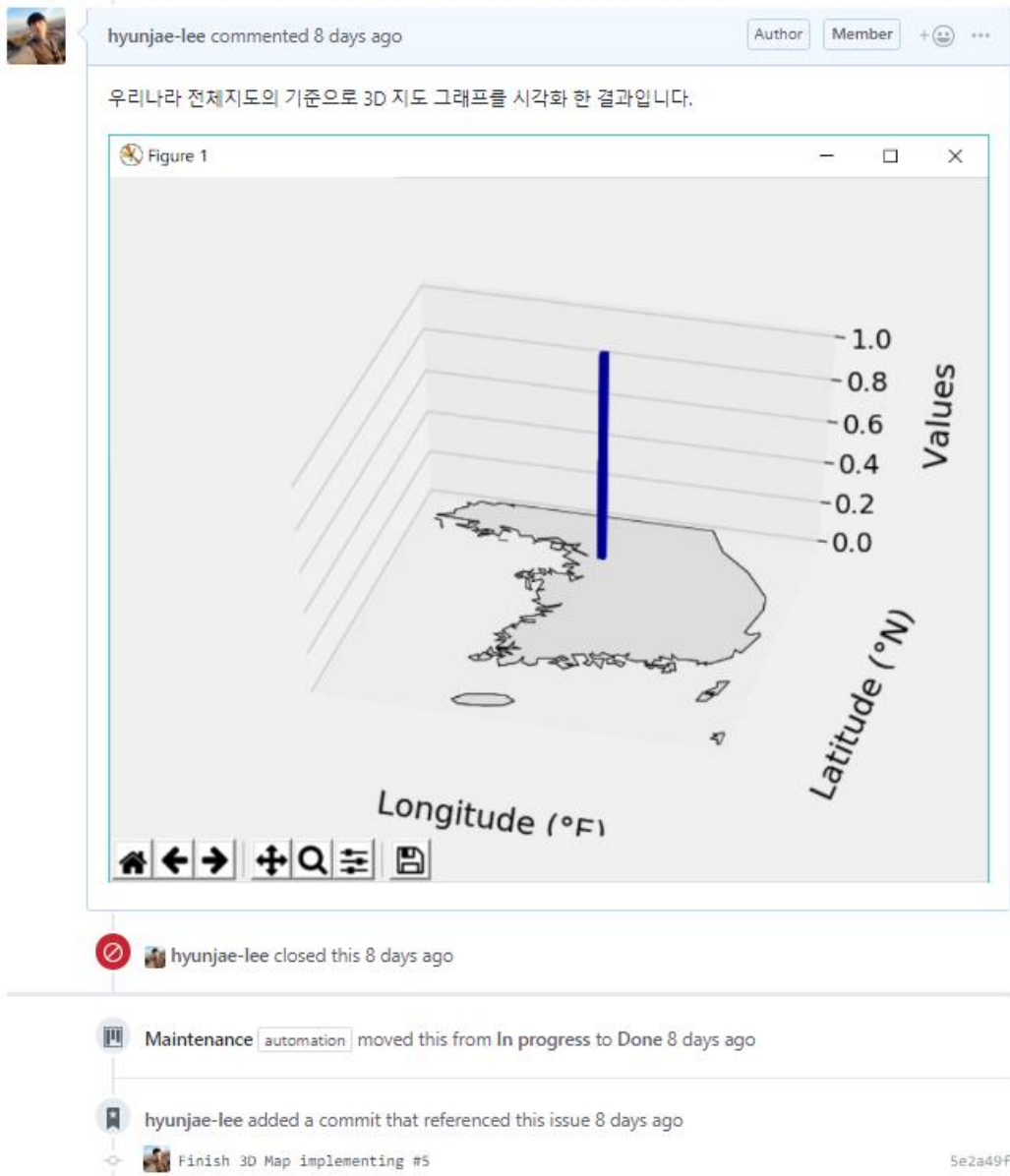


그림 5 3차원 맵 구현 결과

2차원의 지도에 데이터를 표기하는 것도 하나의 데이터 시각화 방법이지만, 더 다양하고 풍부한 시각화 툴을 제공하기 위해 Basemap 을 활용한 3차원 지도 데이터 시각화를 구현하였다. X 축과 Y 축이 각각 위도와 경도를, Z 축은 유저데이터 각 객체들의 Value 값들을 0과 1사이로 정규화를 시킨 값이다. 2차원 지도에서 각 유저데이터 값들을 아이콘으로만 표현을 했다면, 3차원 지도에서는 크기로 보여주기 때문에 더욱 직관적인 이해를 도울 수 있다.

## 2. Sangmin Lee

sangminBangba... commented 23 days ago • edited ▾

Member

+ 😊 ...

**Is your feature request related to a problem? Please describe.**

현재 User\_data를 기준으로 Coverage내의 Data간의 거리 연산한 값만을 Value로 제공  
정보의 다양화를 위해 다른 가치 연산의 필요성 제고 (단, 이 가치연산은 유효한 정보여야 함)

**Describe the solution you'd like**


Dispersion(분산, 표준편차) value를 연산하여 제공

$$v = \frac{\sum_{i=1}^n (X_i - \bar{X})^2}{n}, \quad \sigma = \sqrt{v}$$


**Describe alternatives you've considered**

정보로써 효용가치가 있는 다른 Value가 있다면  
해당 연산 또한 개발될 필요가 있음



 sangminBangbada changed the title ~~Dispersion value scoring~~ Variation value scoring  
23 days ago




 hyunjae-lee added this to In progress in Maintenance 23 days ago



 sangminBangbada added the enhancement label 23 days ago



 hyunjae-lee assigned sangminBangbada 11 days ago

sangminBangba... commented yesterday

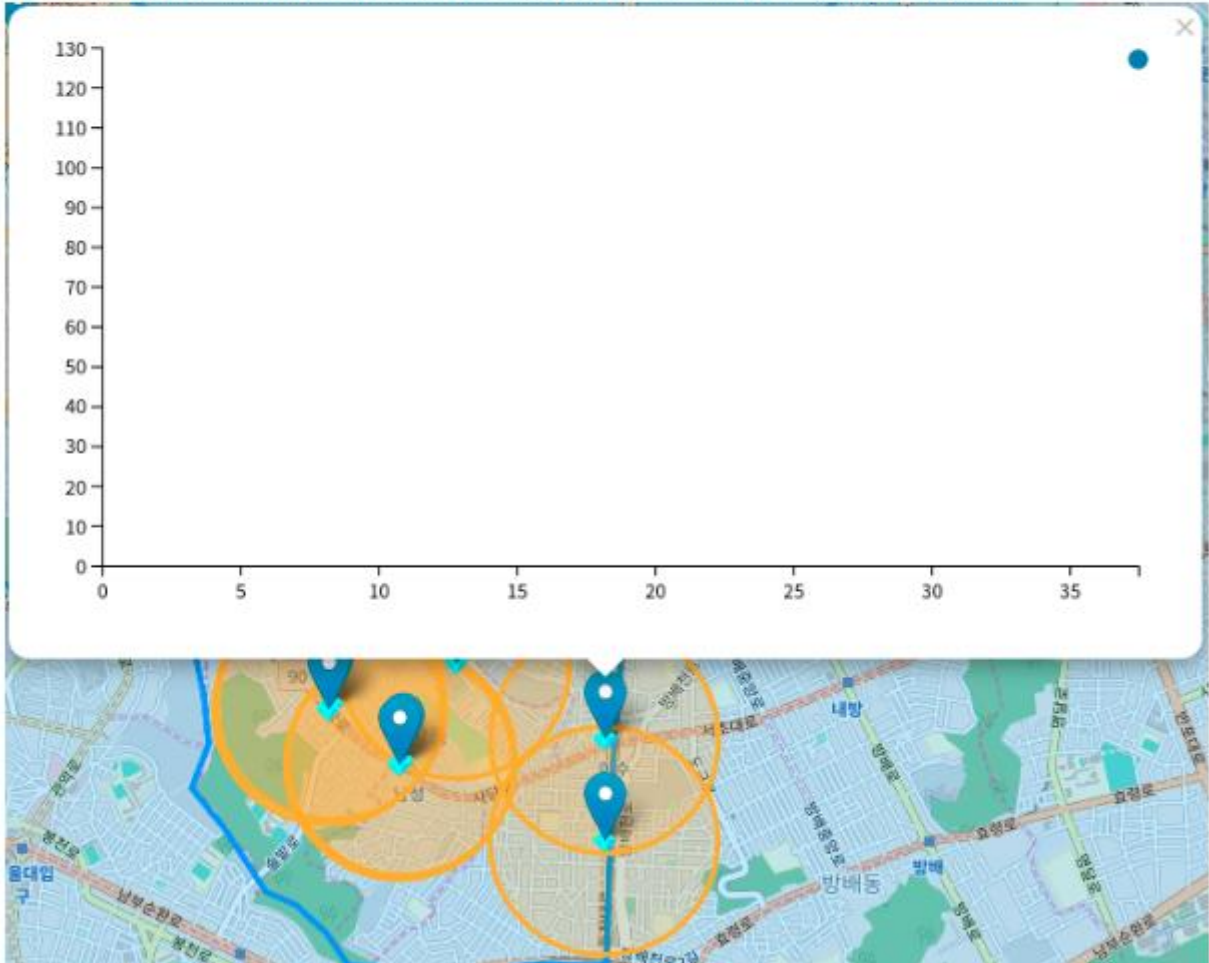
Author

Member

+😊 ...

지도에 표현된 유저 데이터의 커버리지 내에 있는 인풋 데이터 (유저데이터, 로드데이터, 빌딩데이터)의 분포도를 시각화한 자료입니다.

이를 통해 해당 마크의 커버리지 내에 있는 데이터들의 분산정도를 더 직관적으로 파악 가능합니다



Vamap 은 가치를 분석해 그 값을 제공하는 프로젝트인 만큼 가치로 표현될 수 있는 값을 선정, 이를 정확하게 연산하는것이 중요하다. Vamap 에서 각 데이터간의 거리와 사용자가 입력한 가중치를 통해 제공하는 거리기반 가치연산 다음으로 가치를 제공하는 방식으로 분산을 선정했다.

분산은 각 분야에서 데이터를 분석하는데 널리 쓰이는 연산으로 해당 프로젝트에서는 기준이 되는 데이터로 부터 얼마나 분산되어 있는지를 제공함을 통해 다양한 분석을 도출할 수 있다.

기존에는 가치를 분산식을 적용한 가치값으로 제공하려고 기획하였다. 하지만 데이터 분석 측면에서 직관적이고, 활용성이 뛰어난 방식으로 수치보다는 그래프를 통해 유저데이터를 기준으로 얼마나 분포되어 있는지를 보여주는 것이 더 효율적이라 판단하였다.

다음은 유저 데이터의 커버리지 내에 있는 데이터를 그래프상에 띄운 결과이다. 그래프의 x 축은 커버리지 내의 데이터 위도, y 축은 커버리지 내의 데이터 경도를 나타낸다.



## 1) choropleth 기능 추가



alstjdwo1601 commented 22 days ago

Member + 👤 ...

Is your feature request related to a problem? Please describe.

folium 을 통해 출력되는 html 결과물에 우리가 추가한 각 행정구역별 json 파일을 통해 전국의 행정구역을 나누어놓았다. 여기에 각 행정구역마다 주제를 정하여 시각적으로 알아볼 수 있게 색칠을 하되 단계적으로 색이 칠해지는 기능을 추가할 것이다.


예를 들어, 각 행정구역별로 실업률을 보여준다면 실업률이 높은 행정구역은 좀더 짙은 색을 주고 실업률이 낮은 행정구역은 좀더 옅은 색을 줘서 실업률이라는 주제에 각 행정구역이 어떤 분포를 가지는지를 시각적으로 알아볼 수 있는 기능을 추가할 예정이다.

Describe the solution you'd like

### United States unemployment rate, by county November, 2016



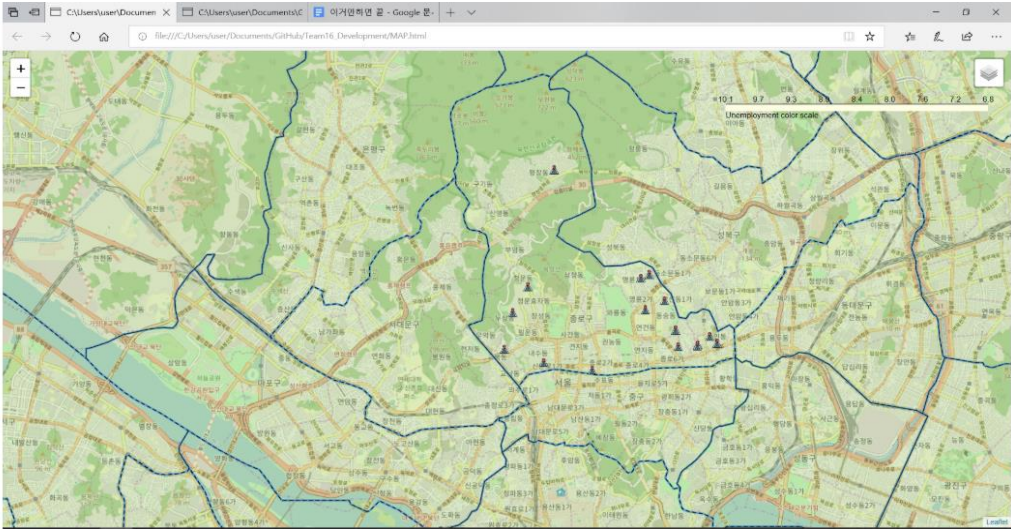
그림 6 choropleth 구현 제시안 (Github Issue)



alstjdwo1601 commented 2 hours ago

Author Member + 👤 ...

실행결과 사진



json 구역별로 색을 입히는 choropleth 기능과 네비게이션 버튼 , color bar 추가

그림 7 choropleth 구현 (Github Issue)

먼저 folium 라이브러리와 branca.colormap 라이브러리를 import 한다.

프로젝트의 json 파일의 경로를 입력 받는다.

그리고 folium.GeoJson 의 style\_function 을 통해서 경계면의 색이나 형태, 구역 별 칠할 색 등을 커스터마이징 할 수 있다. 이렇게 커스터마이징한 지도를 html 로 출력할 지도 객체에 추가해준다.

그리고 LayerControl , color bar 를 추가하였다.



## 2) requirements.txt 이슈 처리



njw1204 commented 17 days ago



@ Pull request 보내기 전에 반드시 Issue 를 아래의 템플릿으로 작성하여 주세요.

@ 해당 Issue 를 저희 팀에서 검토하고, 해결이 필요한 Issue 일 경우 Pull request 요청을 드릴게요.

@ Pull request 를 진행하실 때는, 본 프로젝트의 contribute 브랜치에 요청해주세요.

### 1. 이슈 제목

**requirements 수정 및 설치법에 반영**

### 2. 이슈 발생 환경

#### 2.1 이슈 환경

**Python 3.6 / Windows 10**

### 3. 이슈 내용

#### 3.1 자세한 내용

README.md의 설치법에 나온대로 folium만 설치하게 되면 아래와 같은 오류가 발생합니다.

requirements.txt로 설치하려고 해도, 현재 requirements.txt에는 필요 없는 패키지가 다수 포함되어 있어서 설치가 어렵습니다. 프로젝트를 분석해서 필요한 패키지만 requirements에 남기고 README.md의 설치법을 수정하고자 합니다.

#### 3.2 스크린샷 및 결과 로그

```
Traceback (most recent call last):
  File "main.py", line 2, in <module>
    import pandas as pd
ModuleNotFoundError: No module named 'pandas'
```

저희 requirements.txt 에 저희 로컬에 설치된 모든 패키지가 포함되어 있어서 검색해본 후 필요가 없는 패키지는 삭제를 했습니다.

그리고 저희 VAMAP 을 처음 사용할 때 가이드가 될 수 있는 requirement guide 문서를 GITHUB 에 추가했습니다.



alstjdwo1601 commented 20 hours ago

Member



저희 requirements.txt 에 있던 패키지중에 필요없는 것들을 확인해서 제거했고 저희 프로그램을 맨 처음 클론하여 쓸 때 참고 할 수있는requirement guide를 추가하였습니다.  
의견 감사드립니다.



alstjdwo1601 closed this 20 hours ago

## 4. Seokcheon Ju

## html seperation feature #4

[Edit](#)[New issue](#) Closed

smallfish06 opened this issue 22 days ago · 1 comment



smallfish06 commented 22 days ago

Member

+ 🗨️ ...

Is your feature request related to a problem? Please describe.


VAMAP outputs certain format of html file, hard to modify this html file manually

Describe the solution you'd like

I'm going to add module that will separete output into a format that user wants.

ex) 2x2,3x3, or depend on the input that user wants to add in html file

Assignees

 smallfish06

Labels

enhancement

Projects

Done in Maintenance

Milestone




No milestone

Notifications



Unsubscribe

You're receiving notifications because you modified the open/close state.

1 participant

 Lock conversation hyunjae-lee added this to In progress in Maintenance 22 days ago smallfish06 added the enhancement label 22 days ago hyunjae-lee assigned smallfish06 10 days ago smallfish06 added a commit that referenced this issue 3 days ago Issue<#4> added html seperation code but need to updated.

744e7ce

 smallfish06 added a commit that referenced this issue 3 days ago Issue<#4> added html seperation code but need to updated.

949e2b3

자동으로 map 의 html 파일이 생성되었을 경우 그거를 수동으로 바꾸기는 어려움이 많다. 그렇기 때문에 파일에서 생성한 다양한 결과물 파일들을 한번에 보여주는 maintain 작업을 하였다.

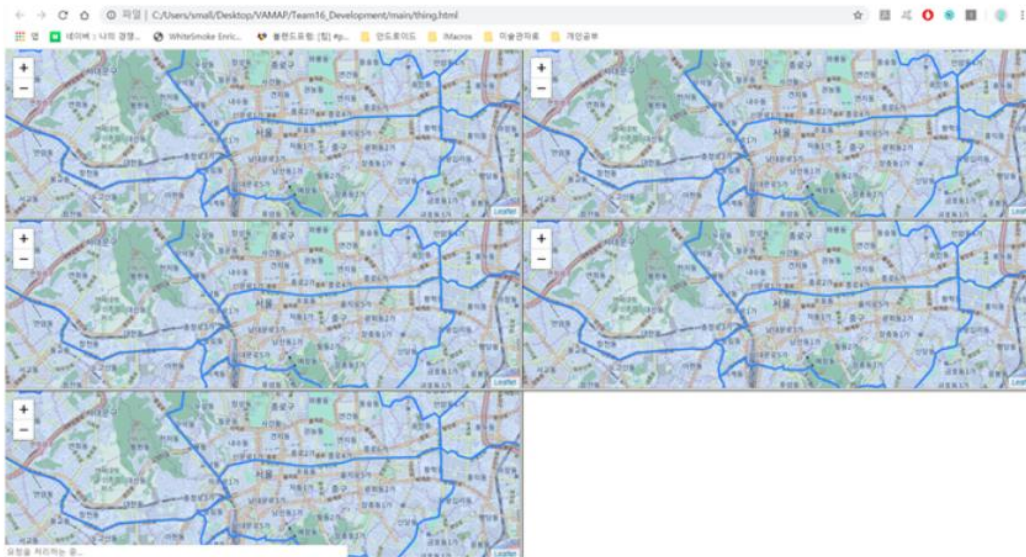


smallfish06 commented 1 minute ago

Author

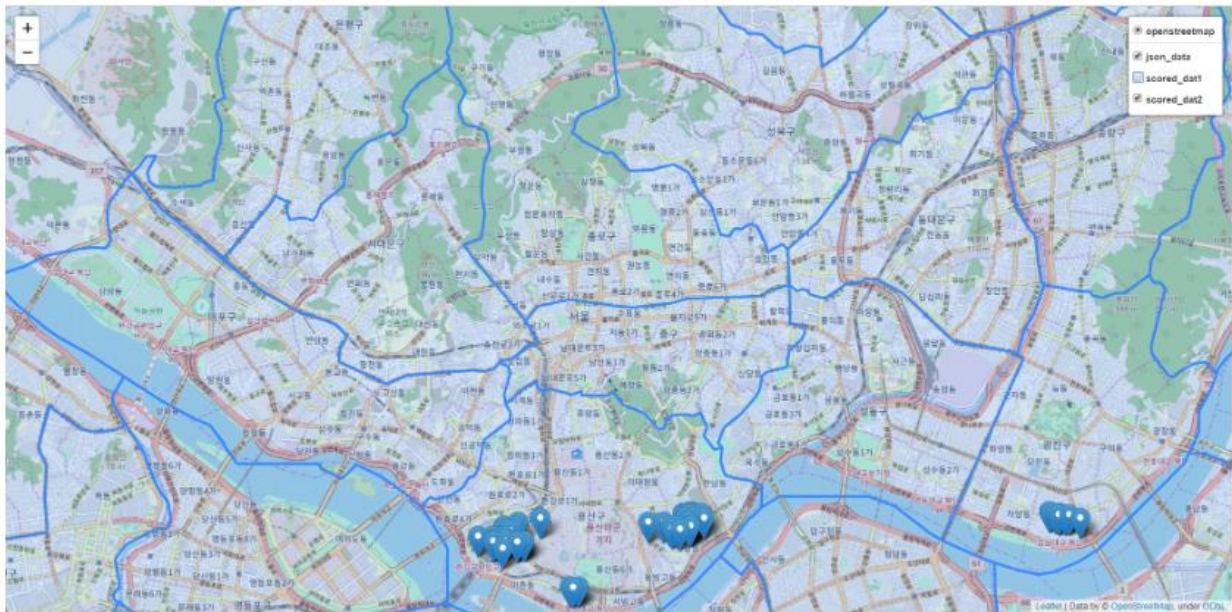
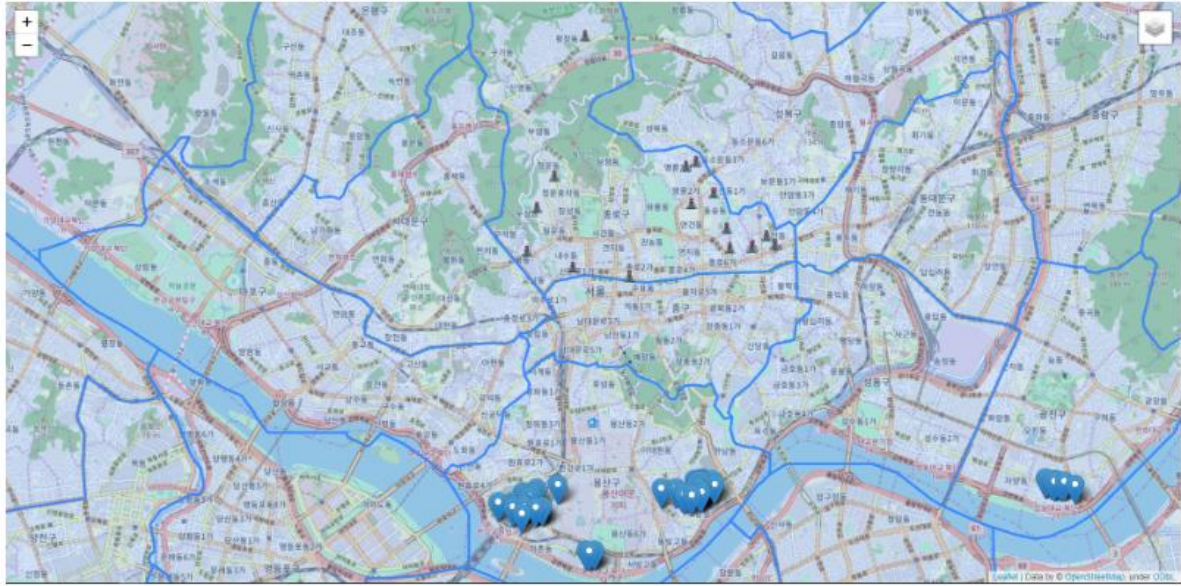
Member

+ 🗨️ ...



1. added html seperation code. Several variations are available. Horizontly 2\*2 etc. If you want to make different schema, just modify some values.

기본적으로는 2열로 출력을 하며 세로로만 출력하거나 가로로만 출력하였다. 사용자가 다양한 방식으로 본인이 원하는 포맷으로 수정할 수 있도록 확장성을 두었다. 아직 다양한 variation 을 가진 함수들은 구현하지 못하였지만 단순 html 파싱을 하는 과정이라 전체적인 포맷만 제작하였다.



2. added html marker layering code. If there's many kind of markers, it would be helpful to see single layer. Also when analyzing data

또한 marker 들의 종류가 굉장히 많을 경우 가독성이나 데이터를 보기에 굉장히 불편한 경우가 있을 것이라고 생각하였다. 그렇기 때문에 오른쪽 상단에서 종류를 선택할 수 있도록 표시를 하고 marker 들을 layer 별로 구분하여 표현해야 할 필요가 있다고 생각하여 새로운 함수를 제작하였다.

## 4. Demo

### 1. roadmanager 실행

```
(Team16_Development) piu82@piu82-15UD760-GX30K:~/Git/OSS/Team16_Development$ python roadmanager.py
road_test.xlsx road_1 start
```



name	start	end	value	length	x	y	
남대로6	재동 330	재동 390	4	102	7.467005	7.0430506	
남대로8	재동 23	재동 390	10	106	7.467970	7.0426553	
남대로8	재동 23	재동 390	10	114	7.468470	7.0414934	
남대로6	재동 330	재동 390	4	130	7.468823	7.0457279	
남대로6	재동 330	재동 390	8	748	7.469109	7.0444898	
남대로8	재동 23	재동 390	6	109	7.469182	7.0424965	
남대로8	재동 23	재동 390	6	106	7.469252	7.0444068	
남대로8	재동 23	재동 390	9	129	7.469349	7.0433001	

name	start	end	value	length	x	y	start_x	start_y	end_x	end_y
남대로6	재동 330	재동 390	4	102	7.467005	7.0430506	37.46838	127.0361	37.46838	127.0361
남대로8	재동 23	재동 390	10	106	7.467970	7.0426553	37.46838	127.0361	37.46838	127.0361
남대로8	재동 23	재동 390	10	114	7.468470	7.0414934	37.46838	127.0361	37.46838	127.0361
남대로6	재동 330	재동 390	4	130	7.468823	7.0457279	37.46838	127.0361	37.46838	127.0361
남대로6	재동 330	재동 390	8	748	7.469109	7.0444898	37.46838	127.0361	37.46838	127.0361
남대로8	재동 23	재동 390	6	109	7.469182	7.0424965	37.46838	127.0361	37.46838	127.0361

2. 결과 전/후

3. 프로젝트 실행

```
(Team16_Development) piu82@piu82-15UD760-GX30K:~/Git/OSS/Vamap$ python main.py Setup.txt
Correct access
```

4. Parsing result

```
-----Read Setup.txt-----
[Coverage] : 500
[User data] : ['seoulbikeinfo_test.xlsx', 'Excel_Import_1', '위도', '경도', -0.4]
[Roads files & Weights] : {'1' 0.6': ['road_test.xlsx', 'road_2', 'x', 'y', 'start_x', 'start_y', 'end_x', 'end_y', 'value', 3]}
[Other files & Weights] : {'2' 0.4': ['seoul_building_1_test.xlsx', 'building_2', 'x', 'y', 'value', 3]}
-----
```

5. Loading excel file results

```

-----Load Excel-----
fulli/home/piu82/Git/OSS/Vamap/data/user_data
initialize

['/home/piu82/Git/OSS/Vamap/data/user_data/seoulbikeinfo_test.xlsx'] ['seoulbikeinfo_test.xlsx']
loadstart/home/piu82/Git/OSS/Vamap/data/user_data/seoulbikeinfo_test.xlsx ['위도', '경도', -0
loaded /home/piu82/Git/OSS/Vamap/data/user_data/seoulbikeinfo_test.xlsx
fulli/home/piu82/Git/OSS/Vamap/data/base_data
initialize

['/home/piu82/Git/OSS/Vamap/data/base_data/road_test.xlsx'] ['road_test.xlsx', 'road_2', 'x'
loadstart/home/piu82/Git/OSS/Vamap/data/base_data/road_test.xlsx ['x', 'y', 'start_x', 'start
loaded /home/piu82/Git/OSS/Vamap/data/base_data/road_test.xlsx
fulli/home/piu82/Git/OSS/Vamap/data/base_data
initialize

['/home/piu82/Git/OSS/Vamap/data/base_data/seoul_building_1_test.xlsx'] [['seoul_building_1_
loadstart/home/piu82/Git/OSS/Vamap/data/base_data/seoul_building_1_test.xlsx ['x', 'y', 'valu

```

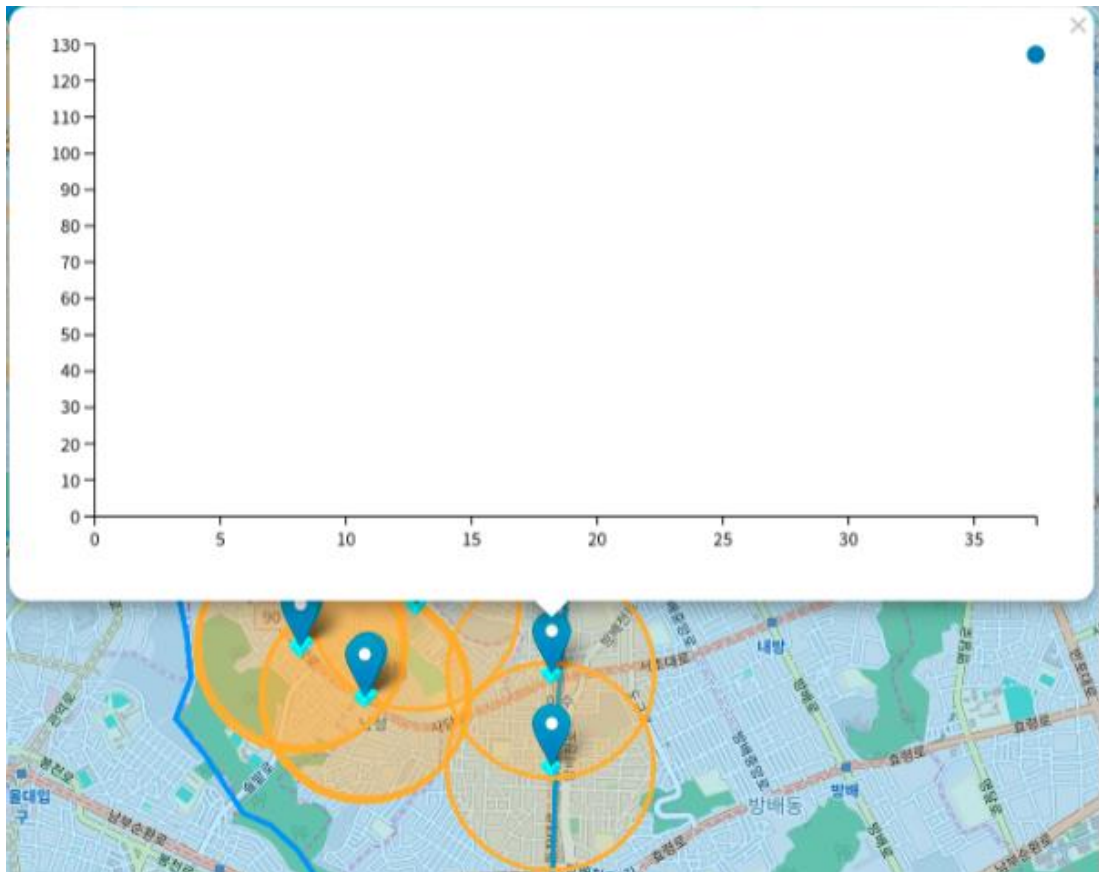
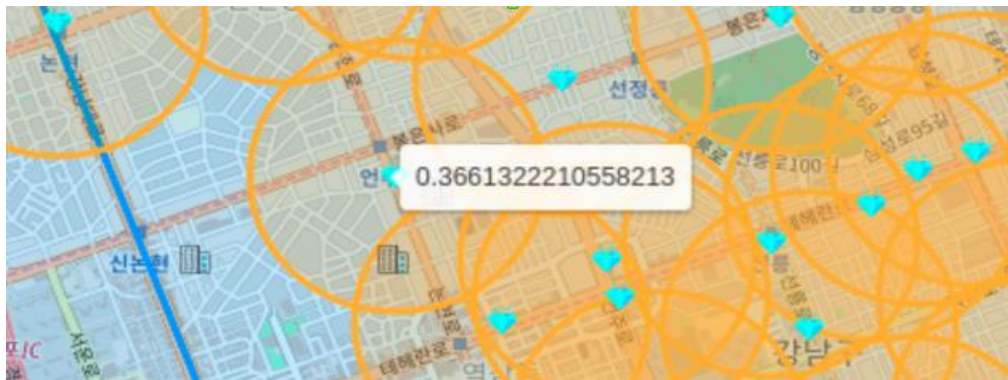
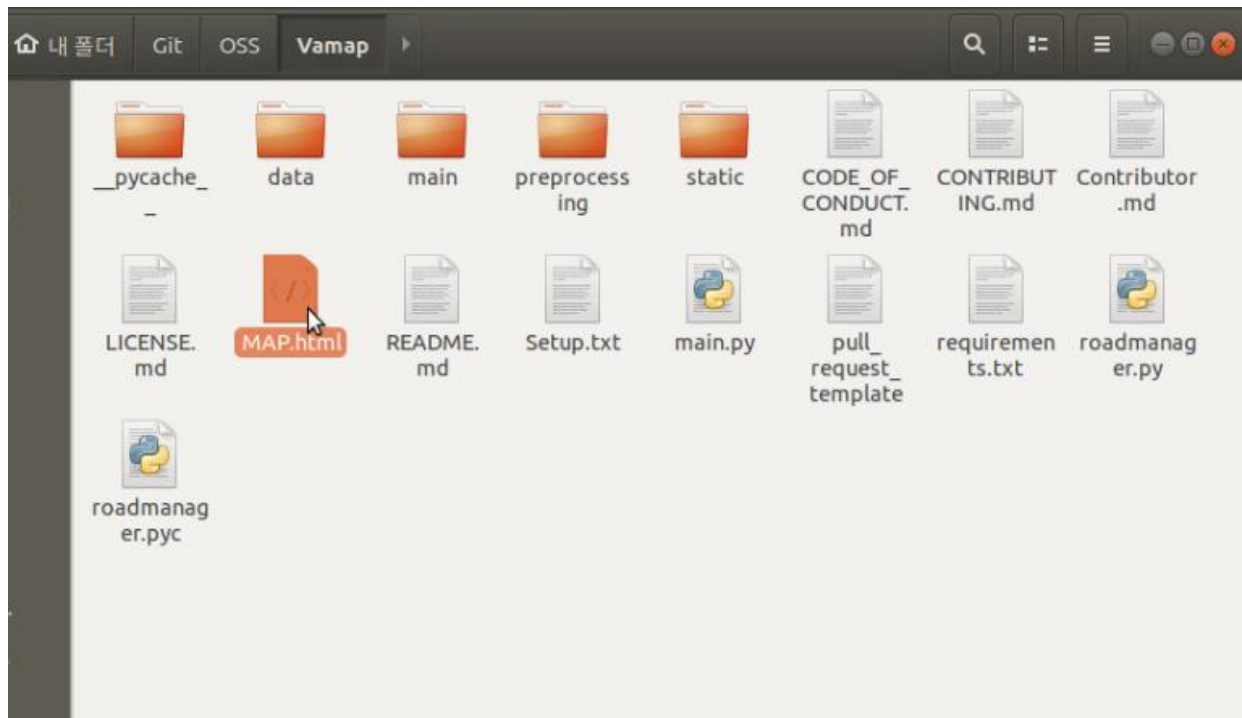
## 6. Example of scored data

```

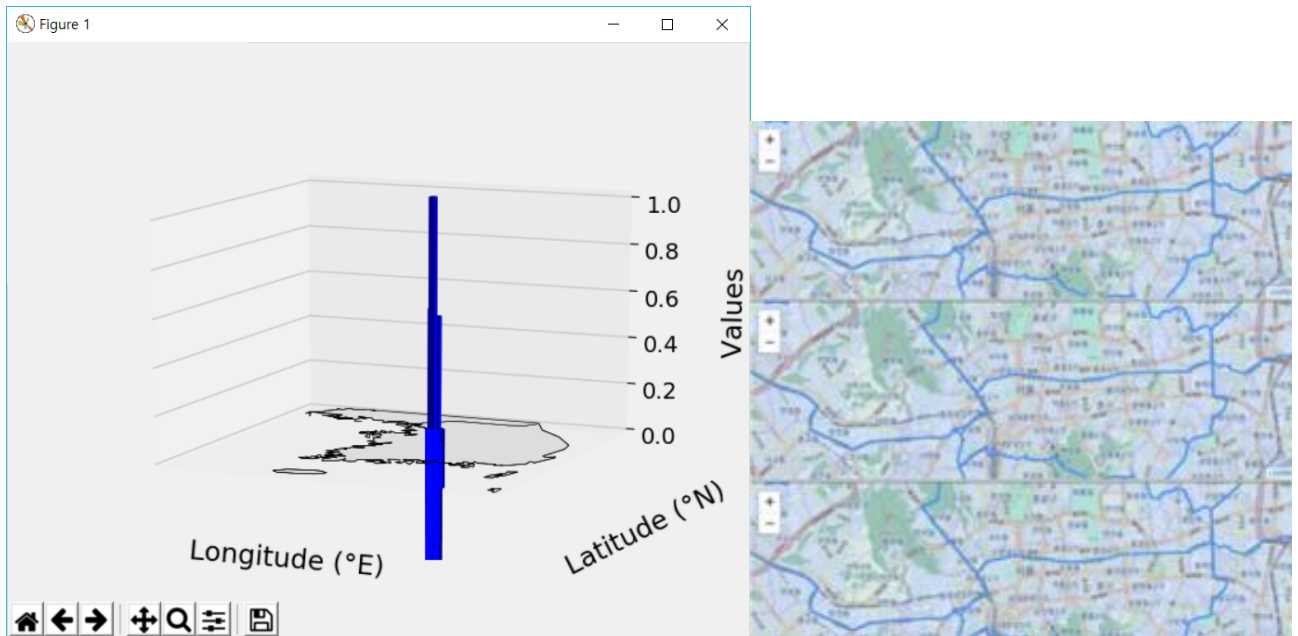
[37.466328, 127.094887, 0.0]
[37.509586, 127.040909, 0.0]
[37.48616, 127.067238, -0.22636691071169493]
[37.474579, 127.05545, 0.0]
[37.47929, 127.055733, -0.0021793426387255095]
[37.512939, 126.934242, 0.0]
[37.488361, 126.906227, 0.0]
[37.499645, 126.927094, -0.0768035556311422]
[37.50938, 126.963615, -0.03867597034393347]
[37.48447, 126.971428, -0.6533728570617965]
[37.494705, 126.982391, 0.0]
[37.506359, 126.969078, -0.03980624270307767]
[37.513248, 126.939972, -0.25187556461316446]
[37.500881, 126.94175, -0.3890307597110255]
[37.484631, 126.97155, -1.0196730135357481]
[37.512901, 126.926613, -0.2728740340102754]

```

## 7. Map.html created



## 8. 3D Map and Html separation results



3D Map 결과의 경우, 서울에 위치한 데이터들로만 측정을 해보았으며 음수 및 양수값이 나오도록 하였습니다.

9. 최종 데모 영상 비디오 링크 : <https://www.youtube.com/watch?v=cWCxjerOPwY&feature=youtu.be>

## 5. Cooperation with Github

### 5.1. Commit, contributors

#### 1. Contributors

##### a. Main developers

- HYUNJAE LEE - *Team leader* - [hyunjae-lee](#)
- SUNGJAE MIN - *Developer* - [alstjdwo1601](#)
- SANGMIN LEE - *Developer* - [sangminBangbada](#)
- SEOKCHEON JU - *Developer* - [smallfish06](#)

됨

##### b. Other contributors

**We really appreciate your contribution**

1. [jeong5575](#)

2. [ajs7270](#)

[Other Contributors [관련링크](#)]

#### 2. Maintenance 개선 전 (~5/26)

- a. Issue 의 등록 및 사용법을 잘 몰라서 사용하지 못했다.



b. Branch 의 활용이 미흡하였다. (Centralized workflow)

c. 프로젝트의 Process 관리가 없었다.

### 3. Maintenance 개선 후 (5/27 ~)

a. 개인별 maintenance 내용을 각각 이슈화 시켰다.

b. 개발이 진행되는 동안 생기는 오류나 개선점 등을 이슈화 시켰다.

c. Blaming 기능을 적극 활용하였다.

d. 우리 프로젝트의 branching workflow 를 Long-running branches 로 정하였다. Master branch 는 최종본(Main repository)으로 정하고, contribute branch 는 Integration manager 를 위해 사용하였다.

e. Maintenance 과정에서 어떤 issue 에 대한 contribution 을 다뤘다면, 해당 commit 에는 해당 issue 를 제목에 언급하도록 규칙을 정하였다.

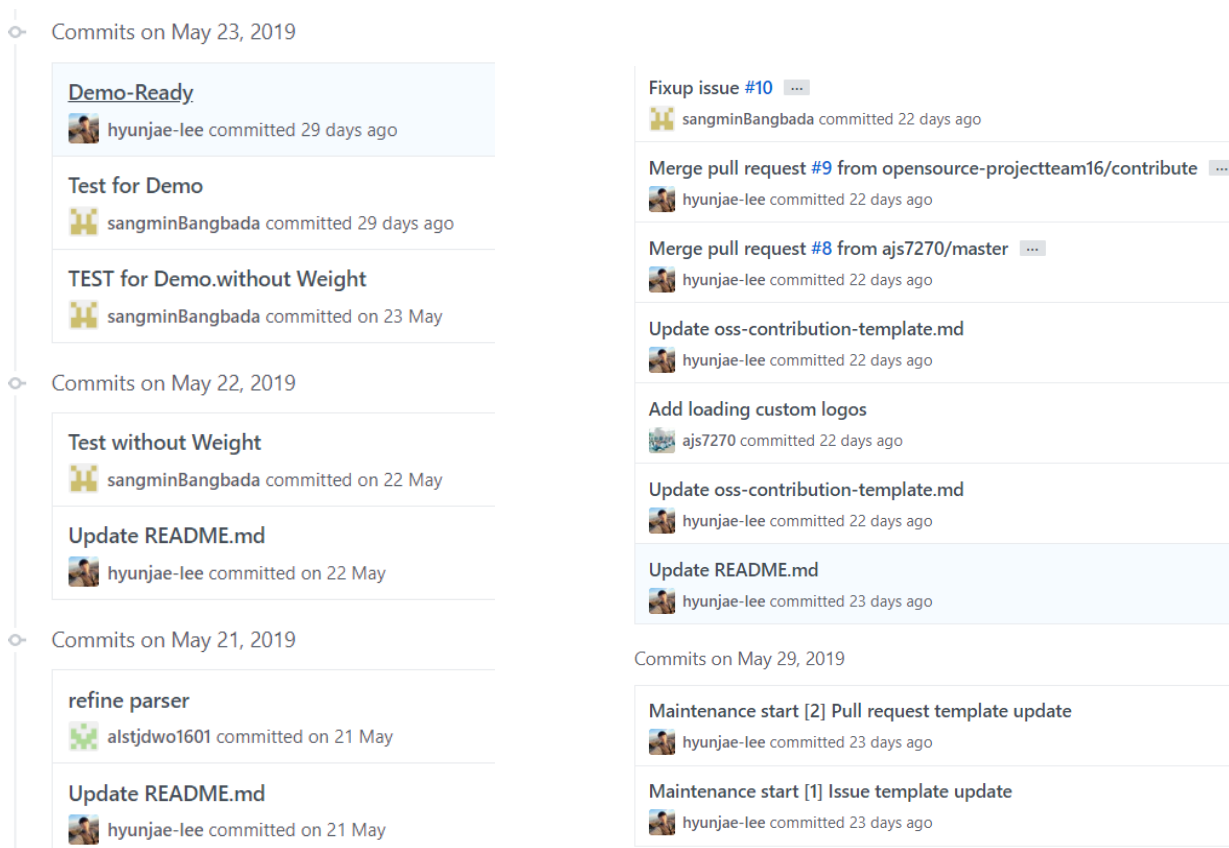


그림 8 Maintenance 개선 이전 (좌), 개선 이후 (우)

f. Github 의 Project 기능을 통해 Project process management 를 하였다.



1 Open ✓ 0 Closed

## Maintenance

Updated 22 minutes ago

Maintenance [19.05.27 ~ 19.06.21]

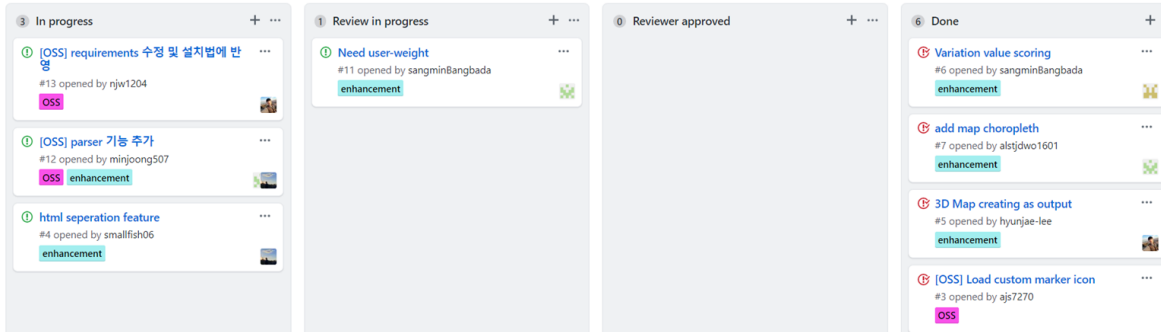


그림 9 Project process management 중간 과정 캡처 사진








## 5.2. Other issues, Pull request

### 1. Issue

<input type="checkbox"/> 0 Open ✓ 13 Closed	Author ▾	Labels ▾	Projects ▾	Milestones ▾	Assignee ▾	Sort ▾
<input type="checkbox"/> [OSS] 데이터 연산시간이 오래걸립니다. OSS	#17 by jjiijj78 was closed 7 hours ago					1
<input type="checkbox"/> Logo copyright issue	#16 by ihooni was closed yesterday					1
<input type="checkbox"/> [OSS] CSV 파일에 대한 Roadmanager.py 실행 OSS enhancement	#14 by sangminBangbada was closed 16 days ago					1
<input type="checkbox"/> [OSS] requirements 수정 및 설치법에 반영 OSS	#13 by njw1204 was closed yesterday					2
<input type="checkbox"/> [OSS] parser 기능 추가 OSS enhancement	#12 by minjoong507 was closed yesterday					
<input type="checkbox"/> Need user-weight enhancement	#11 by sangminBangbada was closed 7 hours ago					1
<input type="checkbox"/> Scoring doesn't work bug	#10 by hyunjae-lee was closed 23 days ago					1
<input type="checkbox"/> add map choropleth enhancement	#7 by alstjdwo1601 was closed yesterday					1
<input type="checkbox"/> Variation value scoring enhancement	#6 by sangminBangbada was closed yesterday					1
<input type="checkbox"/> 3D Map creating as output enhancement	#5 by hyunjae-lee was closed 9 days ago					2
<input type="checkbox"/> html seperation feature enhancement	#4 by smallfish06 was closed yesterday					1
<input type="checkbox"/> [OSS] Load custom marker icon OSS	#3 by ajs7270 was closed 21 days ago					2
<input type="checkbox"/> [OSS] Issue Example OSS	#2 by hyunjae-lee was closed 24 days ago					1

총 12개의 Issue 가 있었습니다. 프로젝트 maintainer 들은 기능 개선, 보완 등을 위한 issue 를 만들고, 본 프로젝트에 contributor 를 희망하는 외부 개발자들은 issue template 을 작성을 하여 진행을 하였습니다. 그리고, 그 중에서 개발에 필요한 내용들을 pull request template 에 맞춰 요청을 하였습니다.

## 2. Pull request

<input type="checkbox"/>	 <b>optimization by filter data</b> #18 by jjijjj78 was merged yesterday 0 of 12	 1
<input type="checkbox"/>	 <b>parser 기능 추가</b> #15 by minjoong507 was merged 4 hours ago	 3
<input type="checkbox"/>	 <b>Merge pull request for loading custom logos #3 issue</b> #9 by hyunjae-lee was merged 23 days ago	
<input type="checkbox"/>	 <b>Add loading custom logos</b> <span>OSS</span> #8 by ajs7270 was merged 23 days ago • Approved 0 of 9	 3

외부 개발자들로부터 총 3개의 pull request 를 받았으며 code review 및 최종 merge 를 하였습니다.