

# Project Walkthrough: Fair Maps Redistricting Tool

This document provides a complete, step-by-step guide to the Congressional Redistricting Analysis project, summarizing our goals, the final working code, and the correct procedures for setup and execution.

## 1. Project Goal & Methodology

The primary goal of this project is to create and analyze fair, non-partisan congressional district maps for any state in the U.S.

- **Methodology:** We use a powerful technique called **Ensemble Analysis**. Instead of trying to find a single "perfect" map, our Python script generates thousands of different, valid maps using a **Markov Chain Monte Carlo (MCMC)** algorithm. This process is "blind" to any political or racial data, focusing only on two core legal requirements:
  1. **Equal Population:** All districts must have nearly the same number of people.
  2. **Contiguity:** All parts of a district must be geographically connected.
- **Final Map:** After creating this large "ensemble" of possible maps, the script analyzes all of them to find the single, real map from the simulation that is the most representative and has the most compact, well-formed districts. This final map serves as a neutral baseline to compare against existing district plans.

## 2. Required Tools

- A Linux environment (like the Crostini you are using)
- Python 3 and the pip package installer
- The venv module for creating virtual environments
- ImageMagick, a command-line tool for merging the final map images

## 3. Complete Setup Instructions

This is the definitive, step-by-step guide to setting up the project from a clean start.

### Step A: Set Up the Project Folder and Environment

First, open your Linux terminal.

1. **Create the main project folder** and navigate into it:

```
mkdir redistricting_project
cd redistricting_project
```
2. **Create a Python virtual environment.** This is a critical step that keeps our project's software separate from the rest of your system.

```
python3 -m venv venv
```
3. **Activate the virtual environment.** You must run this command every time you open a new terminal to work on this project.

```
source venv/bin/activate
```

Your terminal prompt will now start with (venv), which shows the environment is active.

4. **Install the necessary Python libraries** into your active environment:

```
pip install gerrychain geopandas pandas matplotlib networkx
```

5. **Install ImageMagick.** This is a Linux tool our master script will use to stitch the final state maps together.

```
sudo apt-get update
```

```
sudo apt-get install imagemagick
```

## Step B: Get the Data (The Correct Way)

This project requires geographic map files (**shapefiles**) that have been cleaned and merged with official 2020 population data. The most reliable source for this is the **Voting and Election Science Team (VEST) Archive at Harvard University**.

1. **Create the main data directory** inside your redistricting\_project folder:

```
mkdir data
```

2. **For each state you want to analyze, you will manually download the data.** This bypasses all the server issues we encountered.

- o **Go to the Harvard Dataverse Archive:**

<https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/K7760H>

- o Find the state you want in the list (e.g., ca\_2020.zip for California).
- o Click the **Download** button and select the "**Original File Format (.zip)**" option.
- o Move the downloaded file from your ~/Downloads folder into a new, state-specific folder within your data directory and unzip it. For example, for California:

```
# Create the folder for California
```

```
mkdir data/CA
```

```
# Move the downloaded file
```

```
mv ~/Downloads/ca_2020.zip data/CA/
```

```
# Go into the folder and unzip the file
```

```
cd data/CA
```

```
unzip ca_2020.zip
```

```
cd ../.. # Go back to the main redistricting_project directory
```

3. Repeat this manual download-and-unzip process for every state you want to include in your nationwide analysis.

## 4. How to Run the Nationwide Analysis

Once your setup is complete and you have the data for at least one state, you can run the

entire analysis. The project uses two scripts that work together.

- `analyze_maps.py`: The core Python engine that analyzes a single state.
- `run_all_states.sh`: The master script that automatically runs the Python engine for every state you have data for.

**You only need to run the master script.**

1. **Make the master script executable.** This is a one-time command.  
`chmod +x run_all_states.sh`
2. **Fix the script's file format.** This is a one-time command to prevent errors in Linux.  
`sed -i 's/\r$//' run_all_states.sh`
3. **Run the full analysis!**  
`./run_all_states.sh`

The script will now find every state folder in your data directory, run the complete MCMC simulation for each one, and save the output maps.

## 5. Understanding the Output

The script will create a new folder called `output_maps`. Inside, you will find two files for each state that was successfully processed (e.g., for California):

1. `CA_summary_map.png`: A high-quality image of the final, fair map for that state.
2. `CA_best_map.geojson`: The raw geographic data for the map. This file is not an image; it's a data file that you can open in specialized mapping software like the free and open-source **QGIS** for more detailed inspection.

At the very end, the script will also try to create a single `USA_master_map.png` in your main project folder by stitching together all the individual state images.