

대일항쟁기 강제 동원 희생자 지도 시각화

2024년도
청년인턴 데이터 분석 및 컨설팅 과제

데이터정보화담당관 청년인턴
이선희

2024. 08. 28(수)
최종 보고



CONTENTS

대일항쟁기
강제 동원 희생자 지도 시각화

01 개요

02 과제해결 방법

03 데이터 설명

04 과제 수행 방법

05 유지 보수

01

개요

대일항쟁기 강제동원 희생자 지도 시각화

대일항쟁기 강제 동원 희생자의 유해가 각국에 어느 정도 있는지 파악하고
희생자의 유해를 봉환하기 위한 중·장기 계획을 추진하기 위함

- ✓ [희생자 목록]의 데이터를 토대로 지도 시각화 진행
- ✓ 시각화한 내용을 토대로 유해가 집중되어 있는 국가, 지역을 명확하게 식별할 수 있어야 함
- ✓ 향후 지속적인 유지보수가 용의해야 함

02

과제해결 방법

지역명을 토대로 지도 시각화

- ✓ 최초 과제 요구서의 경우 분포를 확인할 수 있는 지도 시각화
- ✓ 동원지역명을 기준으로 위도 경도를 추출하고 어디에 많이 분포하고 있는지 확인할 수 있는 방법 탐색



지리 정보 시각화에 특화된 프로그램으로 다양한 좌표계와 레이어를 이용하여 시각화 가능
다만 지역명이 없거나 정확하지 않은 경우, 데이터를 나타낼 수 없다는 단점이 존재

단순한 지도 시각화를 넘어 분석 활용 방안 마련

- ✓ 지도 시각화를 하면서 데이터를 그래프화 시켜 함께 확인할 수 있으면 좋을 것 같다는 니즈 확인

지도 시각화를 위해 Q-gis 사용 시도



대시보드 제작을 위해 Streamlit 사용 결정

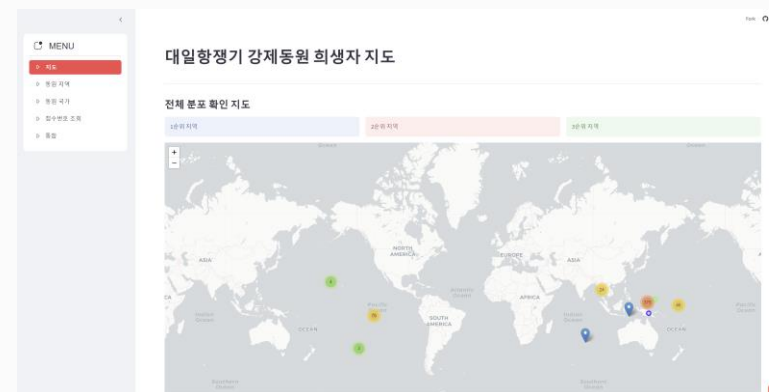




Streamlit 이란?

- ✓ 간단하게 대화형 대시보드를 제작, 배포할 수 있는 파이썬 오픈 소스 프레임워크
- ✓ 다양한 그래프를 한 눈에 확인 할 수 있으며 그래프를 바로바로 저장할 수 있음
- ✓ Github를 통하여 배포하며 유지보수에 용이
- ✓ 웹으로 배포된 Streamlit은 여러 사람이 접속해 같은 화면 확인 가능

```
1 import streamlit as st
2 import pandas as pd
3 import folium
4 from streamlit_folium import st_folium
5 from folium.map import Marker
6 from folium.plugins import MarkerCluster
7 import plotly.express as px
8 import json
9 from streamlit_option_menu import option_menu
10 import base64
11 import io
12 from io import BytesIO
13 import tempfile
14 from collections import Counter
15 from streamlit_folium import folium_static
16 import matplotlib.colors as mcolors
17 import branca.colormap as cm
18
```





Streamlit 동작방식

Streamlit 패키지 설치

Streamlit 코드 작성

Streamlit run 파일.py

`pip install streamlit`

`st.title('hello world')`

웹으로 실행

대시보드에 들어갈 내용 확인

| 대시보드 기능 | 시각화 방법 | 활용 데이터 |
|----------|-----------------|---------------------|
| 전체 분포 확인 | 지도 시각화 | 희생자 목록 - [동원지역명 수정] |
| 작업장 구분 | 파이 차트 | 희생자 목록 - [작업장] |
| 유해 봉환 확인 | 지도 시각화 & 파이차트 | 피해진상결과_봉환, 불상, 기타 |
| 동원지역 | 막대 그래프, 데이터프레임 | 희생자 목록 - [동원지역명 수정] |
| 접수번호 조회 | 지도 시각화, 데이터 프레임 | 희생자 목록 - [접수번호] |
| 통합 | 지도 시각화, 막대그래프 | 희생자 목록 |

과제 수행을 위해 해결되어야 하는 부분

1

동원지역명을 토대로
위도 경도 추출 방법 탐색

Google maps를 활용하여
위도, 경도 추출

2

데이터 연결 방법 확인

접수번호와 이름을 기준으로
데이터 통합 실시

3

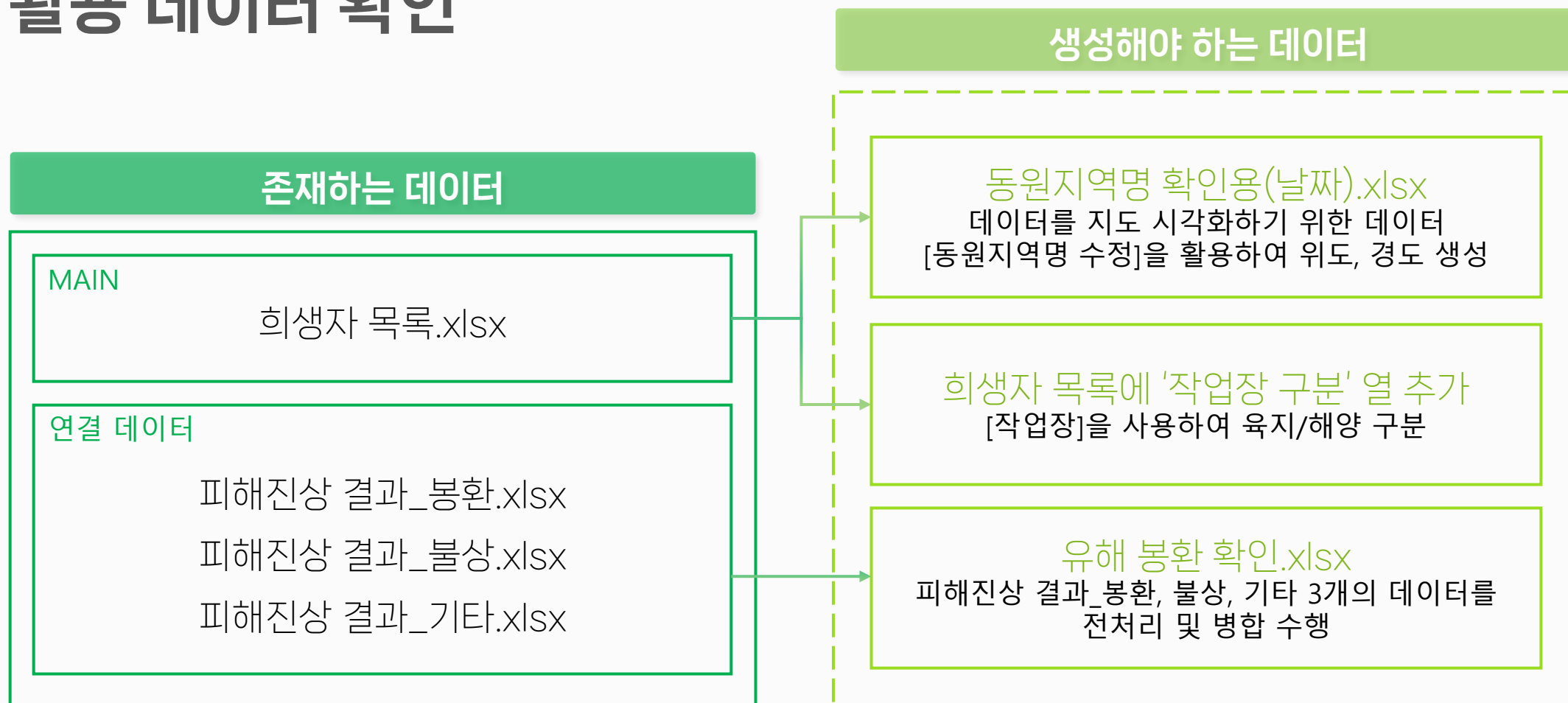
개인정보 보안을 고려한 작업

개인정보를 최대한 배제시킨
실행환경 구성

03

데이터 설명

활용 데이터 확인



희생자 목록.xlsx

| 연번 | 구분 | 접수번호 | 희생자 | 생년월일 | 사망월일 | 본적지 | 동원국가 - 수정 | 동원지(상세지역) - 수정 | 동원지역명 수정 | 작업장 |
|----|----|------|-----|------|------|-----|-----------|----------------|----------|-----|
| | | V | V | | | | | | V | V |

- ✓ 접수번호 : 고유 값으로 희생자의 개인정보를 직접적으로 Streamlit에서 확인되지 않도록 사용
- ✓ 희생자 : 유해 봉환 연결시 접수번호에서 발생하는 중복 문제를 해결하기 위해 사용
- ✓ 동원지역명 수정 : 세부적으로 나뉜 지역명으로 이를 이용하여 위도 경도를 추출
- ✓ 작업장 : 작업장의 특정 단어들을 사용하여 육지 / 해양으로 구분하는 작업 수행

피해진상 결과_봉환, 불상, 기타.xlsx

| 접수번호 | 피해자 | 유골 봉환 여부 |
|------|-----|----------|
| V | V | V |

- ✓ 피해진상 결과_봉환, 피해진상 결과_불상, 피해진상 결과_기타 3개의 데이터를 통합하여 사용
- ✓ 접수번호 : 희생자 목록 데이터와 연결 시 사용
- ✓ 피해자 : 희생자 목록 데이터와 연결 시 사용
- ✓ 유골 봉환 여부 : 희생자 목록에 새로운 열을 추가하기 위해 사용

04

과제 수행 방법

과제 흐름



데이터 확인



필요 데이터 생성



데이터 연결



Streamlit 제작



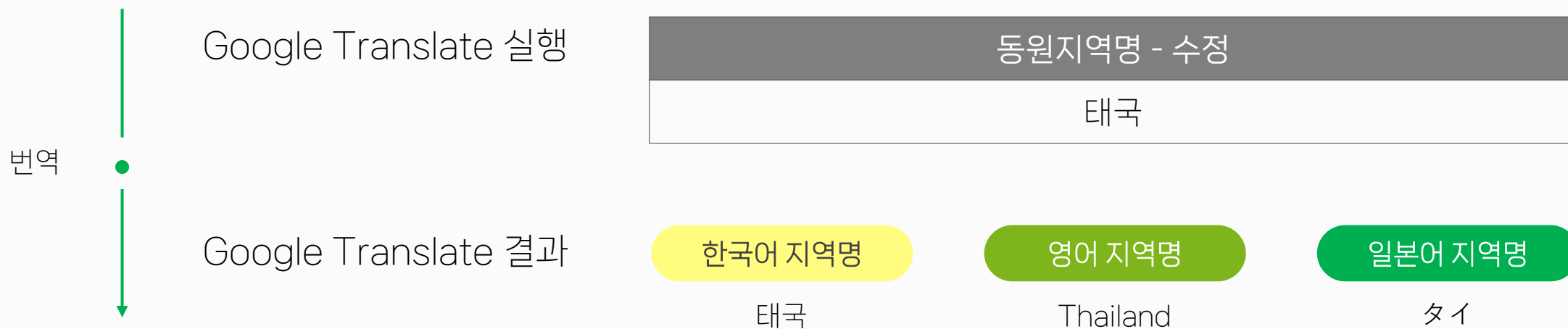
배포

필요 데이터 생성 1

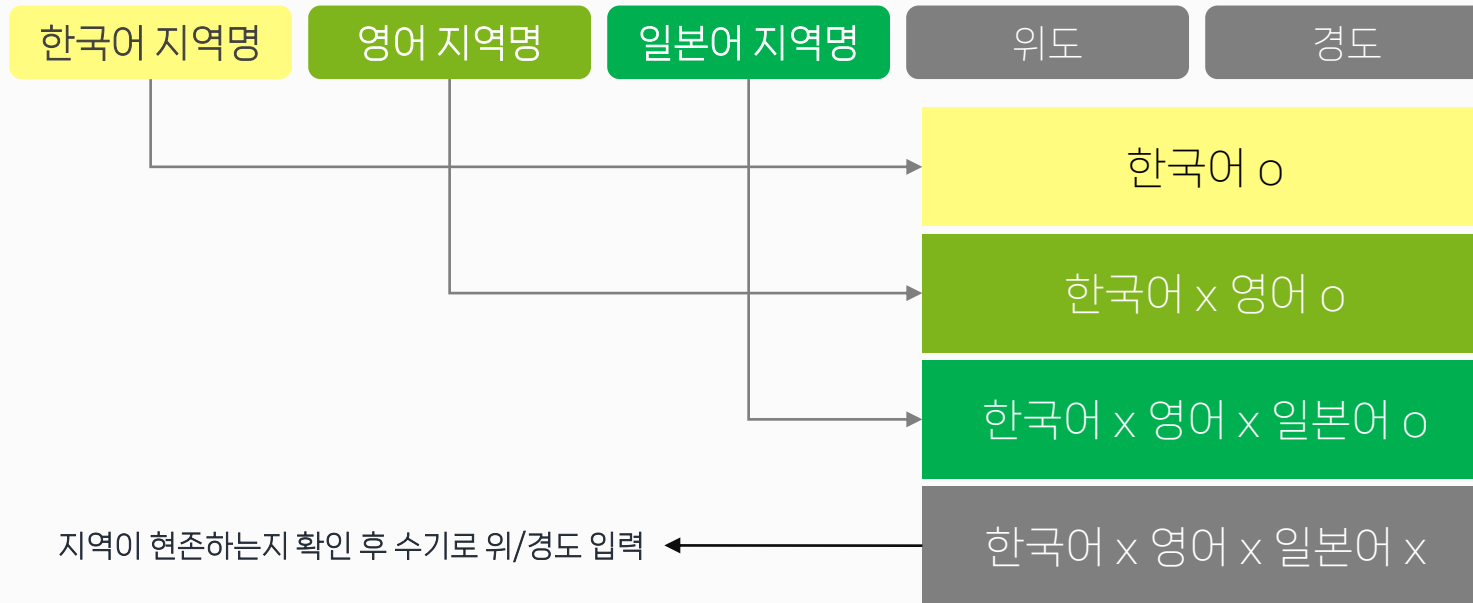
위도 경도 데이터 생성

* Google Translate : 구글의 번역 API

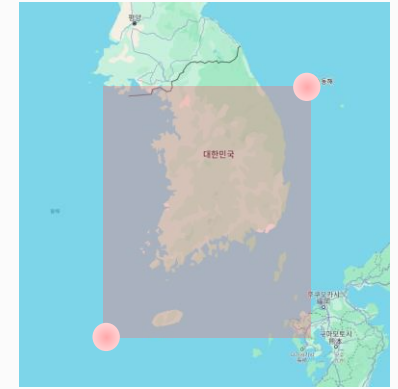
- ✓ 사용 데이터 : 희생자목록의 동원지역명 - 수정 열
- ✓ 위도 경도 추출을 수행해야 하는 데이터 : 221개
- ✓ 동원 지역명은 정확한 주소를 가지고 있지 않기 때문에 바로 위도 경도를 추출할 경우 제대로 추출 불가
- ✓ 효율적으로 위도 경도를 추출해 내기 위하여 일련의 과정을 파이썬 코드를 사용하여 수행
- ✓ **동원지역명 확인용(날짜).xlsx** → 지속적으로 업데이트 하므로 파일명의 날짜는 업데이트 된 날짜 적용



위경도
추출

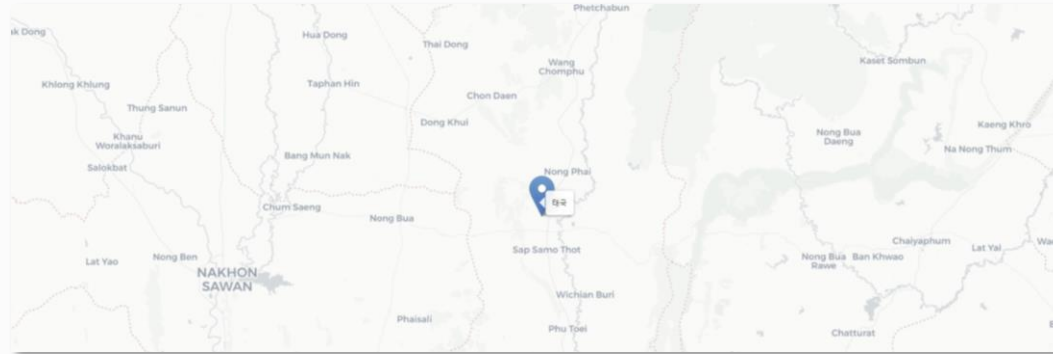


추가 조건



한국범위 내이면 다음 언어로 추출

결과



추출된 위도 경도를 확인하기 위해 folium(지도)를 사용하여 좌표 확인, 확인후 조정이 필요하다면 조정

필요 데이터 생성 2

희생자 목록 내 작업장 구분 열 생성

- ✓ 사용 데이터 : 희생자 목록
- ✓ 작업장 열의 데이터의 고유 데이터는 2171개
- ✓ 작업장명의 단어를 기준으로 육지/해양 구분
- ✓ 희생자 목록 내 **육해구분완료** 열 생성

구분 기준 1

'환', '마루', '기선', '호', '선원', '丸' 이 들어간 작업장의 경우 해양으로 분류

* '환', '호', '만'의 경우 한글자로 처리 불가능

구분 기준 2

작업장의 값이 없을 경우 '해협', '태평양', '만' 이 들어간 동원지역명을 가지고 있다면 지역명으로 작업장의 결측치 채우기
채워진 결측치는 해양으로 분류

필요 데이터 생성 2

희생자 목록 내 작업장 구분 열 생성

- ✓ 사용 데이터 : 희생자 목록
- ✓ 작업장 열의 데이터의 고유 데이터는 2171개
- ✓ 작업장명의 단어를 기준으로 육지/해양 구분
- ✓ 희생자 목록 내 **육해구분완료** 열 생성

그 외의 데이터

불상, 확인 불가 등은 구분불가 처리 후 나머지 전부 육지로 처리

육지 처리 후 분류 작업

작업장명 외에는 다른 기준이 없으며 상당수의 고유 데이터를 가지고 있음
모호한 작업장은 직접 해양 / 육지 / 구분불가 구분 진행

* 과거사관련업무지원단 대일항쟁기강제동원피해지원과에서 수행

필요 데이터 생성 3

유해 봉환 확인 데이터 생성

- ✓ 사용 데이터 : 피해진상 결과_봉환, 피해진상 결과_불상, 피해진상 결과_기타
- ✓ 엑셀 파일 내 병합된 셀 병합해제 및 불필요 데이터 삭제 후 데이터 간 병합 수행
- ✓ 열 이름 변경 : 피해자 → 희생자 변경
- ✓ 희생자 데이터 변경 : 이름(한자) → 이름 변경
- ✓ **유해봉환확인.xlsx 생성**

| 접수번호 | 피해자 | 유골봉환여부 |
|--------|-----|--------|
| 00-000 | 이00 | 봉환 |
| 00-000 | 김00 | 불상 |
| 00-000 | 박00 | 기타 |

과제 흐름



데이터 확인



필요 데이터 생성



데이터 연결



Streamlit 제작



배포

희생자 목록에 데이터 연결 수행

| 기준 데이터 | 연결 데이터 | 연결 기준 | 추가된 열 이름 |
|-------------|---------------|-------------------|---|
| 희생자 목록.xlsx | 위도 경도 확인.xlsx | 동원지역명을 기준으로 연결 | <ul style="list-style-type: none">• 위도• 경도 |
| 희생자 목록.xlsx | 유해봉환 확인.xlsx | 접수번호와 이름을 기준으로 연결 | <ul style="list-style-type: none">• 유골봉환여부 |

희생자 목록 데이터

- ✓ 완성된 희생자 목록 데이터는 조작용과 streamlit 업로드 파일로 나뉘서 관리
- ✓ 0. streamlit_데이터(날짜)_데이터 조작용.xlsx → 데이터 조작용에는 '희생자' 열 존재
- ✓ 0. streamlit_데이터(날짜).xlsx → Streamlit에 업로드 되는 데이터는 '희생자' 열 삭제
- ✓ Streamlit에 업로드 되는 데이터는 누구나 접근 가능하므로 최대한 개인정보 배제

0. streamlit_데이터(날짜)_데이터 조작용.xlsx

| 접수번호 | 희생자 | 동원국가 - 수정 | 동원지(상세지역) - 수정 | 동원지역명 수정 | 유골봉환여부 | 작업장 | 육해구분 | 육해구분완료 | 위도 | 경도 |
|------|-----|-----------|----------------|----------|--------|-----|------|--------|----|----|
| V | V | | | V | V | V | V | V | V | V |

0. streamlit_데이터(날짜).xlsx

| 접수번호 | 동원국가 - 수정 | 동원지(상세지역) - 수정 | 동원지역명 수정 | 유골봉환여부 | 작업장 | 육해구분 | 육해구분완료 | 위도 | 경도 |
|------|-----------|----------------|----------|--------|-----|------|--------|----|----|
| V | | | V | V | V | V | V | V | V |

과제 흐름



데이터 확인



필요 데이터 생성



데이터 연결



Streamlit 제작

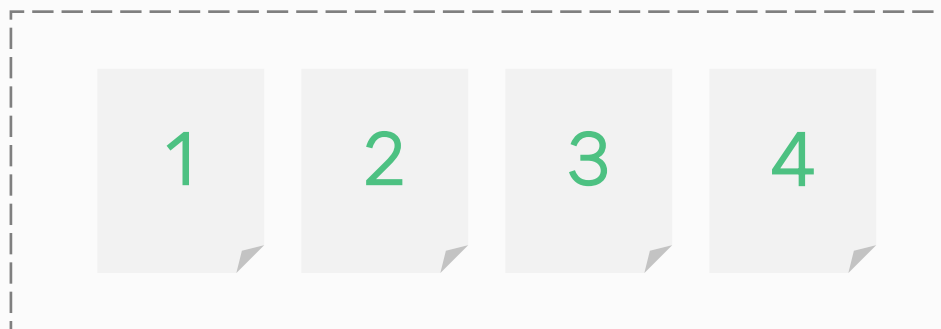


배포

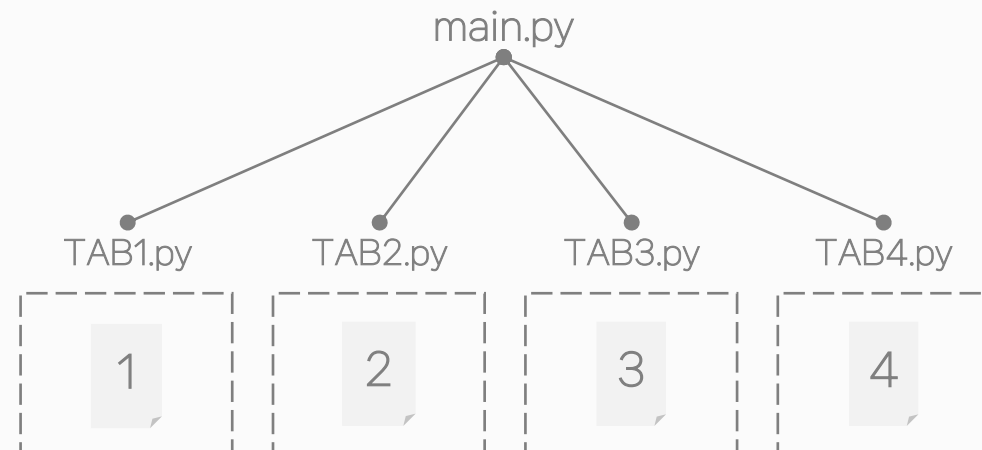
Streamlit 제작 방식

- ✓ Streamlit 파일을 메뉴 별로 분리하여 사용하면 가독성은 올라갈 수 있지만 유지 보수에 문제 발생 가능
- ✓ 원본 데이터가 1개인 점과 유지 보수의 편리성을 가져가기 위하여 1개의 streamlit 파일 생성

Streamlit.py



파일을 분리하지 않고 모든 Tab 관리



Main을 따로 두고 Tab별로 파일을 분리 할 경우

Streamlit

MENU

▷ 전체분포 확인

▷ 유해 봉환 확인

▷ 동원 지역

▷ 접수번호 조회

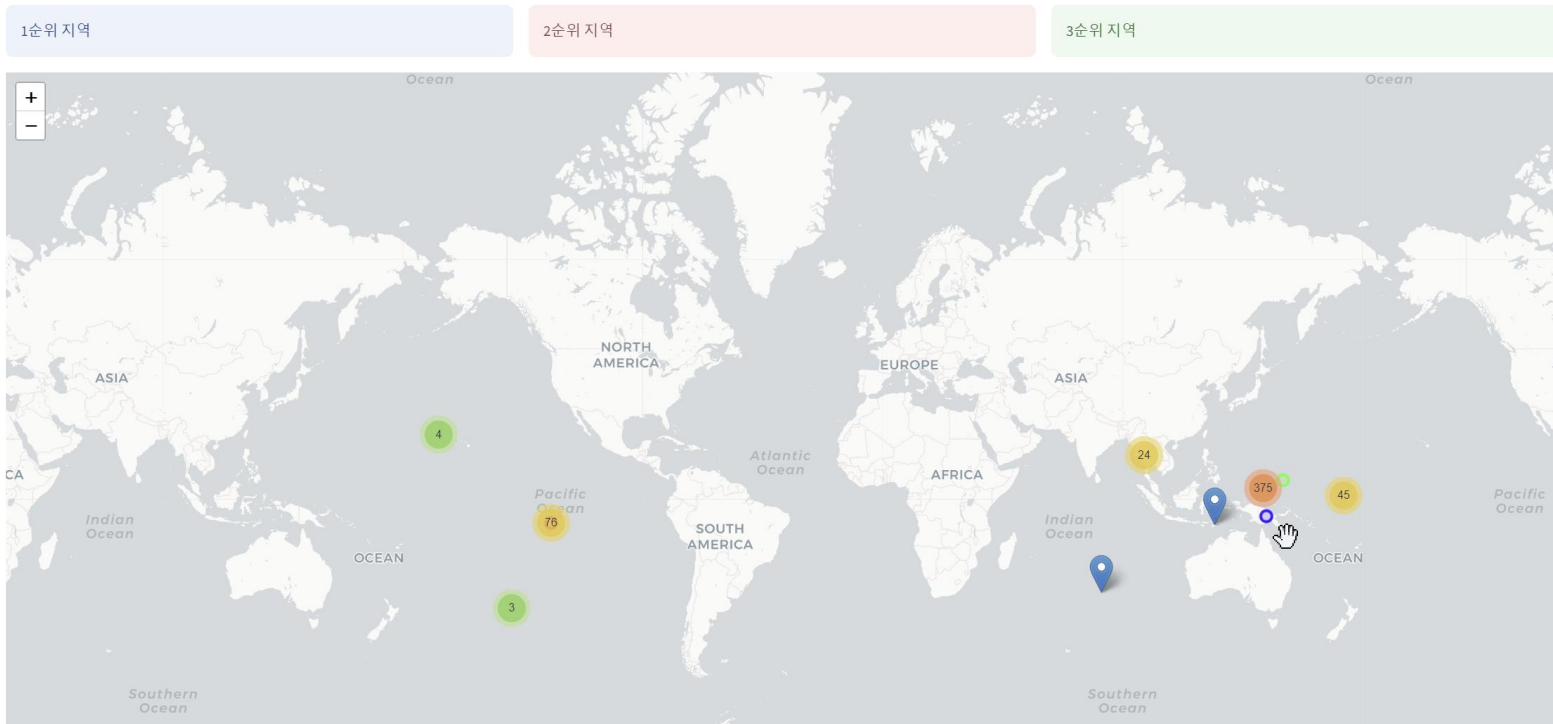
▷ 통합

지도 Layout

전체분포 확인 지도

작업장 구분 파이차트

전체 분포 확인 지도



전체분포 확인 지도

- ✓ 지역명을 기준으로 좌표 표시 및 클러스터링 되어 분포를 확인 가능
- ✓ 1,2,3 순위 지역은 원으로 표시

Streamlit

MENU

▷ 전체분포 확인

▷ 유해 봉환 확인

▷ 동원 지역

▷ 접수번호 조회

▷ 통합

지도 Layout

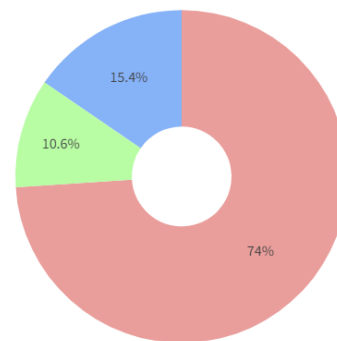
전체분포 확인 지도

작업장 구분 파이차트

작업장 육/해 구분 파이차트 ↗

육지 6095건 / 구분불가 874건 / 해양 1272건

■ 육지 ■ 해양 ■ 구분불가



| | 육해구분 | 육해구분완료 |
|---|--------------|--------|
| 0 | 제17방면군 경성사관구 | 육지 |
| 1 | 제15설영대 | 육지 |
| 2 | 제4시설부 | 육지 |
| 3 | 제4시설부 | 육지 |
| 4 | 야스쿠니마루(靖國丸) | 해양 |
| 5 | 제14방면군 | 육지 |
| 6 | 제4시설부 | 육지 |
| 7 | 확인불가 | 구분불가 |
| 8 | 제4시설부 | 육지 |
| 9 | 확인불가 | 구분불가 |

작업장 구분 파이차트

- ✓ 희생자 목록의 작업장을 기준으로 육지와 해양을 구분한 육지구분완료 데이터 파이차트 시각화
- ✓ 육지 / 해양 / 구분 불가로 이루어진 육해구분 파이차트

Streamlit

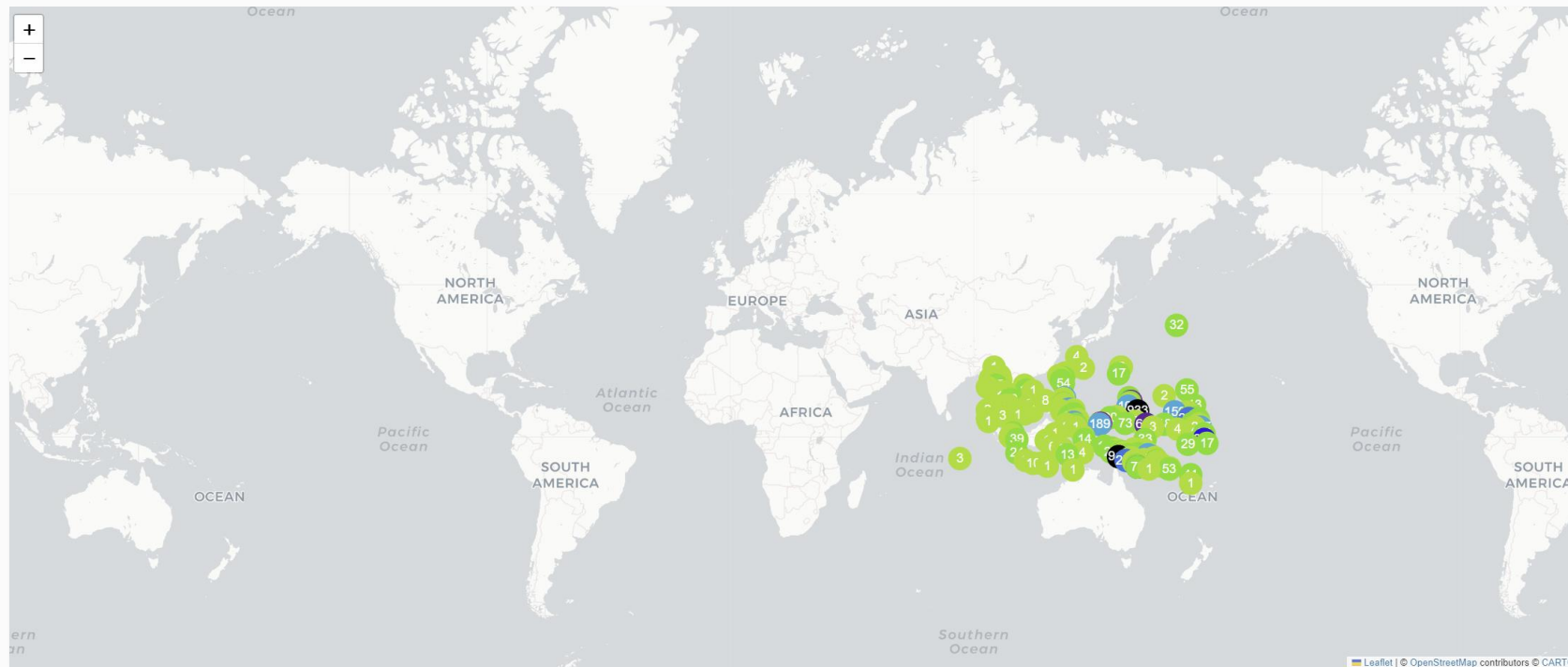
MENU

- ▷ 지도
- ▷ 유해 봉환 확인
- ▷ 동원 지역
- ▷ 접수번호 조회
- ▷ 통합

유해 봉환 확인 Layout

유해 봉환 확인 지도

유해 봉환 확인 파이차트



유해봉환 확인 지도

- ✓ 각 지역에 어느 정도의 희생자가 있는지 원에 숫자로 표시
- ✓ 유해봉환 수에 따른 색상 구분
(10이하, 100이하, 200이하, 300이하, 400이하, 500이하, 600이하, 700이하, 800이하, 그 외)
- ✓ 원 위에 마우스를 올리면 나라 이름, 총 합, 봉환, 불상, 미봉환 및 행불, 기타 수 표시



Streamlit

MENU

▷ 지도

▷ 유해 봉환 확인

▷ 동원 지역

▷ 접수번호 조회

▷ 통합

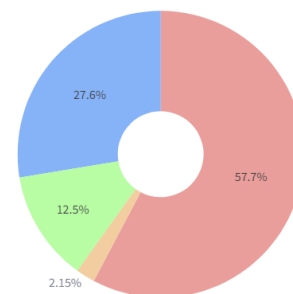
유해 봉환 확인 Layout

유해 봉환 확인 지도

유해 봉환 확인 파이차트

봉환여부 파이차트

■ 불상 ■ 미봉환 및 행불 ■ 봉환 ■ 기타



| | 유골봉환여부 | count |
|---|----------|-------|
| 0 | 불상 | 4,758 |
| 1 | 미봉환 및 행불 | 2,276 |
| 2 | 봉환 | 1,030 |
| 3 | 기타 | 177 |

유해 봉환 확인 파이차트

- ✓ 희생자 목록의 유골 봉환 여부 열의 데이터 파이차트 시각화
- ✓ 봉환, 불상, 미봉환 및 행불, 기타로 이루어져 있는 파이차트

Streamlit

MENU

- ▷ 지도
- ▷ 유해 봉환 확인
- ▷ **동원 지역**
- ▷ 접수번호 조회
- ▷ 통합

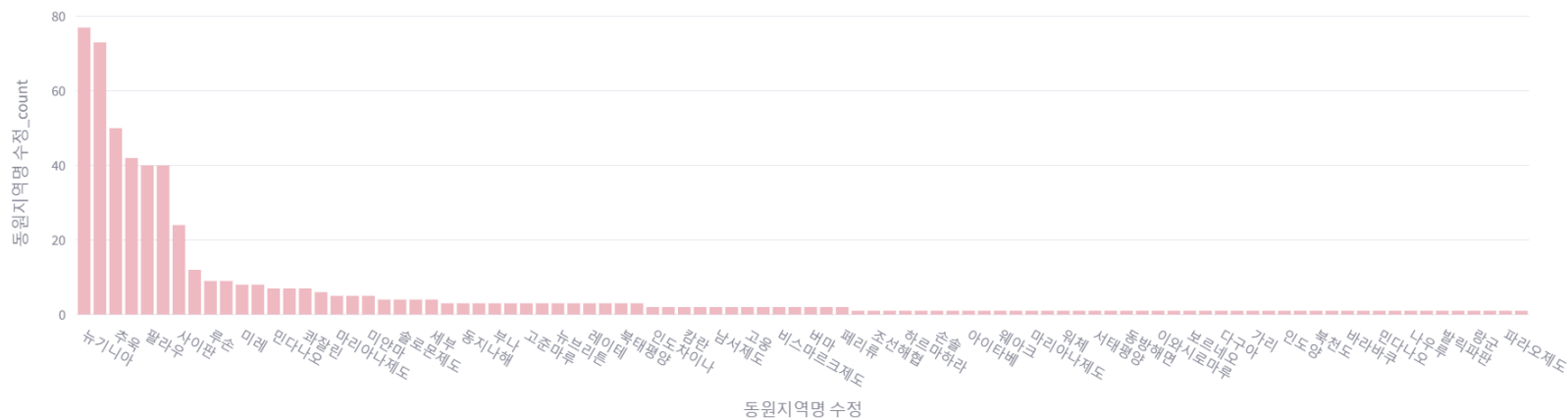
동원 지역 Layout

전체 동원 지역 그래프

선택 동원 지역 그래프

상위 동원 지역 그래프

전체 동원 지역 그래프



전체 동원 지역 그래프

- ✓ 희생자 목록의 동원지역명 수정 열 막대그래프 시각화
- ✓ 동원지역명을 기준으로 개수를 내림차순 정렬한 그래프

04 과제 수행 방법

Streamlit

MENU

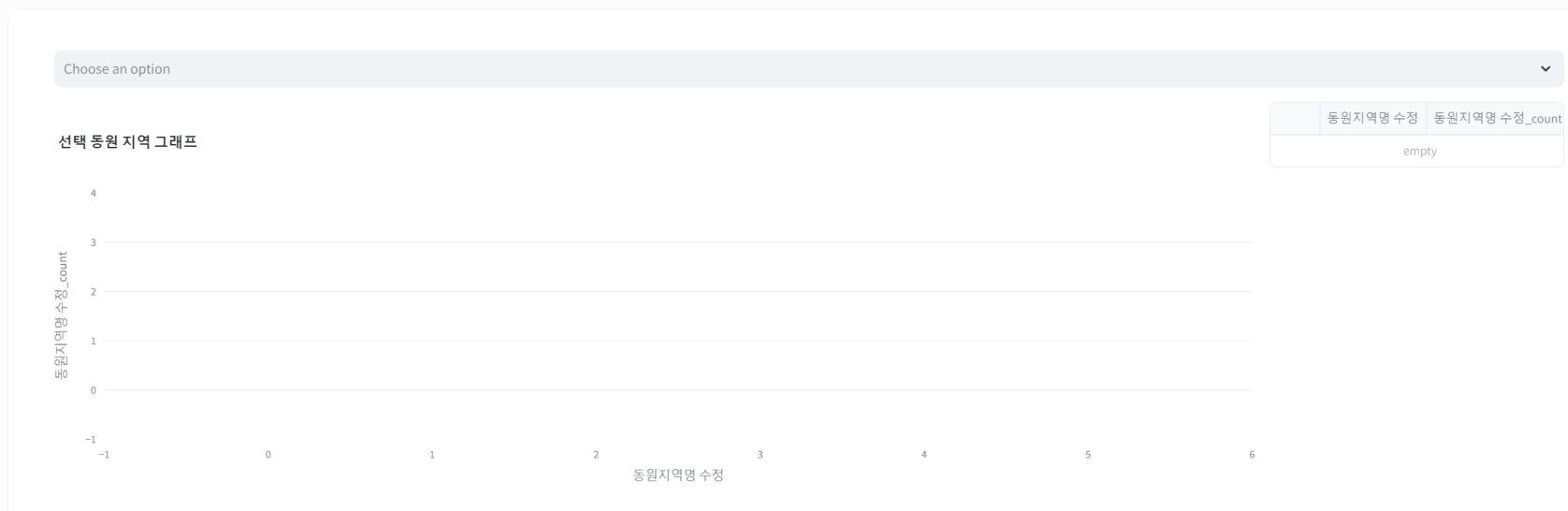
- ▷ 지도
- ▷ 유해 봉환 확인
- ▷ **동원 지역**
- ▷ 접수번호 조회
- ▷ 통합

동원 지역 Layout

전체 동원 지역 그래프

선택 동원 지역 그래프

상위 동원 지역 그래프



선택 동원 지역 그래프

- ✓ 희생자 목록의 동원지역명 수정 열 막대그래프 시각화
- ✓ Multi select를 사용하여 확인하고 싶은 지역 선택, 선택한 지역만 확인할 수 있는 그래프

Streamlit

MENU

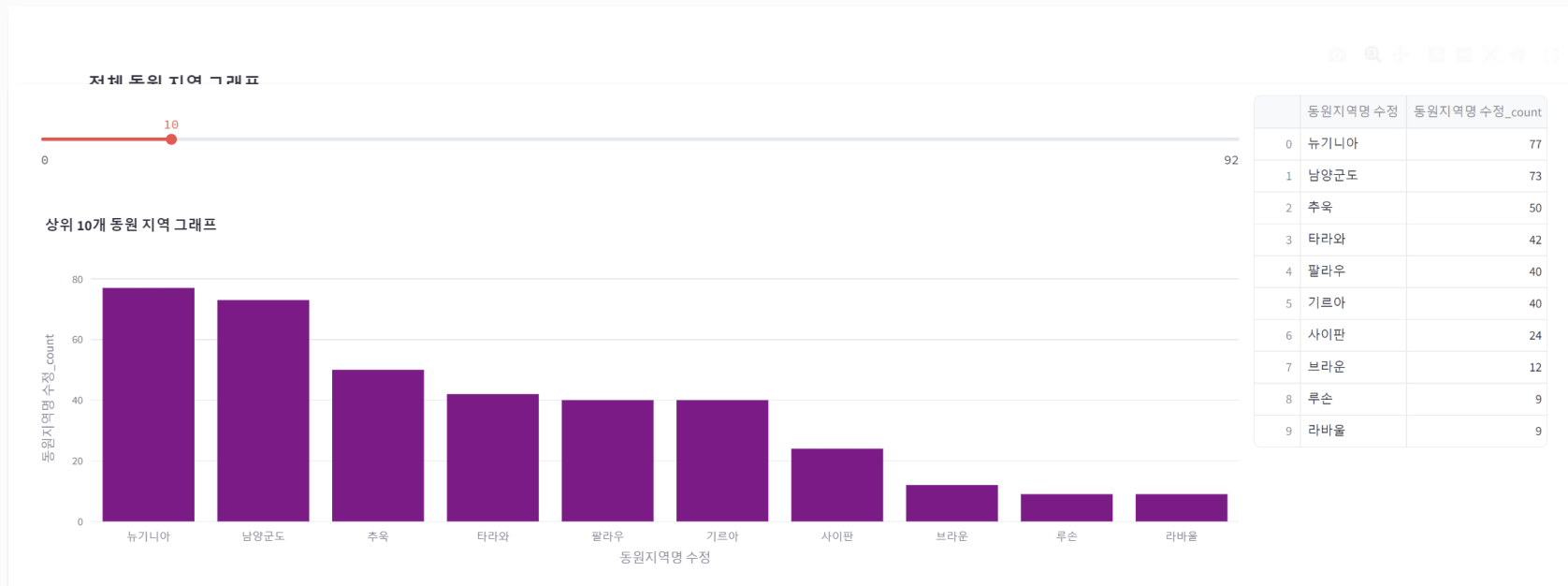
- ▷ 지도
- ▷ 유해 봉환 확인
- ▷ **동원 지역**
- ▷ 접수번호 조회
- ▷ 통합

동원 지역 Layout

전체 동원 지역 그래프

선택 동원 지역 그래프

상위 동원 지역 그래프



상위 동원 지역 그래프

- ✓ 희생자 목록의 동원지역명 수정 열 막대그래프 시각화
- ✓ slider를 이용하여 보고싶은 상위 개수를 선택하고 상위 동원지역을 확인 할 수 있는 그래프

04 과제 수행 방법

Streamlit

MENU

- ▷ 지도
- ▷ 유해 봉환 확인
- ▷ 동원 지역
- ▷ **접수번호 조회**
- ▷ 통합

접수번호 조회Layout

접수번호 조회

접수번호를 입력하세요

접수번호 조회

- ✓ 접수번호 (ex. 통영시-000)를 입력하면 접수번호에 해당되는 데이터와 해당되는 지역을 지도에 출력

Streamlit

MENU

- ▷ 지도
- ▷ 유해 봉환 확인
- ▷ 동원 지역
- ▷ 접수번호 조회

▷ 통합

통합 Layout

동원 국가 선택

동원 지역 선택

선택 지역 지도 출력

그래프

데이터프레임

엑셀 다운로드

지도(html)다운로드

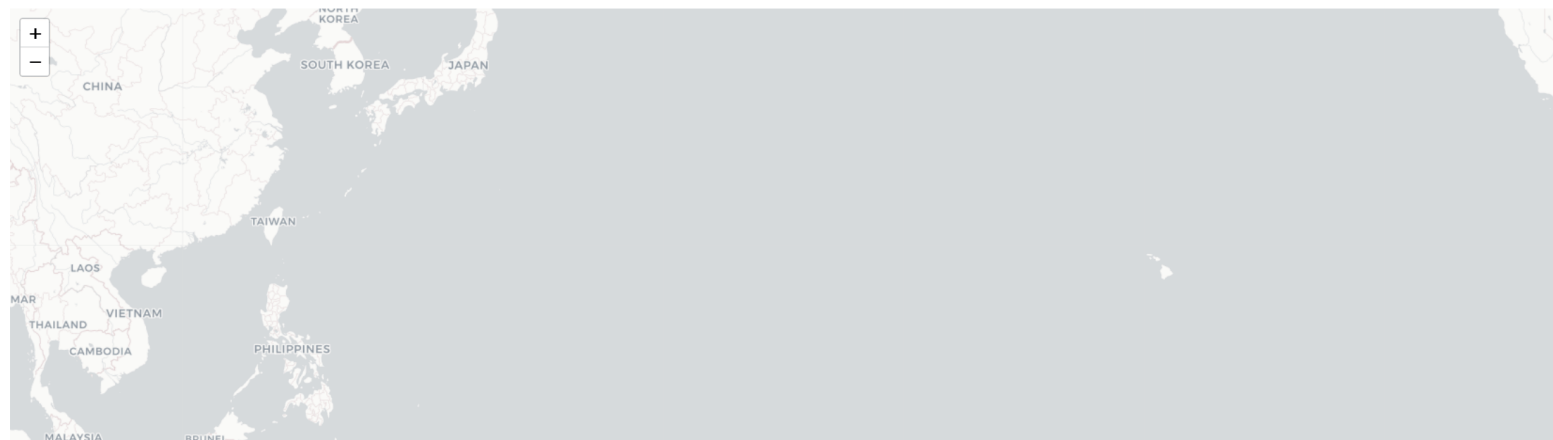
통합분석

동원국가 선택

Choose an option

동원지역 선택

No options to select.



동원국가 선택

확인하고 싶은 국가 선택

동원지역 선택

선택한 국가의 동원지역명만 선택할 수 있으며 확인 하고 싶은 지역만 선택

선택한 지역의 데이터 출력

선택한 지역의 데이터만 지도, 그래프, 데이터프레임으로 출력

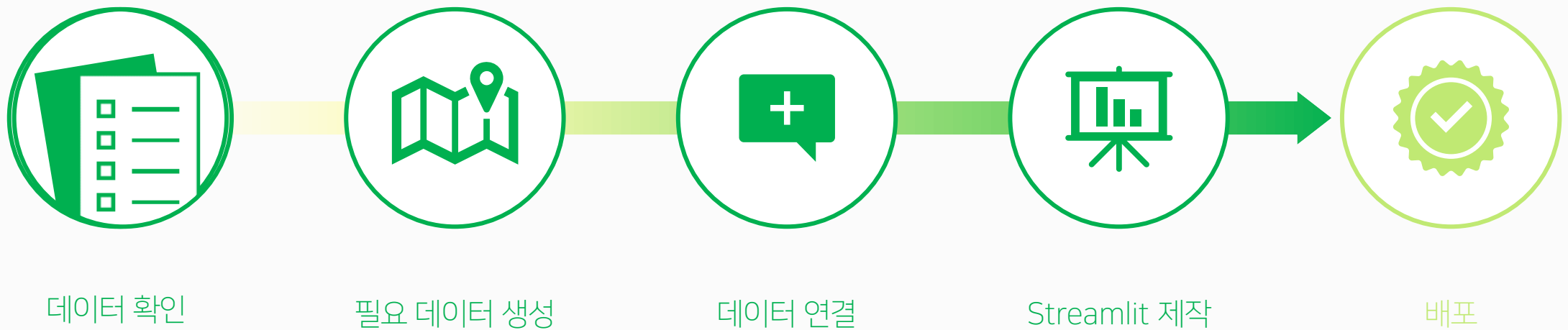
엑셀다운로드 버튼

데이터프레임을 바로 저장할 수 있지만 csv로 저장되므로 엑셀로 변환 다운로드 버튼 배치

지도(html)다운로드 버튼

선택한 지역의 지도를 다운로드 할 수 있는 버튼 배치

과제 흐름



Streamlit 배포 방법

✓ 작성된 Streamlit 코드는 깃허브를 통하여 업로드 후 Streamlit 사이트에 연결 후 배포 가능



Streamlit

✓ Streamlit코드 작성

0. streamlit_데이터(날짜).py 생성

✓ 저장소

Requirements.txt

Streamlit_geo.py

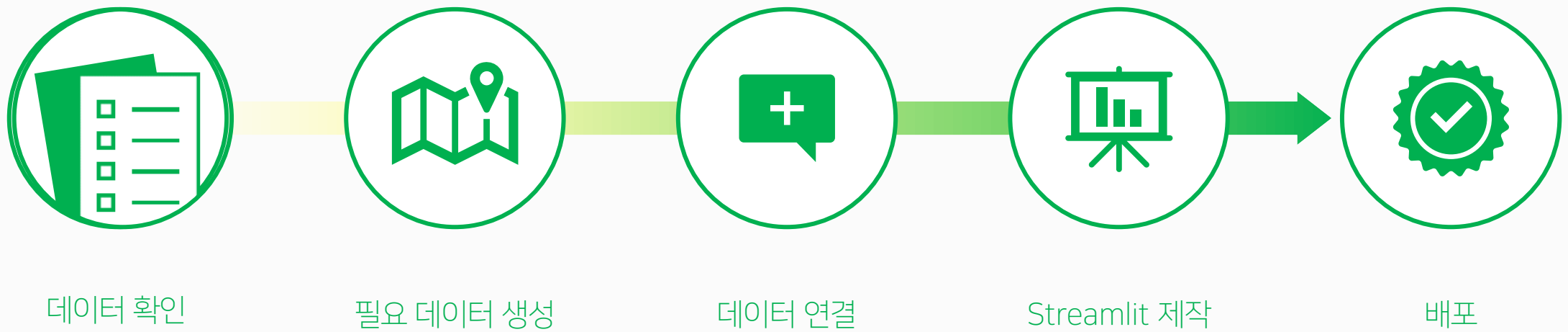
0. streamlit_데이터(날짜).py

✓ 코드 변환과 주소생성

깃허브와 streamlit share 연결

Streamlit 리부트, 랜더링 수행

과제 흐름



05

유지 보수

유지 보수

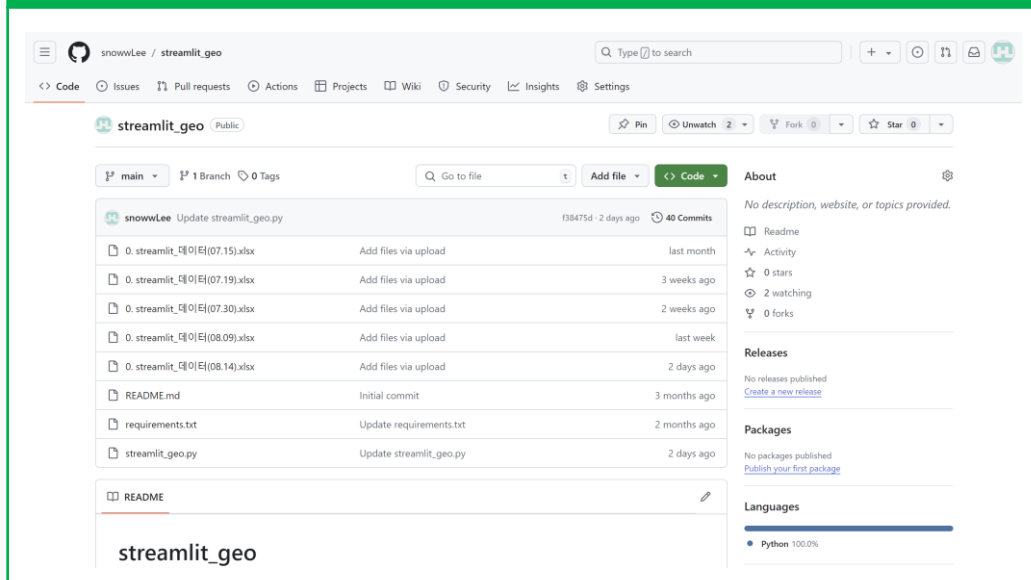
- ✓ 저장 데이터, 코드 변경시에는 Github
- ✓ Streamlit 사용중 오류 발생시는 Streamlit 사이트에서 변경
- ✓ Github저장소와 Streamlit 사이트의 아이디 소유자 → 과거사관련업무지원단 대일항쟁기강제동원피해지원과
- ✓ 로그인시 인증 필요하므로 사전 연락 필요함



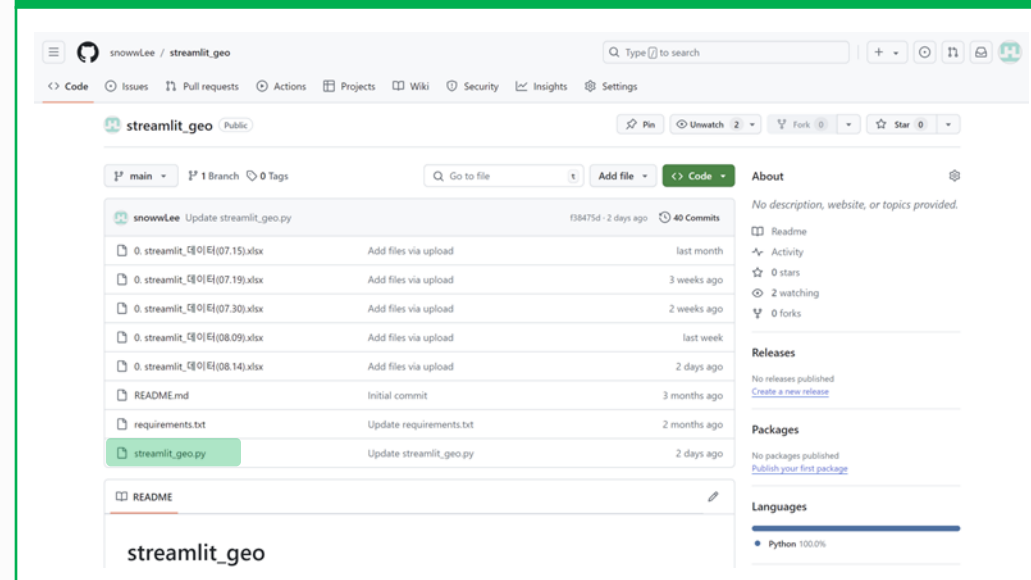
유지 보수

- ✓ Streamlit code 변경시 _ 방법 1 (코드를 복사하여 부분 업데이트 할 경우)

1. Streamlit 파일이 올라간 Github 접속



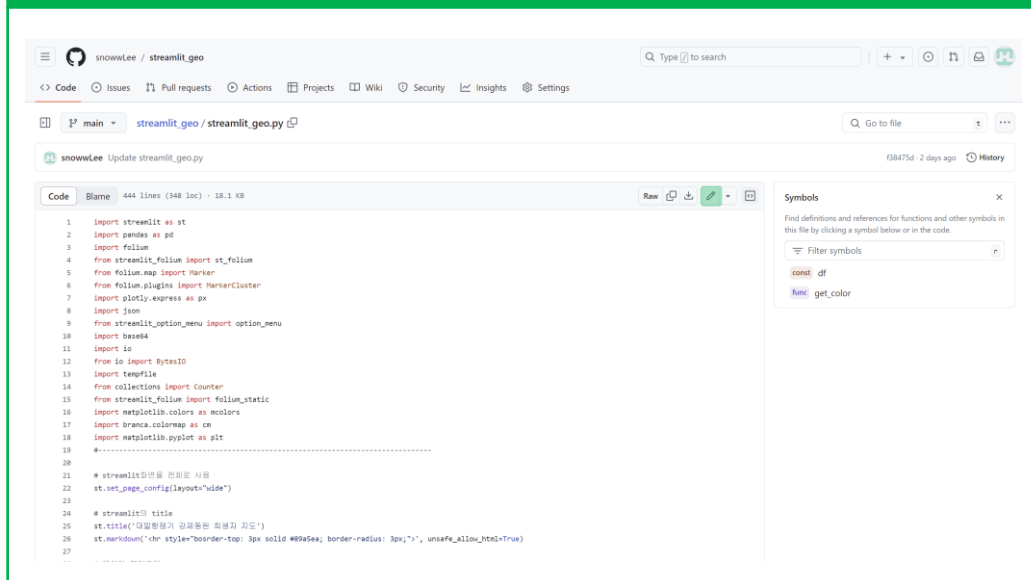
2. streamlit_geo.py



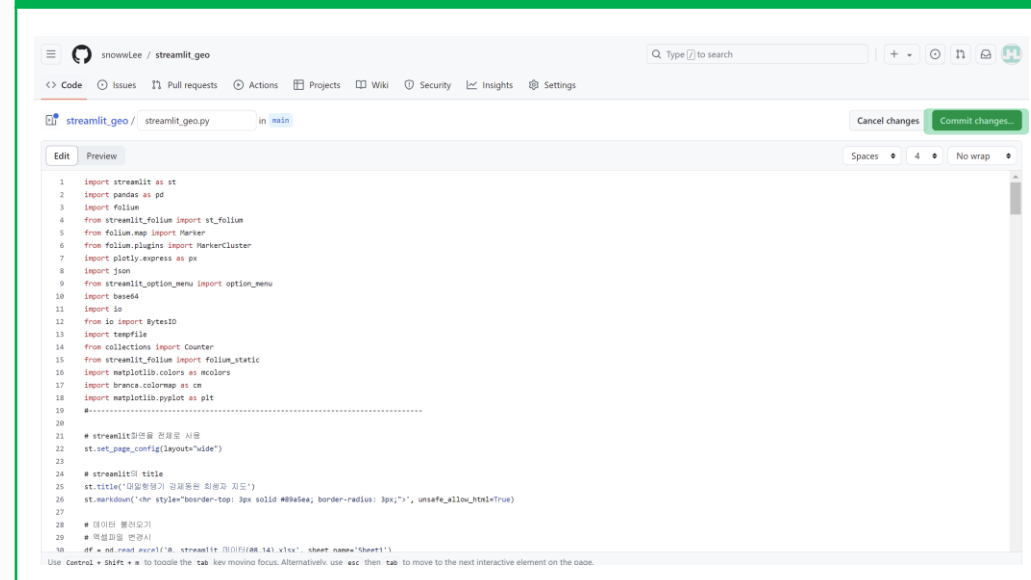
유지 보수

- ✓ Streamlit code 변경시 _ 방법 1 (코드를 복사하여 부분 업데이트 할 경우)

3. Edit this file 클릭



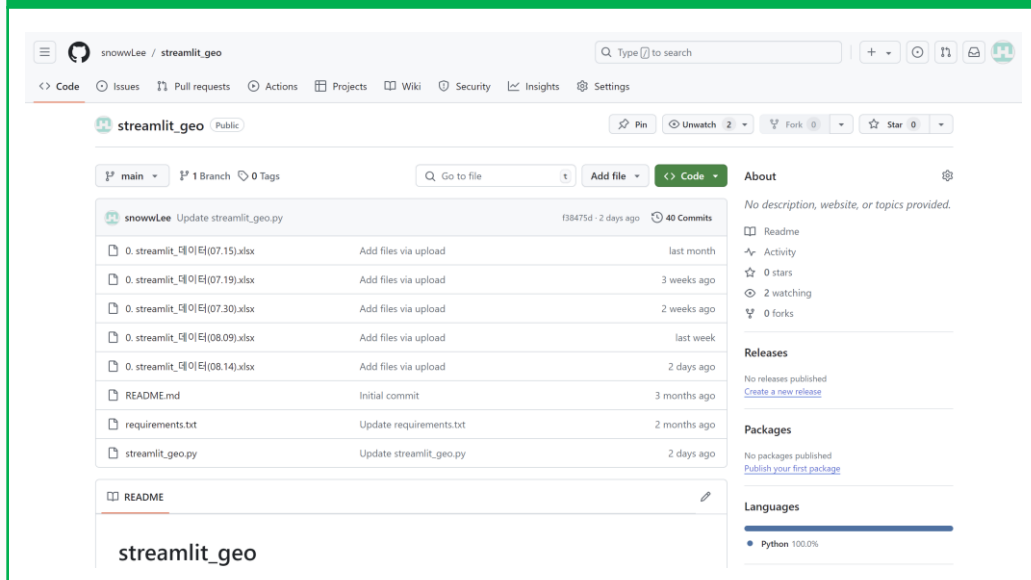
4. 내용 변경 후 Commit change 클릭



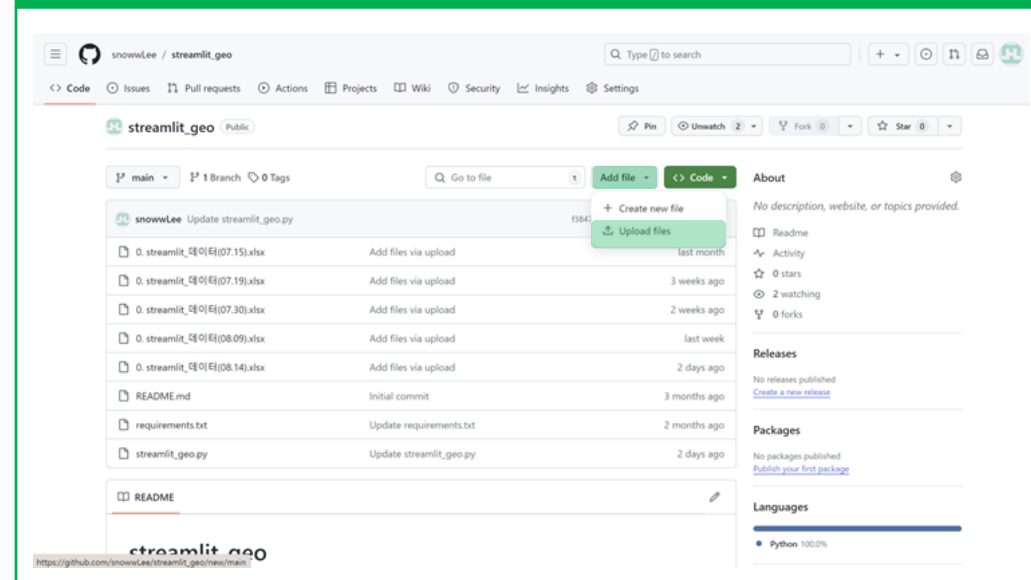
유지 보수

- ✓ Streamlit code 변경시 _ 방법 2 (파일 자체를 업데이트 할 경우)
 - * 파일 이름은 동일해야함

1. Streamlit 파일이 올라간 Github 접속



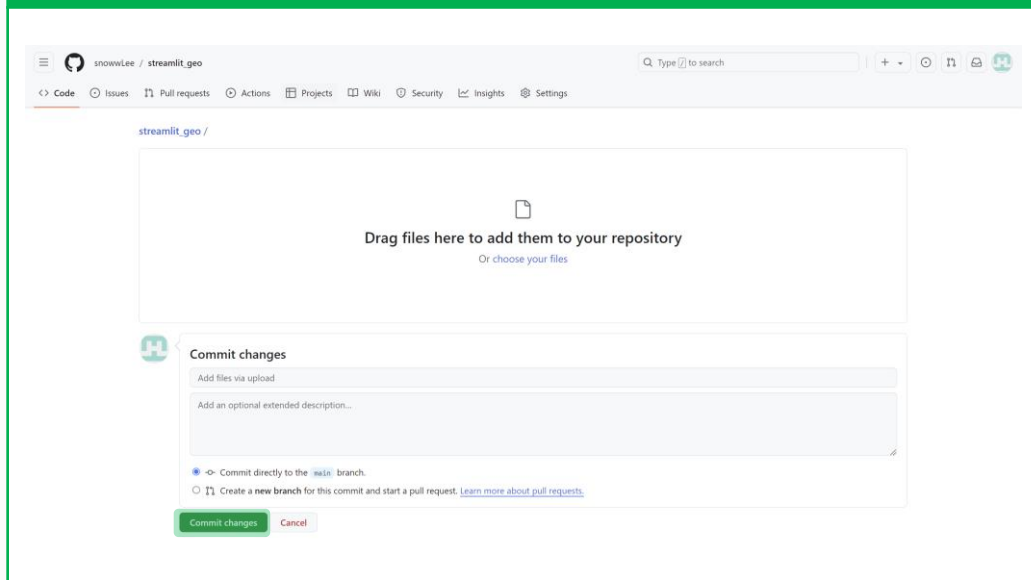
2. Add file – Upload files 클릭



유지 보수

- ✓ Streamlit code 변경시 _ 방법 2 (파일 자체를 업데이트 할 경우)
 - * 파일 이름은 동일해야함

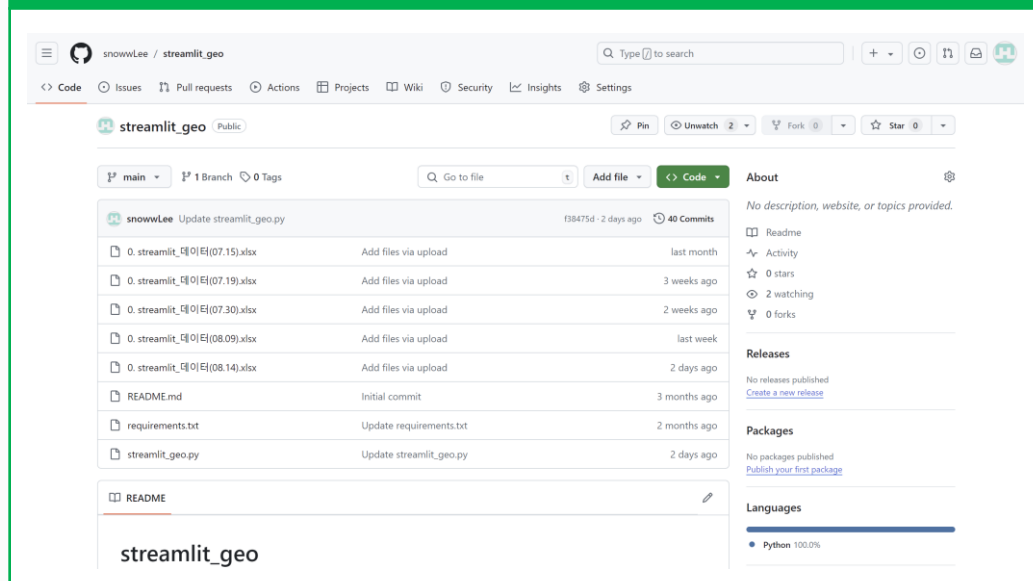
3. 변경하려는 파일 업로드 후 Commit Change



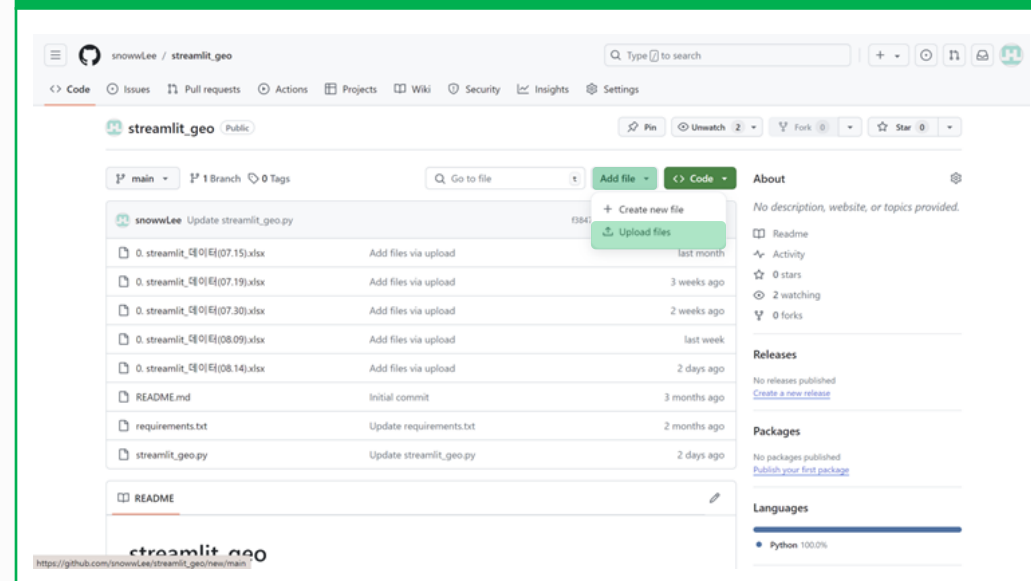
유지 보수

✓ 엑셀 데이터 변경시

1. Streamlit 파일이 올라간 Github 접속



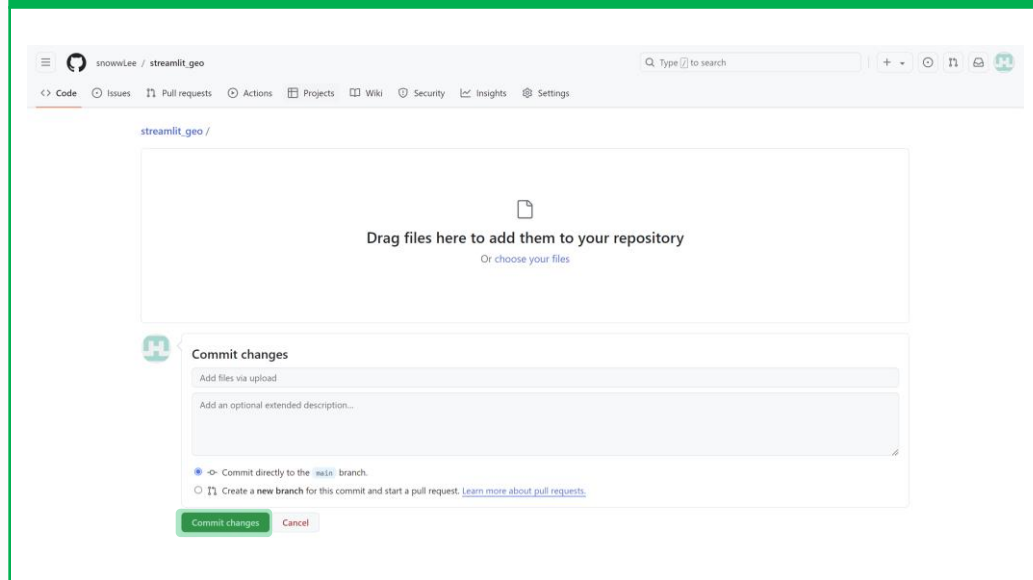
2. Add file – Upload files 클릭



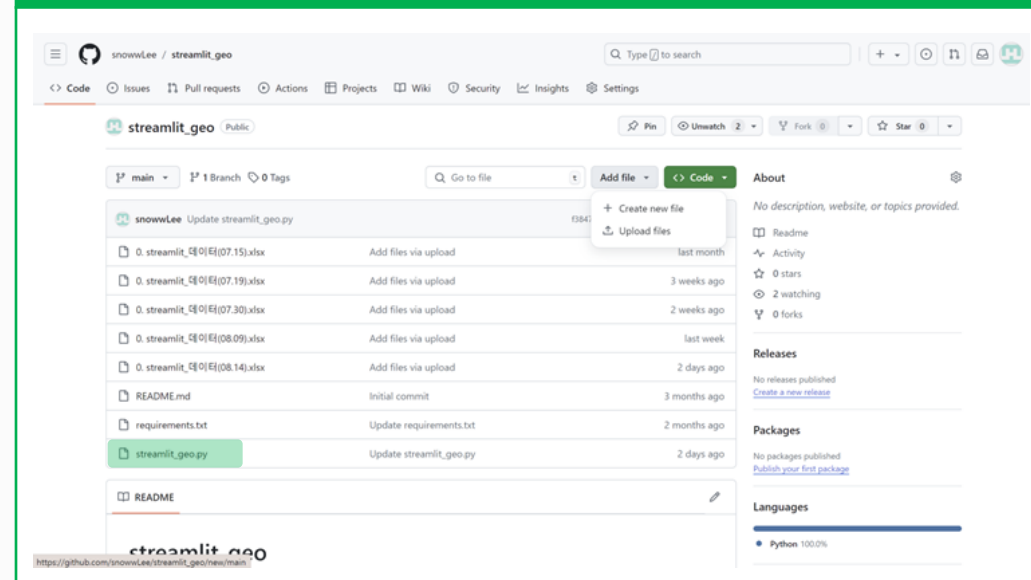
유지 보수

✓ 엑셀 데이터 변경시

3. 엑셀 파일 업로드 후 Commit Change



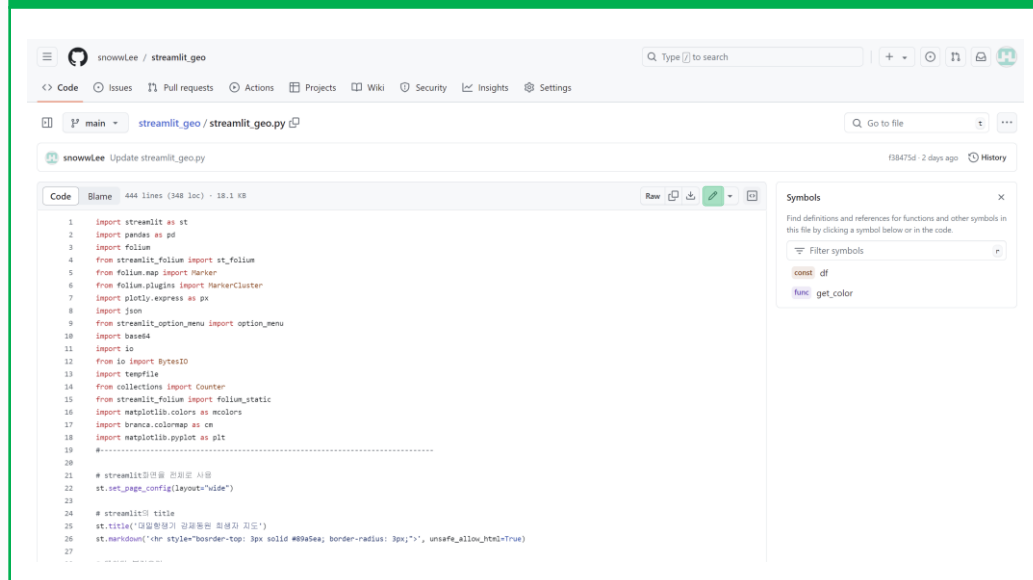
4. streamlit_geo.py 접속



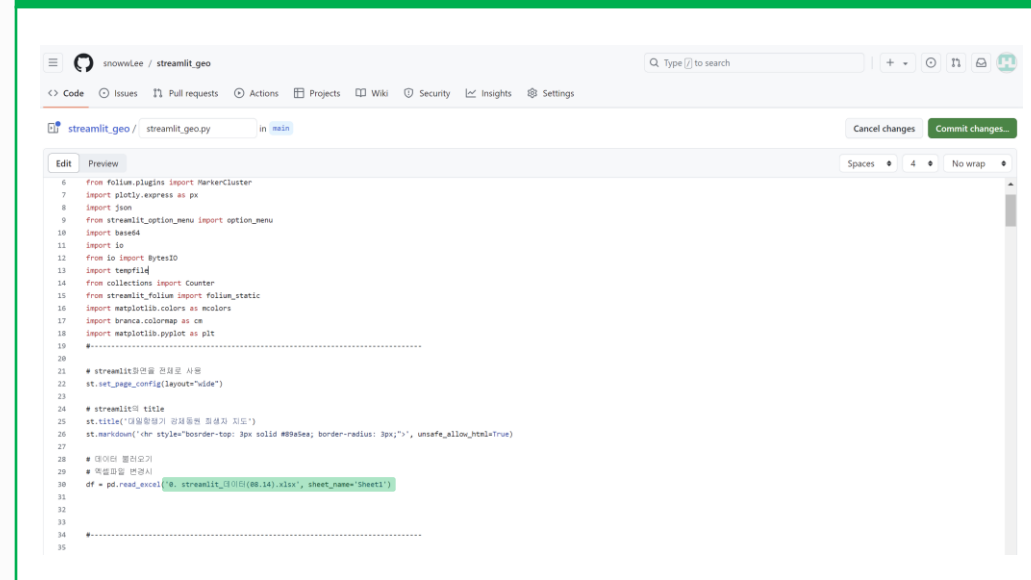
유지 보수

✓ 엑셀 데이터 변경시

5. Edit this file 클릭



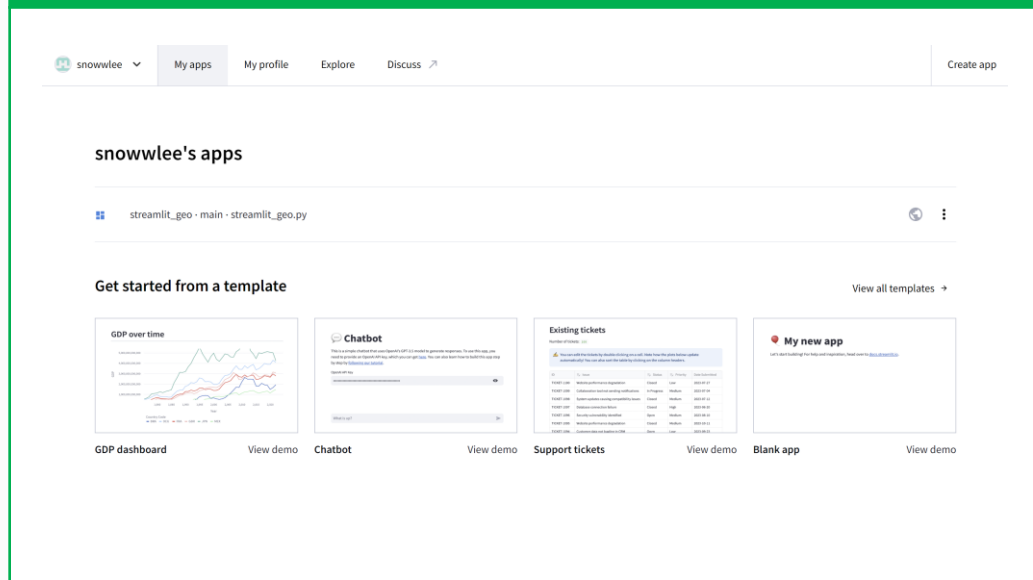
6. 엑셀 파일명 변경 후 Commit Change



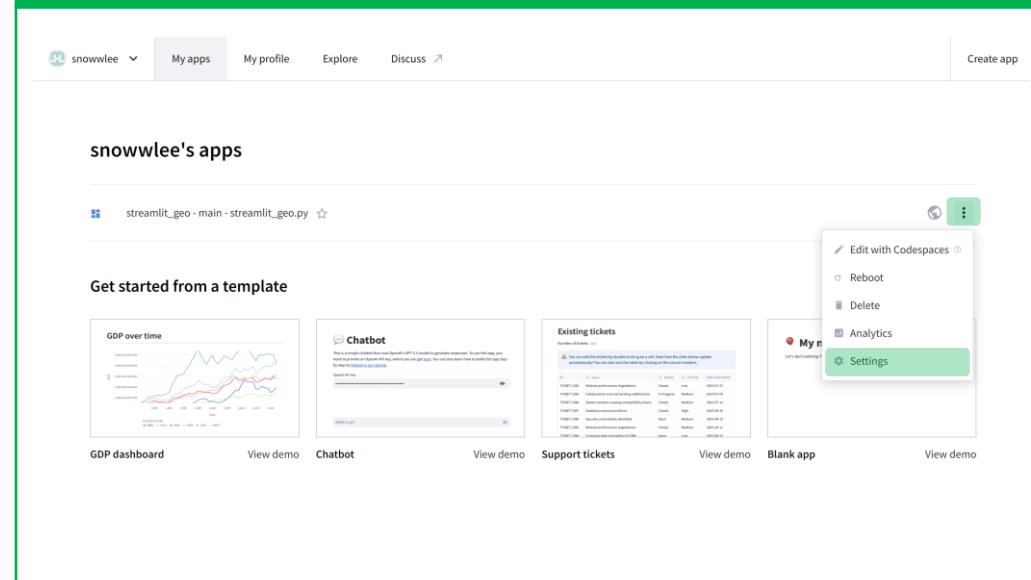
유지 보수

- ✓ Streamlit 접속 URL 변경시

1. Streamlit 사이트 접속 후 github 계정으로 로그인



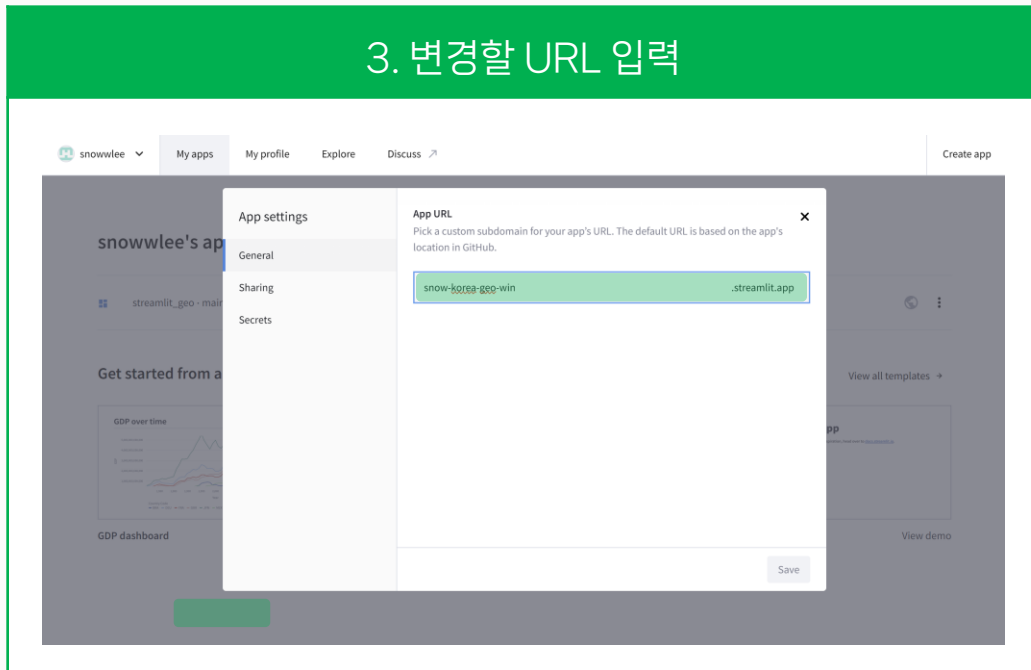
2. My Apps 내 streamlit_geo 우측 setting 클릭



유지 보수

- ✓ Streamlit URL 변경시

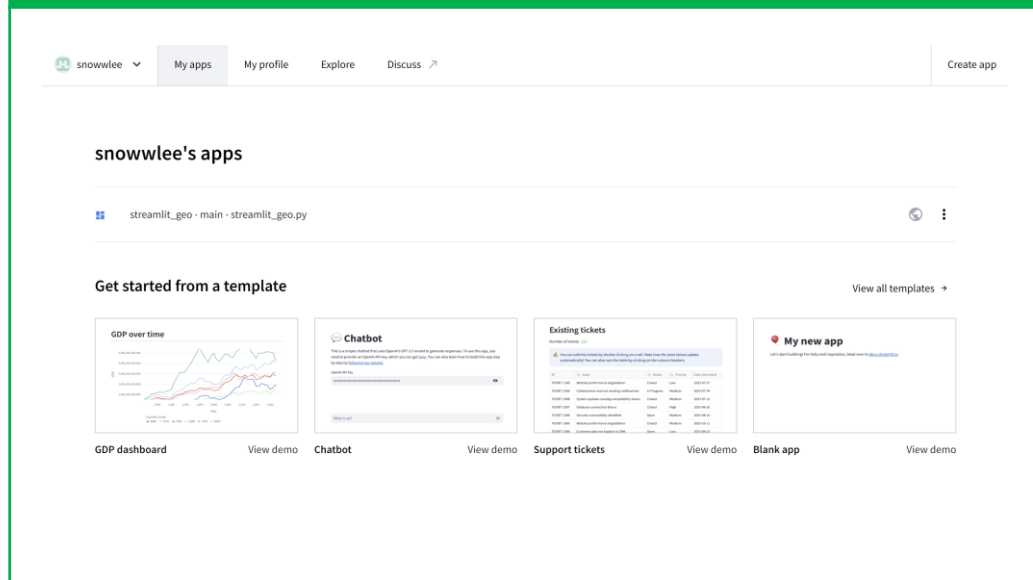
3. 변경할 URL 입력



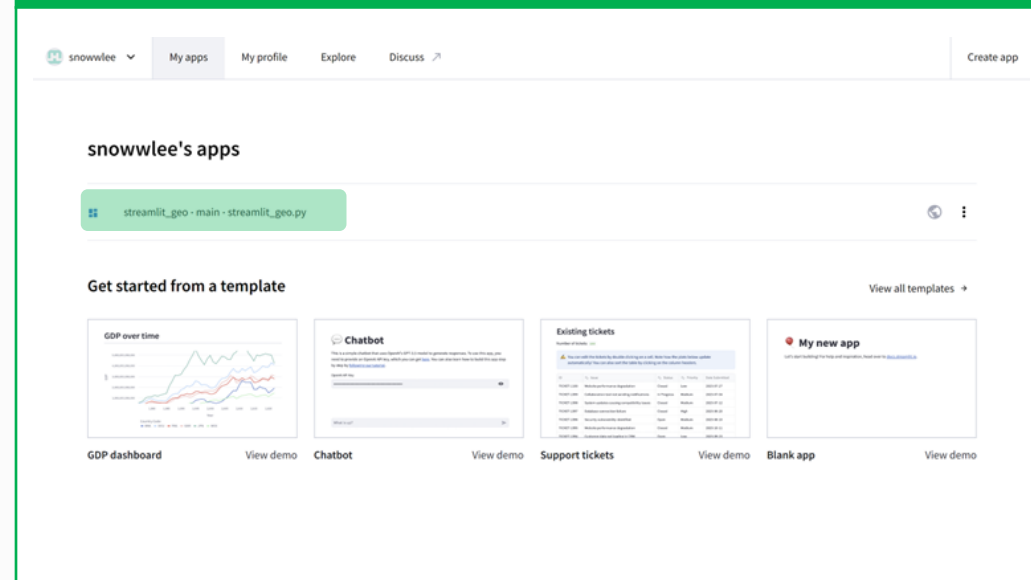
유지 보수

- ✓ Streamlit sleep 상태 혹은 오류로 재부팅 필요시

1. Streamlit 사이트 접속 후 github 계정으로 로그인



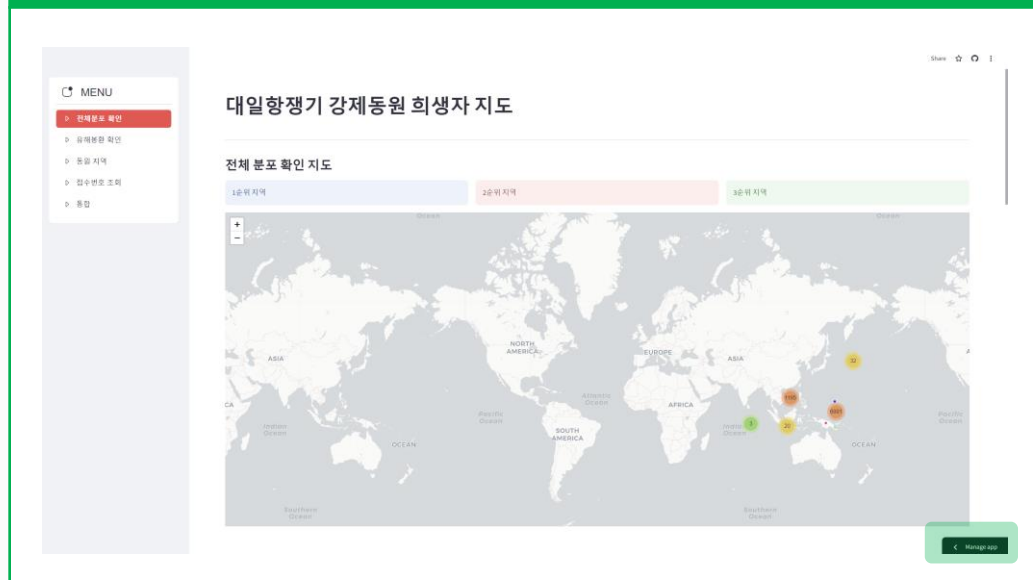
2. Streamlit 접속



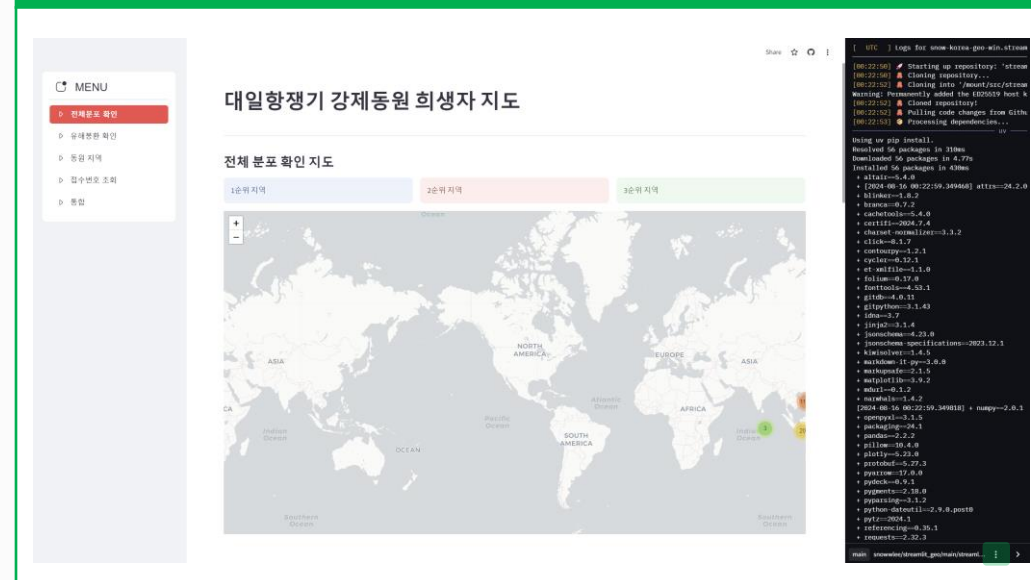
유지 보수

- ✓ Streamlit sleep 상태 혹은 오류로 재부팅 필요시

3. 우측 하단 Manage app 클릭



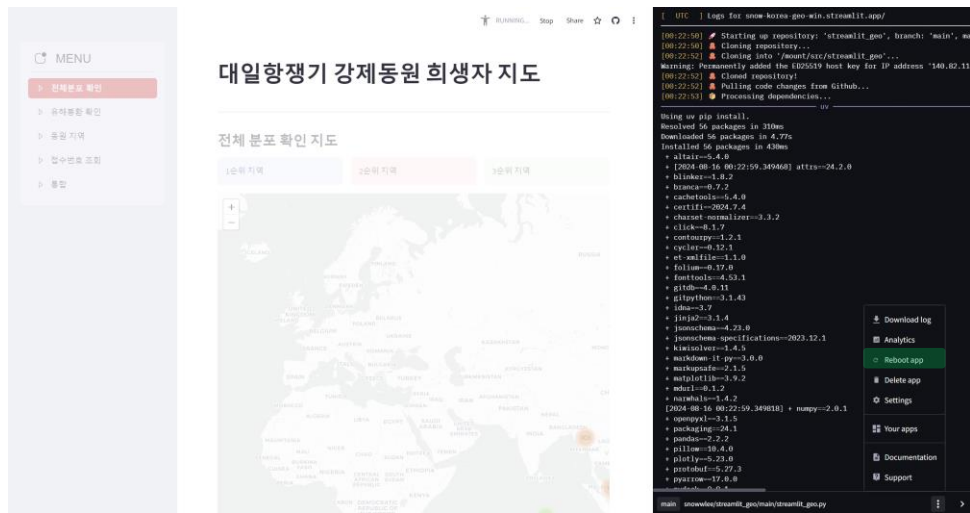
4. ⋮ 버튼 클릭



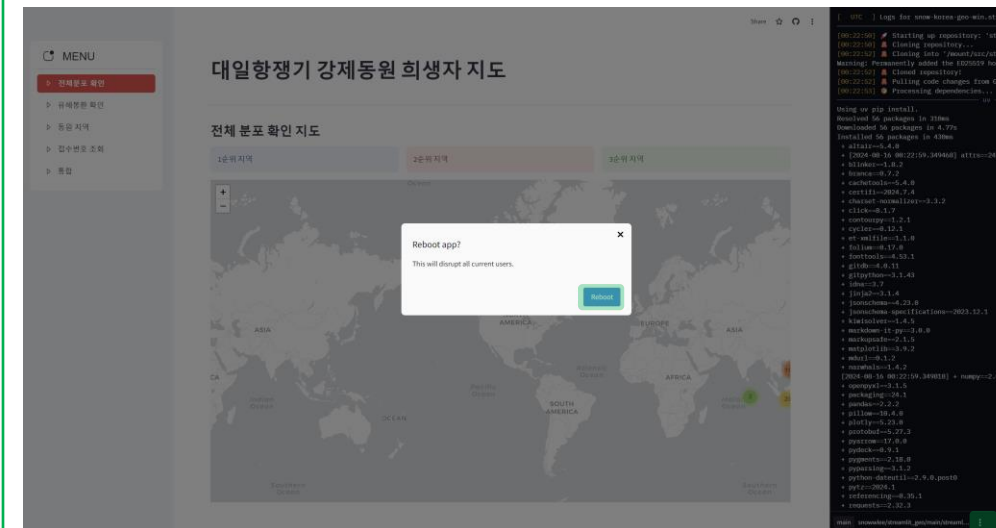
유지 보수

- ✓ Streamlit sleep 상태 혹은 오류로 재부팅 필요시

5. Reboot app 클릭



6. Reboot 클릭



감사합니다