



NISMOD

Hands-On 5 – Geospatial Assessment of Infrastructure Exposure to Climate Hazards

This hands-on session focuses on an assessment undertaken in Ghana as part of the ‘Enhancing the resilience of Ghana’s national infrastructure systems’ project, carried out between Ghana’s Ministry of Environment, Science, Technology and Innovation, the UN Office for Project Services (UNOPS), and the University of Oxford. The focus of this tutorial lies in identifying the exposure of Ghana’s roads to climate hazards.

Learning objectives

- Understand where to find, extract and use infrastructure asset and hazard data
- Undertake initial assessments of infrastructure exposure to climate hazards
- Visualise and explore data using QGIS.

Getting started with the appropriate software

Python & Jupyter notebook

First, **create a working directory**, for example create a folder called “Ghana Tutorial” on your desktop.

Next, you need to go to GitHub and **download the material needed for this tutorial** – to do that, download [nismod/snail v0.1](#). Download the ZIP folder on the website. Extract the ZIP folder and save it in your working directory. The Jupyter Notebook we want to work with is in the subfolder “tutorials” - we’ll come back to this later.

Then, download and install Miniconda – to do this, click [here](#)

Once you have installed Miniconda, open the terminal and type in prompt (one after the other):

Create new environment in Python:

```
conda create --name mtop python=3.7
```

Activate new environment:

```
conda activate mtop
```



Configure access to some python packages (click [here](#), for more info):

```
conda config --add channels conda-forge
conda config --set channel_priority strict
```

Install packages:

```
conda install geopandas
conda install pandas
conda install GDAL
conda install matplotlib
```

```
conda install networkx
conda install notebook
conda install seaborn
conda install tqdm
```

```
pip install snkit
```

Open jupyter notebook by typing:

```
jupyter notebook
```

Then, navigate to your working directory; open the folder **snail** (you've downloaded from GitHub) and open the **Notebook "tutorial 1"** in the subfolder tutorials.

More useful links:

- Learn more about Jupyter Notebook in the [Jupyter notebook documentation](#)
- Learn more about Geopandas in the [Geopandas documentation](#)
- If you're new to Python, take a look at the first four chapters of [Think Python](#)

QGIS download

First, a user must install the QGIS software. Details of how to obtain the open-source QGIS software – recommended to be used to visualise the spatial database – are shown below. The guide focuses on use of the database with the recommended software QGIS, but similar features are available in ArcGIS.

The QGIS software (Version 3.8 (64 bit)) can be downloaded from [the QGIS site](#)

For detailed instructions on how to navigate and perform functions in QGIS, see the following tutorials available in the [QGIS documentation](#) and support available from the [QGIS user community](#).

Activity 1: Extract infrastructure data

As a first step we want to extract infrastructure data from an open online database. In this example we are going to work with data from the Open Street Map platform. After saving the data on your computer, we will open it in Jupyter Notebook and take a look at it.



Try It: Extract infrastructure data and add it to your Jupyter Notebook

1. Make sure that you have created a working folder with the name “**Ghana_tutorial**” (for example on your desktop) as described in the “Getting started” section above. This directory will **host all the files**. Later we want to open that directory and the path to that working folder in jupyter notebook.
2. **Open jupyter** as explained in the “Getting started” section above using your terminal. Don’t forget to activate the right environment using “conda activate ...” and type `Jupyter Notebook`. Then navigate to tutorial 1 (in `Ghana_tutorial/snail/tutorials`) and open the notebook.
3. In your jupyter notebook, **define your data_folder** using the following commands, depending on whether you are a Mac or Windows user. “*NAME*” needs to be replaced by your user name – you can check in your explorer.
4. For Mac users, you type
`data_folder = "/Users/NAME/Desktop/Ghana_tutorial"`
in the corresponding cell and press run or shift + enter

For Windows users, you type

```
data_folder = "C:\\Users\\NAME\\Desktop\\Ghana_tutorial"
```

in the corresponding cell and press run or shift + enter

5. The code to import libraries is already provided in your jupyter notebook. Press **run** or shift + enter in your jupyter notebook so that you can import all needed **libraries** (these are links to existing scripts that basically make your life a lot easier). These are already in the notebook.
6. We want to use the website “geofabrik” to download OpenStreet Map Data. To do this, **open this link**. And click on the link that says “**ghana-latest-free.shp.zip**” to download the zip folder.
7. **Extract the zip folder** and save it in the directory, that you have created in step 1 (e.g. your folder `/Desktop/Ghana_tutorial`). When you open the folder, you can see many different files, containing different infrastructure assets. In this hands-on we want to work with the **road data**.
8. Now we want to open the roads dataset in Jupyter Notebook. We want to call a new variable “**GHA_OSM_roads**”. To do that, type the following code
`GHA_OSM_roads = gpd.read_file(os.path.join(data_folder, 'ghana-latest-free.shp/gis_osm_roads_free_1.shp'))` in jupyter and run it.
9. To take a look at the **attribute table**, you **type** `GHA_OSM_roads.head()` and press run or shift + enter
10. Next, we want to make a few changes to the data

Select the **data columns that we want to keep:**

```
GHA_OSM_roads = GHA_OSM_roads[['fclass', 'name', 'oneway', 'geometry']]
```



Filter out minor and residential roads, paths and tracks:

```
roads = roads[
    roads.fclass.isin([
        'motorway',
        'motorway_link',
        'trunk',
        'trunk_link',
        'primary',
        'primary_link',
        'secondary',
        'secondary_link',
        'tertiary',
        'tertiary_link'
    ])
]
```

11. Run the next cells to create topological network information - this adds information that will let us find routes over the road network.

12. Finally **save** the pre-processed data in our **data_folder**. Note this will be the same for both mac and windows users. Type:

```
GHA_OSM_roads.to_file(os.path.join(data_folder, 'GHA_OSM_roads.gpkg'), layer='OSM-roads', driver="GPKG")
```

Voila! We downloaded infrastructure asset data and pre-processed it in Jupyter Notebook.

Activity 2: Extract hazard data

As a second step, we will download the hazard flood layers, provided by Aqueduct in the form of TIFF files. These need to be converted (polygonised) into shapefiles in order to perform the intersection between hazards and infrastructure assets.

Try It: To extract and polygonise the flood hazards, follow these steps:

1. **Download** a TIFF file from Aqueduct. You can access full details on the [World Resources Institute Aqueduct site](#).

For this tutorial, download the following file

- [River flooding \(historical, 100-year return period\)](#)
2. Save the downloaded file in a new folder titled '**flood_layer**' under your data_folder (home directory, see Activity 1, Step 1 (for example: Desktop/Ghana_tutorial))
 3. **Polygonise** all TIFF files into geopackages to ensure their useability. Polygonising means that we convert the raster data into vector data. While raster data consists of a grid with a value in every cell of this grid, vector data contains polygons, lines and points.

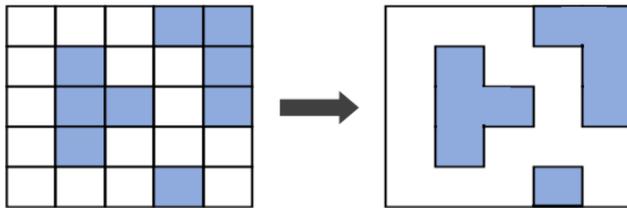


Figure 5.1: Polygonising raster data into vector data

To do this, run the following code, which is already in your jupyter notebook. This goes through every TIFF file in your folder and clips then polygonises it.

```
for root, dirs, files in os.walk(data_folder, 'flood_layer'):
    print("Looking in", root)
    for file in sorted(files):
        if file.endswith(".tif") and not file.endswith("p.tif"):
            print("Found tif file", file)
            stem = file[:-4]
            input_file = os.path.join(root, file)

            # Clip file to bounds
            clip_file = os.path.join(root, f"{stem}_clip.tif")
            try:
                os.remove(clip_file)
            except FileNotFoundError:
                pass
            p = subprocess.run([
                "gdalwarp", "-te", xmin, ymin, xmax, ymax, input_file,
                clip_file],
                capture_output=True)
            print(p.stdout.decode('utf8'))
            print(p.stderr.decode('utf8'))
            print(clip_file)

            # Create vector outline of raster areas
            # note that this rounds the floating-point values of flood
            # depth from
            # the raster to the nearest integer in the vector outlines
```

```

polygons_file = os.path.join(root, f"{stem}.gpkg")
try:
    os.remove(polygons_file)
except FileNotFoundError:
    pass
p = subprocess.run([
    "gdal_polygonize.py", clip_file, '-q', '-f', 'GPKG',
    polygons_file],
    capture_output=True)
print(p.stdout.decode('utf8'))
print(p.stderr.decode('utf8'))
print(polygons_file)

```

At the end of this process, you should have a number of shapefiles (shp) in your flood_layer folder.

Activity 3: Intersect hazards and assets

In this step, we will intersect the flood hazard data with the road data, as specified above under Activity 1 and Activity 2.

Try It: To do the intersection of the flood hazard and the road data, follow these steps:

1. Create a results folder in your working directory
2. Next we want to run an intersection of our infrastructure data and the hazard data. Intersection means that we **intersect both layers** and **create a new data layer** which only contains the parts that are contained in both input layers.

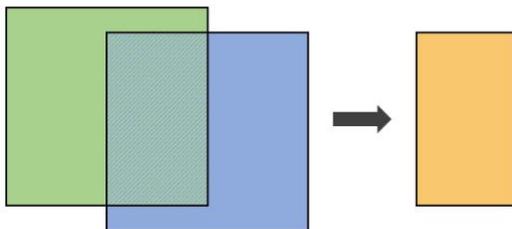


Figure 5.2: Intersection of two data layers

Run the intersection by typing the following line in the next cell in your Jupyter Notebook:

```
flood_intersections = gpd.overlay(GHA_OSM_roads, flood, how='intersection')
```

Then click run or shift + enter to run the code

Activity 4: QGIS

The software QGIS can be used to display vector and raster data. Additionally, the data can be explored and maps can be generated.

Try It: Now we want to use QGIS to explore our data and the results we have created

1. A user must open the QGIS they have downloaded and open a **new project**.

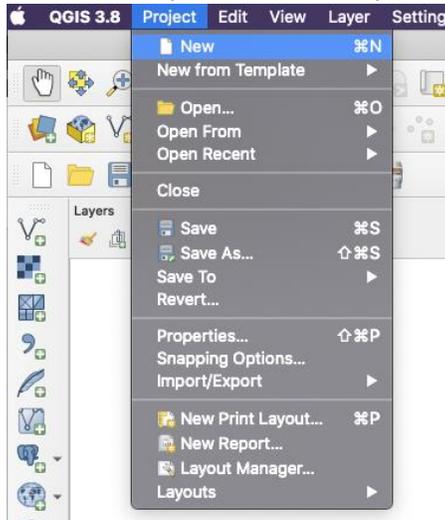


Figure 5.3: How to open a new project in QGIS

2. **Drag the road and hazard shapefile** into the new QGIS project by opening your data_folder, and drag the road shapefile 'gis_osm_roads_free_1.shp,' and some of the **hazard shapefiles**, e.g. 'inuncoast_rcp8p5_wtsub_2050_rp0500_0' into QGIS.

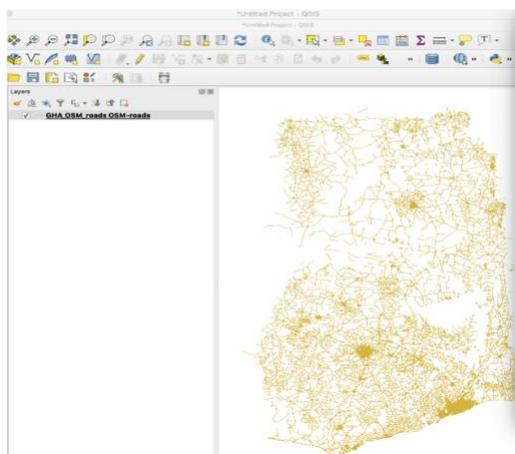


Figure 5.4: Exploring our infrastructure data in QGIS

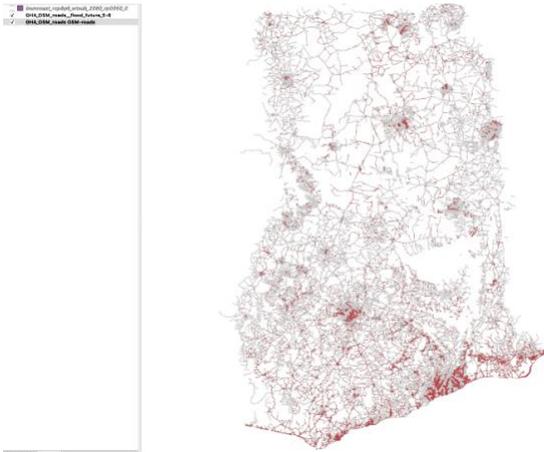


Figure 5.5: Exploring our results in QGIS

The first time the files are opened, the whole country is displayed.

3. To ensure this is the case, **right click** on a layer (all geospatial information on the left side of the QGIS screen), e.g. 'gis_osm_roads_free_1,' then press '**Zoom to layer**' in order to ensure the extend of the map is correct.
4. A user can **zoom into an area** of particular interest by clicking > the '**+** button on the top banner of QGIS, the mouse can then > be used to zoom into a specific area of interest. The '**-**' > button can be used to zoom out.



Figure 5.6: Zooming into our results in QGIS

5. Once the view is chosen, the **contents of the view can be modified**. The different layers on the left-hand side show the impact of **hazards** (inuncoast_rcp8p5_wtsub_2050_rp0500_0) on the road layer. The impact of different hazards can be explored in QGIS by **switching layers on and**



off (for example, undo the tick next to the layer 'inuncoast_rcp8p5_wtsub_2050_rp0500_0' to exclude it from the view.

6. If you want to learn how to export maps from QGIS, see the documentation in this [link](#).

Summary

In the exercises above we learned how to use Jupyter Notebook and QGIS to explore and process geodata. First, we learned where and how to download infrastructure and hazard datasets. Additionally, we processed those datasets in Jupyter Notebook. Then, we learned how to do intersections in Python to identify potentially exposed infrastructure assets. Finally, we opened the data in QGIS to explore our results.