Shida Nyimbili

B00814941

EGM722 Assignment Part 1

Programming for GIS and Remote Sensing

Introduction

In order to achieve this, the Ministry of Health plans to expand healthcare access in rural areas. This is to enable people on HIV medication have their viral load monitored and suppressed. The ministry plans to construct more health facilities especially in rural areas and also deploy GeneXpert machines in the already existing health facilities (Brooke, et al., 2018).

The aim of this project is to determine the mean distance HIV patients currently travel to access healthcare services in the rural parts of Zambia.

This Python code performs spatial analysis on healthcare data in rural Zambia. It starts by importing the necessary libraries and reading in shapefiles containing patient data, health facilities, and roads. The data is projected using the UTM projection. Next, a buffer of 5km is created around the health facilities and patients within the buffer are spatially joined. The code then calculates the mean distance from each patient to the nearest health facility and determines how many patients within and outside the 5km buffer have their viral load suppressed or unsuppressed. Patients outside the buffer are spatially joined with health facilities and their distance to the nearest facility is calculated. Finally, the results are plotted on a map that shows the roads, health facilities, and patients.

Setup and installation

Download my repository at my Github.

You can find my repository at the following link: https://github.com/shidaNyimbili/EGM722-Project

1. Getting Started

Downloading git, GitHub desktop, and Anaconda on your PC is the first thing you must do. Download here GitHub Desktop, Anaconda, and Git .

2. Installation/clone this repository

Once all the programs above are installed, you need a GitHub account, if you already have then you can clone this repository of this project to your computer by performing the following steps:

1. Visit the gitHub repository for the project's materials at https://github.com/shidaNyimbili/EGM722-Project . Click the "fork" button located in the top right corner.
   Make sure to uncheck everything on the new page that will pop up after clicking it. Click 'Create fork' only after copying the main branch. This will copy the entire repository to your account and establish a fork of the repository.
2. Open GitHub Desktop and select **File > Clone Repository**. Select the **URL** tab, then enter the URL for this repository.
   1. You should also see your repository listed under **Your repositories**. Click on **Clone a repository from the Internet**, and you should see your forked version of

the **EGM722-Project** repository, Select a local path to save the repository and click "clone". The report is now cloned, for further guidance , see here

Installing & Setting Up Conda / Anaconda

1. After completing the installation process, proceed to set up your Conda environment.
2. The creation of an environment in Anaconda requires using the yml file found in the cloned repository. To begin, open Anaconda Navigator by clicking on the Windows icon and searching for "Anaconda."
3. Once Anaconda is open, navigate to the Environments tab located in the left-hand menu.
4. Next, select "Import" from the bottom menu bar, browse to the location of the cloned repository folder, select the environment.yml file, and click "Import." The yml file contains a list of channels from which to install packages and dependencies that the code relies on. This process may take some time, and once completed, the newly created environment will appear in the environment tab of Anaconda.
5. To select the environment you just created, go to the Home tab in Anaconda, where you will notice two dropdown boxes. The first box allows you to filter the displayed applications, while the second allows you to choose the environment you would like to use. Select your newly created environment from the second dropdown.
   For further guidance see here

4. Run Code/Starting Jupyter notebook

You can launch Jupyter Notebook from Anaconda Navigator and access your project folder by following these steps:

1. Open the Command Prompt from Anaconda Navigator, ensuring that your environment is active or selected.

2. Navigate to the folder where you have saved the repository and open the Jupyter file.

Requirements / Dependencies

- Jupyter
- Anaconda
- Chrome or similar browser
- GitHub Desktop
- Git

Methods

Five shapefiles including patient location information, hospital location information, road network information, viral suppressions information about patient viral suppression status, and administrative district border information were used during this assessment. These datasets can be found on the ministry's Master facility list website, which was provided by the Zambia Statistics Agency and the Ministry of Health https://mfl.moh.gov.zm/ .

The code begins by loading all prerequisite packages needed for this evaluation.

The code then defines two functions, “generate_handles” a function that takes in a list of labels and a list of colors as input, be used when creating a legend for a map. Another function “scale bar” is created to be used at a later stage, this function basically creates a scale bar.

The code then provides two functions, one of which, "generate handles," uses a list of labels and a list of colors as input to create a map legend. The other function "scale bar" creates a scale bar, these functions will be used later.

The code then opens the datasets and load the five shapefiles as geoDataFrame, and these are reprojected to the UTM Zone 35S projection. The “ccrs.UTM()” method of geopandas is used to carry out the reprojection, with the argument "EPSG:32735" specifying the desired projection for Zambia (Esri, 2016).

Using geopandas' "buffer()" and "within" functions, the script conducts a geospatial analysis to create a 5000-meter buffer around each facility and count the number of patients inside of it. This prints a message to the console that includes the facility's name and the number of patients inside of its buffer radius. It also facilitates for the code to count the number of patients outside each health facility's buffer zone of 5000 meters, producing a printout to the console that reports the facility's name and the number of patients outside the zone (Esri, 2016)

Furthermore, to determine the average distance to the nearest health facility a GeoDataFrame called “patients_outside_buffered” is created by filtering the “patients” GeoDataFrame for patients who are not within any of the health facilities within the 5km buffer. The “unary_union” method is used to combine all the geometries in the “facilities_buffered” GeoDataFrame into a single geometry for the purpose of the “within” method.

Next, a new GeoDataFrame called “patients_outside_buffered_nearest” is created, containing only the geometry of the patients outside the 5000meters buffer. For each patient in “patients_outside_buffered”, the nearest health facility is found using the “nearest_points” method from the “shapely” package. The “nearest_points” method returns a pair of points, so `[1]` is used to select the second point, which represents the nearest facility. The distance between each patient and their nearest facility is then calculated and added as a new column called “distance_to_facility”. The average distance to the nearest health facility is printed using the “mean()” method of the “distance_to_facility” column.

The code loops through each health facility to count the number of patients with suppressed and unsuppressed viral loads within a 5000 meters radius in order to understand the distribution of suppressed and unsuppressed patients. This is made easier by using the GeoPandas library's "gpd.sjoin()" method to do a spatial join. Additionally, the number of patients with suppressed and unsuppressed viral loads outside of a 5000 meters radius is counted. The console is printed with the results.

The code then defines the handles for the facilities, patients, roads, and buffer layers and it also creates a map of patient and health facility locations. It then builds a map showing the locations of the patients and medical facilities, superimposing shapefiles of the district boundaries and roads. The script pushes the map to the GitHub repository and exports it as a PNG file.

Parts of the code were adapted from Dr. McNabb 2023 EGM722 “Programming for GIS and Remote

Sensing" and from week 2 Practical - Mapping with cartopy https://github.com/iamdonovan/egm722 .

Expected Results

The code will show that only 8 patients are within the catchment service areas of the 3 health facilities. Chalabesa Rural Health Centre has 2 patients within 5km radius, Chaya Health Post has 2 patients within 5km radius and Chibansa Urban Health Centre has 4 patients within 5km radius. This can be useful for analyzing the spatial distribution of patients in relation to the health facilities, and identifying areas that may be underserved by the healthcare system.

It will further show that patients outside the catchment service area travel an average distance of 11755.7 meters to access HIV healthcare services at the nearest health facility.

Results from count the number of patients with suppressed and unsuppressed viral load within and outside the radius of each health facility showed that Chalabesa Rural Health Centre and Chaya Health Post has 2 patients within 5km radius with suppressed viral load, 0 patients within 5km radius with unsuppressed viral load, 21 patients outside 5km radius with suppressed viral load and 9 patients outside 5km radius with unsuppressed viral load. Chibansa Rural Health Centre has 2 patients within 5km radius with suppressed viral load, 2 patients within 5km radius with unsuppressed viral load, 21 patients outside 5km radius with suppressed viral load and 7 patients outside 5km radius with unsuppressed viral load.

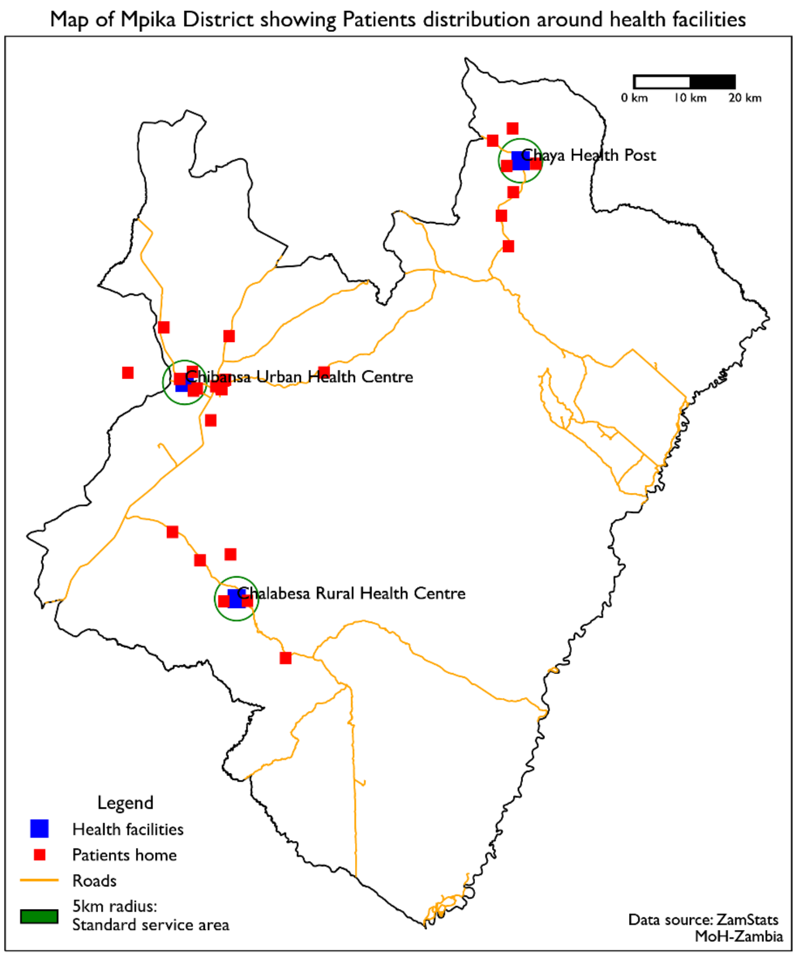

Figure 1: Map showing facility and patient distribution.

Troubleshooting

If the script throws a ValueError: Axis limits cannot be NaN or Inf. This error indicates that one or more of the axis limits (minimum or maximum values) are set to NaN (Not a Number) or Inf (Infinity), which is not a valid value. Try Set axis limits manually, it is possible that Matplotlib is having trouble automatically setting the axis limits.

The code might also throw this error "pyproj.exceptions.CRSError: Invalid projection: +init=epsg:XXXX" if you have modified the projection of the script. Check your CRS string, pyproj version, data, and code, you can usually identify and fix the issue.

References

Africa, E. S., 2016. *Introduction to Geoprocessing Scripts using python.* 8 ed. Midrand: Esri South Africa.

Brooke, E. et al., 2018. Impact of a borderless sample transport network for scaling up viral load monitoring: results of a geospatial optimization model for Zambia. *Journal of the international AIDS society.*

HIV, g., 2022. *HIV.gov.* [Online]
Available at: https://www.hiv.gov/hiv-basics/staying-in-hiv-care/hiv-treatment/viral-suppression
[Accessed 25 November 2022].