OnSSET/Global Electrification Platform
Hands-on 5: Data extraction to CSV[1]

In order to facilitate the extraction of GIS data to CSV format, a Jupyter Notebook has been developed. To be able to run the Jupyter Notebook, please follow the installation instructions:

1) First, go here and click **Code** > **Download zip** to download the *OnSSET GIS Extraction codes*. Extract the folder in a directory of your choice on the computer.
2) This exercise requires that you have Python and Anaconda installed. To install these, follow the instructions here.

Next, download all of the GIS datasets required for this exercise for Benin here.

If you have any questions, you can post them on the OnSSET Forum.

Learning outcomes

By the end of this exercise, you will be able to:

1) Use the Jupyter Notebook to extract the data to a csv file.

2) List the datasets that are used in the GIS extraction.

3) Create the CSV file required for Hands-on 6.

---

[1] This exercise is an exercise developed by Khavari, B., 2019. Exercise 5: Data extraction to CSV [WWW Document]. OnSSET Teaching Kit. URL https://onsset.github.io/teaching_kit/courses/module_2/Excercise%205/ (accessed 2.18.21).

1. Installing the necessary Python packages

First, you need to install the necessary Python packages to be able to run the Jupyter Notebook to extract the GIS data to a csv-format.

1. Start by opening the Anaconda Prompt.
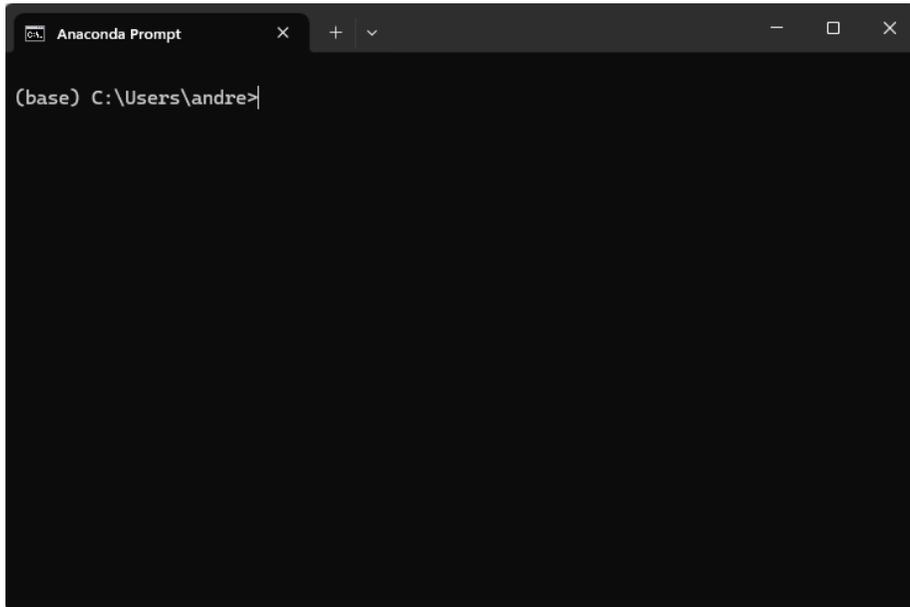
2. Next, you need to go to the location on your computer where you have downloaded and extracted the folder with the *OnSSET GIS Extraction Codes.* On a windows computer, you can copy the path in the File Explorer:
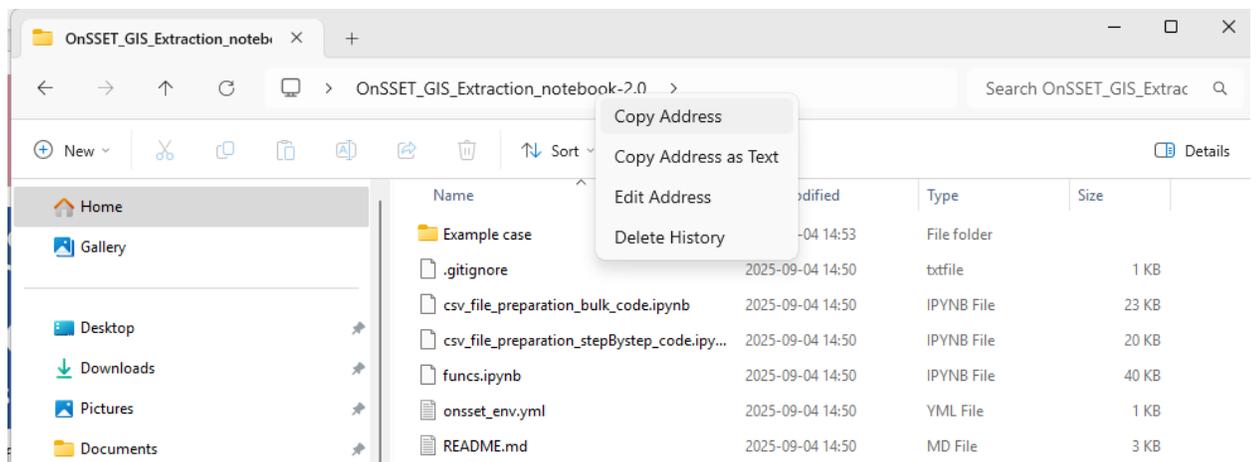
3. In Anaconda Prompt, type **cd** followed by a space and then the path, then press enter.

4. Next, to install all the packages in a new environment, write the following command and press enter:
conda env create --name onsset_env --file onsset_env.yml

5. If everything works correctly, you should see the following, and the installation is complete:

2. Running the GIS extraction Jupyter Notebook

To launch the Jupyter Notebook, open Anaconda Prompt and run the following three commands:

- **cd PATH** (replace PATH with the location on your computer where you have downloaded and extracted the folder with the *OnSSET GIS Extraction Codes*)
- **conda activate onsset_env**
- **jupyter notebook**

This will open up jupyter in a browser. Note that everything is still offline, the browser is simply used as the interface.
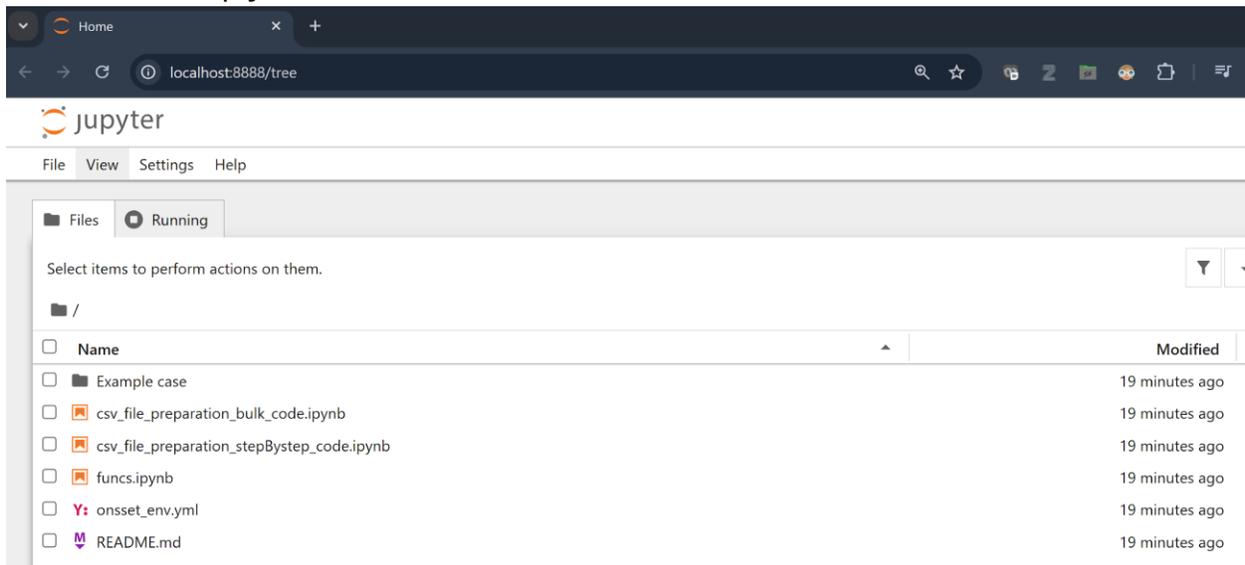
Click on the one called **csv_file_preparation_stepBystep_code.ipynb** to open it.
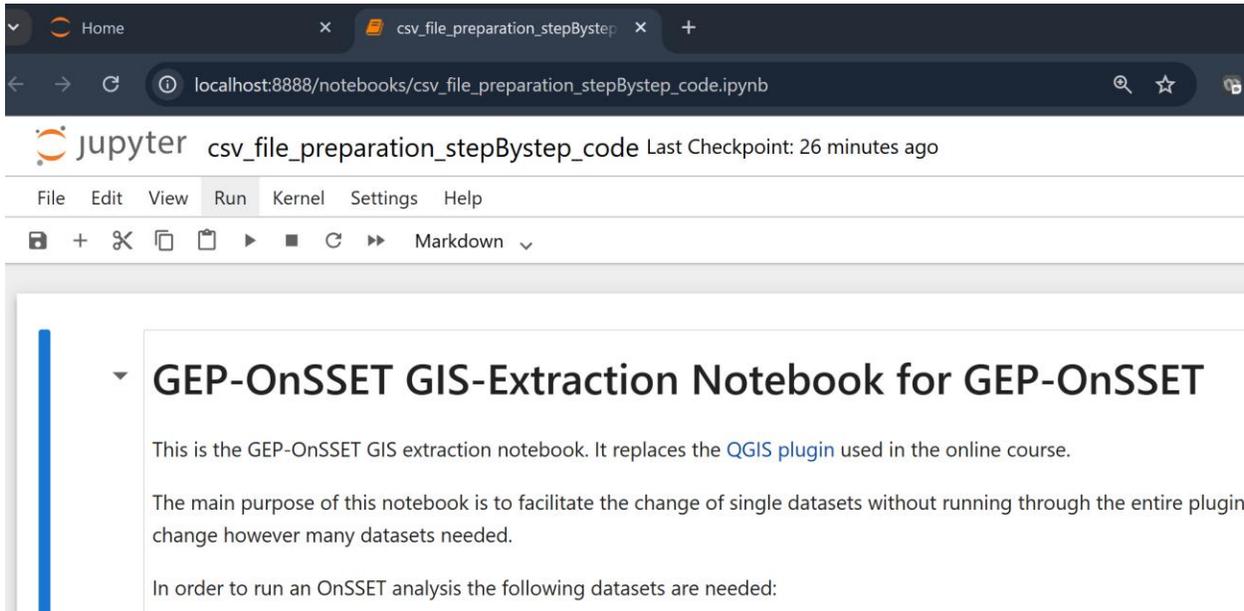
Click on the Run button to run the selected cell (highlighted in blue).
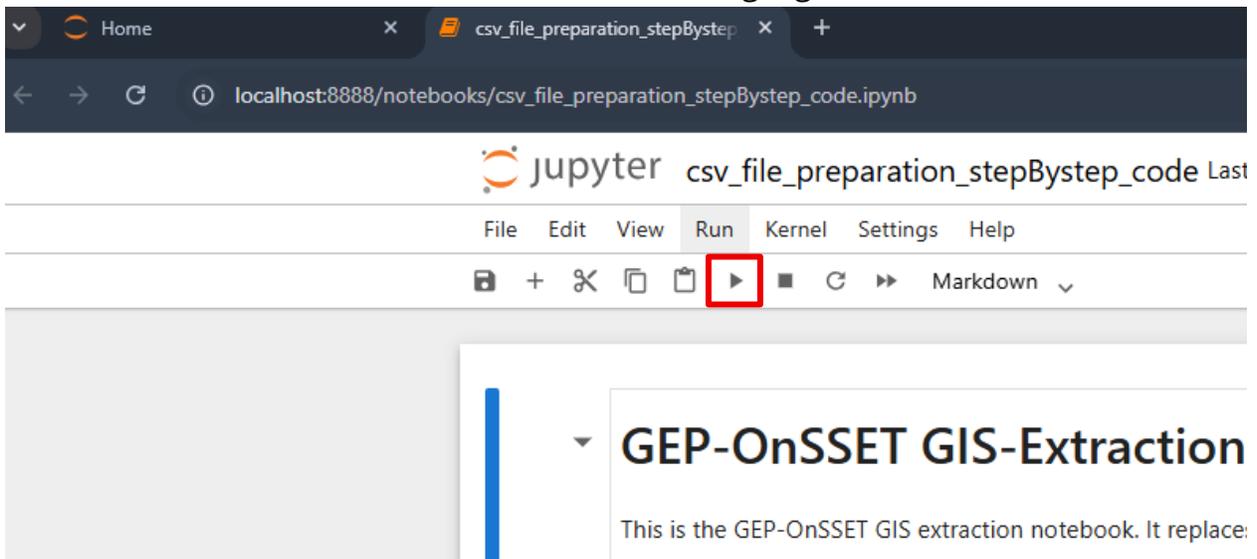
First, you will import the necessary packages and codes. If you receive an error message here, that typically means something went wrong with the installation, or that you have not activated the environment while opening the notebook. Note that while the cell is executing, this is shown with a star * on the left-hand side. Once finished, this is replaced by a number in the brackets (e.g. [1]).

Cells that only contain text do not show the star and numbers.

Cell 1 - Importing necessary packages (Mandatory)

Packages to be used are imported from the funcs.ipynb.

```
[*]   %run funcs.ipynb
      import time
```

In the next cell, we need to define the coordinate system. In this case, you can keep EOSG:3395, so simply run the cell.

Cell 2 - Setting the target coordinate system (Mandatory)

When calculating distances it is important to choose a coordinate system that represents distances correctly in your area of interst. The coordinate system that is give is the World Mercator, these coordinate system generally work well, but the distortions get larger as you move away from the equator.

In order to select your own coordinate system go to epsg.io and type in your area of interest, this will give you a list of coordinate systems to choose from. Once you selected your coordinate system replace the numbers below with the numbers from your coordinate system **(keep the "EPSG" part)**.

NOTE When selecting your coordinate system make sure that you select a system with the unit of meters, this is indicated for all systems on epsg.io

```
[2]:  crs = 'EPSG:3395'
```

In cell 3, create an empty folder where you wish to save the results. When you run the first cell, a pop-up window will appear and you will be asked to select that folder. Click **Ok** and select the folder.

Cell 3 - Select the workspace and the administrative boundaries (Mandatory)

Define the workspace. The output layers will populate this folder. It is highly recommended to select an empty folder as your workspace.

For the administrative boundaries you will have to select an **Polygon** la[...] of interest.

```
[*]:  messagebox.showinfo('OnSSET extraction', 'Output folder')
      workspace = filedialog.askdirectory()

[4]:  messagebox.showinfo('OnSSET', 'Select the admin boundaries')
      admin_path = filedialog.askopenfilename(filetypes = (("vector[...]"*.geojson"]),("all files","*.*")))
      admin = gpd.read_file(admin_path)
```

Next, you will be asked to select the administrative boundaries.

Cell 3 - Select the workspace and the administrative boundaries (Mandatory)

Define the workspace. The output layers will populate this folder. It is highly recommended to select an empty folder as your workspace.

For the administrative boundaries you will have to select an **Polyg**[...]terest.

```
[3]:  messagebox.showinfo('OnSSET extraction', 'Output folder')
      workspace = filedialog.askdirectory()

[*]:  messagebox.showinfo('OnSSET', 'Select the admin boundarie[...]
      admin_path = filedialog.askopenfilename(filetypes = (("ve[...]geojson"]),("all files","*.*")))
      admin = gpd.read_file(admin_path)
```

Click **OK** and select the administrative boundaries GIS layer called

gadm36_BEN_0.shp, found in the *Benin_datasets > Admin folder* that you have downloaded, and click on **Open.**
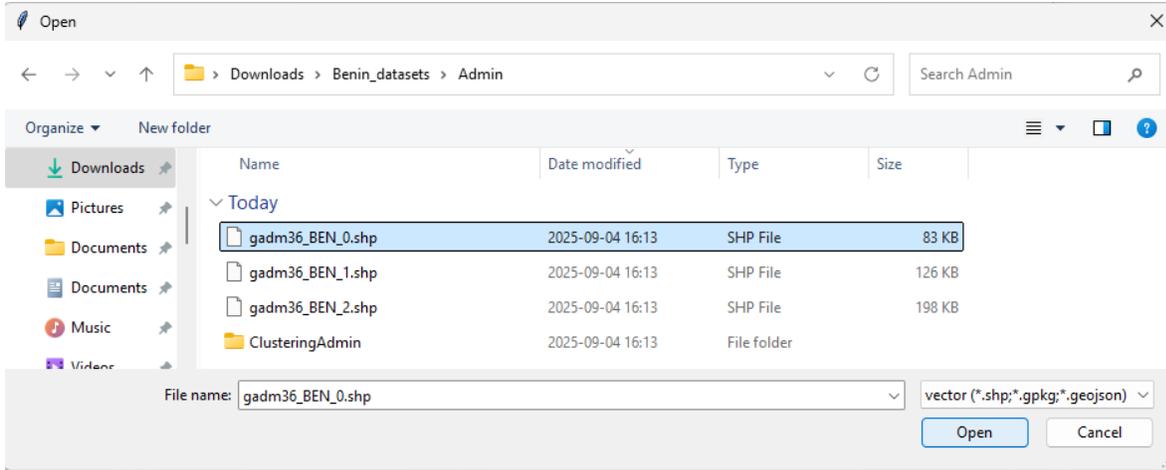
In the next cell, you will be asked to open the population clusters. Select the layer called **Benin.shp** found in *Benin_datasets > Clusters*.
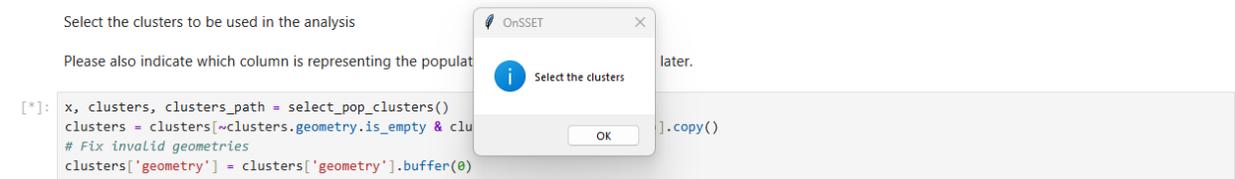
Another pop-up will ask you to select the **Population** column.
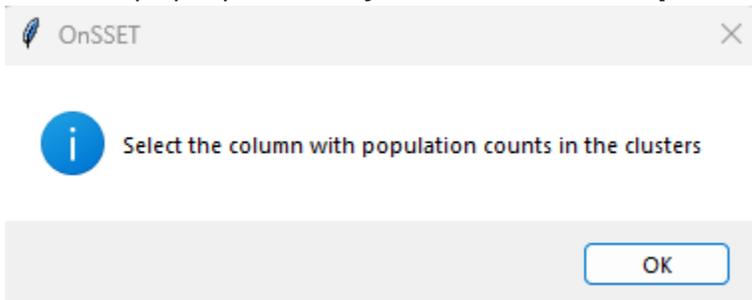
Click OK, then in the drop-down select **Population**.
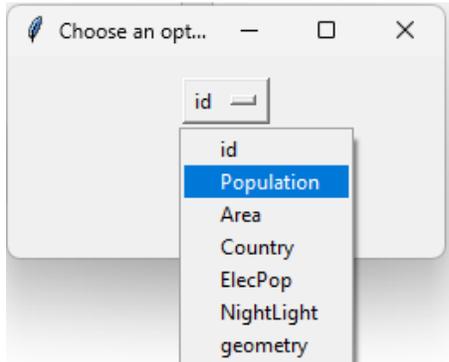
Keep running the following cells, and select according to the list below. Note that some of the datasets may not be mandatory and available in the training exercise, in that case simply click Cancel.

Cell 5: From *Benin_datasets > Solar* select **GHI.tif**

Cell 6: From *Benin_datasets > Traveltime* select **Traveltime.tif**

Cell 7: From *Benin_datasets > windvel* select **windvel.tif**

Cell 8: From *Benin_datasets > NightLights* select **NightLights.tif**

Cell 9: From *Benin_datasets > CustomDemand* select **CustomizedDemand2.tif**

Cell 10: No need to select anything here.

Cell 11: From *Benin_datasets > Substations* select **substations.shp**

Cell 12: From *Benin_datasets > HV* select **Existing.shp**

Cell 13: From *Benin_datasets > HV* select **Planned.shp**

Cell 14: From *Benin_datasets > MV* select **Existing.shp**

Cell 15: From *Benin_datasets > MV* select **Planned.shp**

Cell 16: From *Benin_datasets > Roads* select **roads.shp**

Cell 17: Not available, click *Cancel*

Cell 18: From *Benin_datasets > Hydro* select **hydro_points.shp**
In the pop-up windows for this dataset, select first **PowerMW** and then **MW.**

Cell 19: Not available, click *Cancel*

Cell 20: From *Benin_datasets > Admin* select **GADM36_BEN_1.shp**
In the pop-up window for this dataset, select **NAME_1**

Cell 21: This is the final cell. It will export the data into a csv-file called **GEP-OnSSET_InputFile.csv** in the output folder you selected in the beginning (you should see the message as below).

Cell 21 - Conditioning & Export (Mandatory)

This is the final cell in the extraction. This cell has to be run.

```
[22]: clusters = conditioning(clusters, workspace, x)
      print('Workspace: ', workspace)

      Processing finished: 2025-09-05 10:20:18
      The extraction file is now ready for review & use in the workspace directory as 'OnSSET_InputFile.csv'!
      Workspace:  C:/Users/andre/Downloads/GIS_Extraction
```

In the csv-file, every row represents one settlement. To understand what each column means, have a look at [this document](#) or the [OnSSET Documentation](#).

GridCellAr	Country	id	Pop	NightLight	ElecPop	GHI	TravelHou	WindVel	Residentia	Substatio	CurrentHV	CurrentM	RoadDist	Hydropow	Hydropow	Hydropow	Admin_1	PlannedH	PlannedM	Transform
0.008755	Benin	0	35.04101		0			3.2891	2109.333	48.64474	92.56765	34.01263	22.15677	29.82892		41	2809.16	Benin	92.56765	34.01263	99999
0.10506	Benin	1	420.4921	0	0	2114	2.911985	3.214466	2114	45.34328	89.67838	30.66213	19.42012	26.36925	41	2809.16	Benin	89.67838	30.66213	99999
0.017509	Benin	2	52.56151	0	0	2114	2.421116	3.274856	2114	43.42408	87.93505	28.72733	17.83034	24.39386	41	2809.16	Benin	87.93505	28.72733	99999
0.017509	Benin	3	70.08201	0	0	2110.167	3.283333	3.293075	2110.167	44.02903	88.2383	29.37427	17.94877	25.16949	41	2809.16	Benin	88.2383	29.37427	99999
0.035018	Benin	4	140.164	0	0	2112	2.766667	3.219613	2112	42.19968	86.86232	27.48982	16.92813	23.10916	41	2809.16	Benin	86.86232	27.48982	99999
0.026263	Benin	5	105.123	0	0	2110	3.283333	3.301775	2110	43.89049	88.05619	29.24426	17.75115	25.06348	41	2809.16	Benin	88.05619	29.24426	99999
0.008754	Benin	6	17.5205	0	0	2110	3.5	3.323397	2110	44.79868	88.67241	30.20338	18.27399	26.13338	41	2809.16	Benin	88.67241	30.20338	99999
0.017509	Benin	7	52.56151	0	0	2112	2.933333	3.180689	2112	41.47427	86.26194	26.7549	16.48989	22.32695	41	2809.16	Benin	86.26194	26.7549	99999
0.140069	Benin	8	525.6151	0	0	2109.646	3.565451	3.348192	2109.646	44.23998	88.09759	29.65306	17.70029	25.60651	41	2809.16	Benin	88.09759	29.65306	99999
0.017508	Benin	11	87.60251	0	0	2110	2.516667	3.242803	2110	39.29613	84.17303	24.57449	14.66688	20.14831	41	2809.16	Benin	84.17303	24.57449	99999
0.035016	Benin	12	192.7255	0	0	2109	3	3.390752	2109	41.56787	85.74604	26.93576	15.49753	22.81528	41	2809.16	Benin	85.74604	26.93576	99999
0.008754	Benin	13	17.5205	0	0	2109	3.683333	3.439604	2109	43.35098	86.98892	28.82585	16.57024	24.91063	41	2809.16	Benin	86.98892	28.82585	99999
0.008754	Benin	14	35.04101	0	0	2109.833	4.066667	3.508441	2109.833	43.24693	86.63203	28.79185	16.21747	24.99975	41	2809.16	Benin	86.63203	28.79185	99999
0.008754	Benin	15	35.04101	0	0	2109	3.583333	3.366796	2109	42.29603	85.9267	27.78603	15.5107	23.9127	41	2809.16	Benin	85.9267	27.78603	99999
0.017507	Benin	16	52.56151	0	0	2110	4.066667	3.476808	2110	43.37633	86.57931	28.97279	16.19154	25.26082	41	2809.16	Benin	86.57931	28.97279	99999
0.017507	Benin	17	35.04101	0	0	2107	2.475772	3.339276	2107	38.30426	83.02224	23.60162	13.40182	19.29159	41	2809.16	Benin	83.02224	23.60162	99999
0.035014	Benin	18	140.164	0	0	2107	2.698681	3.351528	2107	37.90114	82.6119	23.20117	13.01925	18.90787	41	2809.16	Benin	82.6119	23.20117	99999
0.052521	Benin	19	227.7665	0	0	2108.861	3.625082	3.451236	2108.861	44.04235	86.91531	29.73027	16.6281	26.14181	41	2809.16	Benin	86.91531	29.73027	99999
0.05252	Benin	20	332.8896	0	0	2109	3.55	3.356618	2109	40.99062	84.7252	26.46989	14.33225	22.5962	41	2809.16	Benin	84.7252	26.46989	99999
0.02626	Benin	21	87.60251	0	0	2110	3.65	3.463419	2110	43.63537	86.51842	29.32797	16.22307	25.75255	41	2809.16	Benin	86.51842	29.32797	99999
0.070026	Benin	22	455.5331	0	0	2108.708	3.592816	3.416511	2108.708	40.19518	83.90821	25.69165	13.51912	21.86305	41	2809.16	Benin	83.90821	25.69165	99999
0.008753	Benin	23	35.04101	0	0	2109	2.433333	3.276457	2109	36.0692	81.01677	21.34941	11.99399	16.97086	41	2809.16	Benin	81.01677	21.34941	99999

The CSV file is the final step of this exercise, which we will use in the final hands-on exercise to run OnSSET.