

# climada

March 11, 2024

## 1 Climate risk modelling with CLIMADA

Doris Folini, Mathias Hauser and Victor Wattin Håkansson

Spring Term 2024

together with **Svenja Seeber** and **Stefanie Börsig**

All relevant course information to be found on the webpage <https://iac.ethz.ch/edu/courses/bachelor/vorbereitung/python-in-geosciences.html>

This exercise is about the use of the python package CLIMADA. It was first designed by **Chahan Kropf** (ckropf@ethz.ch) and later adapted by **Alessio Ciullo**.

Table of Contents

### 1. Introduction

What is climate risk modelling?

What is CLIMADA?

CLIMADA resources

This seminar

### 2. Installation

Before installation

Installation steps

### 3. Obligatory exercises

Exercise 1 : First full impact calculation

Exercise 2 : Load NetCDF storm data into CLIMADA

Exercise 3 : Compute impacts from the storm data

Exercise 4 : Compute impacts for different countries

### 4. Elective exercises

Exercise 1 : Manipulate the impact matrix

Exercise 2 : Use future scenarios

Exercise 3 : Use another hazard

```
[ ]: # This code is to hide the warnings (such as deprecation notices).
# You may add this at the beginning of each script if you want to hide warnings.
import warnings
warnings.filterwarnings('ignore')

# This code is to set the resolution of printed graphs.
import matplotlib as mpl
mpl.rcParams["figure.dpi"] = 150
```

## 1.1 1. Introduction

### 1.1.1 What is climate risk modelling?

Extreme weather events such as tropical cyclones, wildfires, floods, droughts, heavy rains, heat-waves, etc..., (also called natural hazards) are “rare” by their nature. Computer models are therefore needed to help scientists, policy makers, and planners to understand, assess and adapt to the risks associated with such events, today and in the future.

In the context of natural hazards, risk is understood as the product of the hazardous event probability (i.e., how likely it is for the event to occur) and its severity (e.g., the monetary damage on houses or the number of deceased people). In particular, the basic building blocks of most computer models used to assess weather and climate risk are: - Hazard: probabilistic set of events. In such a set, each event is a map of the event’s intensity (e.g., the maximum windspeed per location in the case of a storm, the maximum flood depth in the case of a flood, etc.). - Exposure: a map of physical and non physical objects that are impacted by the hazard. Examples include population per area, houses value, biodiversity level per area. - Vulnerability: an function that captures how much the exposure is affected from a given hazard intensity (e.g. wood houses take more damage to wind than stone houses, poorer people might suffer more from a flood than richer)

An example of such computer models is CLIMADA.

### 1.1.2 What is CLIMADA?

CLIMADA is an event-based climate risk assessment tool written in Python and it is fully open source and open access, which means that anyone can use, contribute to or improve CLIMADA. The full code is available at [Github](#). CLIMADA is primarily developed and maintained by the Weather and Climate Risk Group at ETH and its users community include public agencies (e.g., MeteoSwiss), private companies (e.g., Celsius Pro), other researchers (e.g., Frankfurt School of management, UN-university) and NGOs.

CLIMADA provides a framework for users to combine exposure, hazard and vulnerability to calculate risk. Users can create probabilistic impact data from event sets, look at how climate change affects these impacts, and see how effectively adaptation measures can change them. CLIMADA also allows for studies of individual events, historical event sets and forecasts. The model is a highly customisable, meaning that users can work with out-of-the-box data provided for different hazards, population and economic exposure, or can provide their own data for part or all of the analysis. The pre-packaged data make CLIMADA particularly useful for users who focus on just one element of risk, since CLIMADA can ‘fill in the gaps’ for hazard, exposure or vulnerability in the rest of the analysis.

The model core is designed to give as much flexibility as possible when describing the elements of risk, meaning that CLIMADA isn't limited to particular hazards, exposure types or impacts. We love to see the model applied to new problems and contexts. CLIMADA provides classes, methods and data for exposure, hazard and impact functions (also called vulnerability functions), plus a financial model and a framework to analyse adaptation measures. Additional classes and data for common uses, such as economic exposures or tropical storms.

### 1.1.3 CLIMADA Resources

- [CLIMADA github repository](#)
- [CLIMADA documentation](#)
- [CLIMADA zotero library](#)
- [CLIMADA impact assessment paper](#)
- [CLIMADA case study examples](#)
- [CLIMADA data API](#)

### 1.1.4 This seminar

In this seminar you will learn how to use CLIMADA to make simple risk analyses. We will explore some of the core functionalities of CLIMADA, together with a few key Python modules useful for handling geographical and scientific data.

The exercises are divided in two sets: - obligatory exercises for everyone - three elective ones, of which you are expected to do at least one. You can do the elective exercises in any order.

All results are to be handed-in as jupyter notebooks 'ipynb'.

## 1.2 2. Installation

### Before installation:

- Have at least 10 GB of **free storage space** on your computer.
- Never install CLIMADA in a (Microsoft) OneDrive.
- Ensure a **stable internet connection** for the installation procedure. Do **not** use a metered, mobile connection!
- Have [miniforge](#) installed.

### Installation steps:

1. Open a terminal. Follow steps 2-4 below, entering the commands in your command prompt, but do not include the '>' character.
2. Navigate to the course folder `ip_python`. Depending on where you created the folder you need to adapt the path, e.g.:

```
> cd Documents/ETH/ip_python
```

3. Create a new Conda environment named `climada_env`:

```
> conda create -n climada_env -c conda-forge climada
```

4. Activate the environment:

```
> conda activate climada_env
```

You should now see (`climada_env`) appear in the beginning of your command prompt in the terminal. This means the environment is activated.

**All the exercises below require you to work in a jupyter notebook. In order for CLIMADA to work, make sure to select the kernel `climada_env` before running your scripts.**

### 1.3 3. Obligatory Exercises

#### 1.3.1 Exercise 1 : First full impact calculation

We begin with a brief example of how to compute risk for winter storms in Switzerland. This example uses pre-computed data from the CLIMADA API.

As a first introductory exercise, please reproduce the code from this whole section in a jupyter notebook `impact_climada_YOUR_NAME.ipynb`.

#### Tasks for this exercise:

- Reproduce the code from this exercise and get familiar with it.
- Reproduce all the plots, add (meaningful) titles to the plots
- For each plot, try to understand what it means and provide a brief comment of what info can be retrieved from it. See the CLIMADA [documentation](#) if needed.
- Save all the plots to `.png` files
- When there is a **question**, provide the answer in your `impact_climada_YOUR_NAME.ipynb` in written form.

*Tip: the figure of a `matplotlib.axes` (typically named `ax`) is obtained as `fig = ax.get_figure()`. A figure can easily be saved to file.*

*Tip: files paths are most easily manipulated with the `Pathlib` library.*

- [Pathlib: documentation](#)
- [Pathlib: Article on how to use and why](#)

*Tip: Do not forget that you must have the CLIMADA environment `climada_env` activated when working with the following exercises. If working with jupyter lab select the `climada_env` kernel, see [course material](#).*

```
[ ]: #Load the relevant climada modules
from climada.entity import ImpactFuncSet
from climada.entity.impact_funcs.storm_europe import ImpfStormEurope
from climada.util.api_client import Client
```

```
[ ]: #API client module
client = Client()
```

First, we load the exposures from the CLIMADA [API](#). Here we use the data produced by the LitPop module, which combines nightlight satellite imagery with population census data to estimate the distribution of ‘physical assests’ (mainly buildings). We retrieve this info for Switzerland.

*Tip: For more details about the LitPop module see:*

- [Asset exposure data for global physical risk assessment](#)

- [LitPop documentation](#)

```
[ ]: assets_ch = client.get_litpop(country='CHE')
```

The data is saved in the form of a GeoDataFrame from [GeoPandas](#) (geopandas is a package to represent data as tables and is optimized for geo-coded data). It contains information on the latitude, longitude and value of each data point. In addition, the impact functions for each data point must be set in `impf_` (see below). Finally, the `region_id` is the iso-3 code of the country of the datapoint, and `geometry` is a column used for certain data transformations.

```
[ ]: assets_ch.gdf
```

```
[ ]: ax = assets_ch.plot_raster();
ax.set_title('Exposures');
```

*Tip: do you obtain 'deprecation notices'? This is because Python packages evolve independently of one another. When someone wants to change something, say change the name `exposures` to `exposure`, developers do not simply change the name as this would break the code of everyone using the older naming. First, one makes a 'deprecation' warning to inform users that in future iterations, the name will change. Thus, users have time to adapt their code.*

Second, we need to load the hazard, in this case a probabilistic set of winter-storms from the CLIMADA API.

*Tip: For more details on the generation of winter-storms see:*

- [Comparing an insurer's perspective on building damages with modelled damages from pan-European winter windstorm event sets: a case study from Zurich, Switzerland](#)

```
[ ]: client = Client()
storm_ch = client.get_hazard('storm_europe', properties={'country_iso3alpha': 'CHE'})
```

*Tip: checkout the iso3 country codes system.*

- [https://en.wikipedia.org/wiki/ISO\\_3166-1](https://en.wikipedia.org/wiki/ISO_3166-1)
- <https://www.iso.org/iso-3166-country-codes.html>

`storm_ch` now contains a probabilistic set of winterstorms.

```
[ ]: #Number of storms
storm_ch.size
```

```
[ ]: #Probabilities in terms of frequency per year
storm_ch.frequency
```

```
[ ]: storm_ch.plot_intensity(event=-1)
```

**Question:** What does the '-1' mean for `.plot_intensity()`? What other values are acceptable? What do they mean?

*Tip: See the documentation of CLIMADA for the method [plot\\_intensity](#).*

Third, we need to relate the intensity of the hazard to the impact on the exposures, in others words we need to quantify the vulnerability. This is done with an Impact Function. For a real case study, this would have to be calibrated using existing impact data. Here, we use the function that was calibrated for building damage in the canton of Zürich based on data from [Welker et al. \(2021\)](#) and that is directly available in CLIMADA. This impact function is obtained with the method `from_welker`.

```
[ ]: # Vulnerability
impf = ImpfStormEurope.from_welker()
impfset = ImpactFuncSet()
impfset.append(func=impf)
impfset.plot()
```

*Tip: in CLIMADA, different exposure points can have different impact functions. Think for instance that a wind storm probably impacts differently industrial buildings and residential buildings. This is why the `ImpactFunc` is added to a set of impact functions, i.e. `ImpactFuncSet`. In the present case, we considered a single impact function identical for all exposures points.*

- For more info, see the documentation for [ImpactFuncSet](#)

Now, we have to assign to all exposure points the impact function defined above (with `id=1`).

```
[ ]: # Associate the impact function 1 from WS to all assests in the exposures
assets_ch.gdf.rename(columns={'impf_': 'impf_WS'}, inplace=True)
assets_ch.gdf['impf_WS'] = 1
```

**Question:** What do MDD, PAA and MDR mean?

*Tip: See the [CLIMADA paper](#) or the [CLIMADA documentation on impact functions](#).*

**Question:** is the choice of the impact function appropriate for the chosen exposures?

We are now ready for the impact computation. The only thing to do is to combine the three elements - hazard, exposures, and impact function.

```
[ ]: from climada.engine import ImpactCalc
impact = ImpactCalc(exposures=assets_ch, impfset=impfset, hazard=storm_ch).
↳ impact(save_mat=True)
```

```
[ ]: impact.plot_raster_eai_exposure();
```

Different metrics can now be extracted, such as:

```
[ ]: # Average annual impact aggregated over all exposures points
impact.aai_agg
```

**Question:** What other metrics can be obtained from an `impact` object?

*Tip: See the documentation for the [impact class](#)*

*Tip: The value `save_mat=True` is needed so that the full impact matrix is saved as attribute to `impact.imp_mat`. The impact matrix contains the impact per exposures (one column per point) and event (one row per event).*

**Question:** Can you calculate the average annual impact as a percentage of the total assets value at risk?

*Tip:* One way to get the total value of the assets at risk, is to use the method `assets_ch.affected_total_value()` together with the hazard object `storm_ch`. For a description, type:

```
[ ]: help(assets_ch.affected_total_value)
```

### 1.3.2 Exercise 2 : Load NetCDF storm data into CLIMADA

In the previous exercise we carried out an end-to-end impact calculation using the hazard, exposure and vulnerability data in the CLIMADA API. Normally, however, these data come from different sources and are not readily available in CLIMADA. In these cases, some pre-preprocessing is needed to load the data in CLIMADA. Hazard and Exposure data will likely come as NetCDF files, since NetCDF is among the most commonly used data format for geographical data.

In this exercise, we will replicate what done in Exercise 1. Instead of using the hazard data from the Data API, however, we will load a NetCDF file that contains winter-storms in Europe. The netcdf file is named `fp_eraint_1999122606_507_0.nc` and can be found on the [course website](#). Please download it and save it into the folder `/ipp_analysis`. In this exercise, we will use the CLIMADA built-in methods to transform these data into a StormEurope CLIMADA object, which will then allow us to carry out impact calculation in exercise 3.

For this exercise, please save your script as a jupyter notebook with name `wisc1_YOUR_NAME.ipynb`.

#### Tasks for this exercise:

- Read the netcdf file for one of the storms using the `xarray` Python module. Print the xarray.
- Describe in words the data included in the file.
- When there is a **question**, provide the answer in your `wisc1_YOUR_NAME.ipynb` in written form.
- Find the position (lon/lat) with the largest windspeed and print its value. Where is it (description of the place including the country, region, etc.)?
- For the impact calculation in the next exercise we will only use the windfields over Switzerland. Thus, extract these data from the whole dataset.
- Save the extracted data back to a netcdf file called `storm_ch.nc`.

*Tip:* NetCDF is a very useful format to store data with different coordinate systems, including information on how the data was made and what the data is (i.e. metadata).

- <https://www.unidata.ucar.edu/software/netcdf/>
- <https://en.wikipedia.org/wiki/NetCDF>

```
[ ]: import xarray as xr
ds = xr.open_dataset("fp_eraint_1999122606_507_0.nc")
```

```
[ ]: print(ds)
```

In this file, the maximum wind speeds at each `latitude` and `longitude` is saved in the variable `max_wind_gust`.

```
[ ]: ds.max_wind_gust
```

The data can easily be plotted.

```
[ ]: ds.max_wind_gust.plot();
```

*Tip: To get the axes object associated with the plot you can use `ax = plt.gca()`*

**Question :** What storm was this? When did it happen? Find information (in newspapers archives for example) about the impact of this storm in Europe? in Switzerland?

*Tip: below is an example code that can be adapted to select data for a specific area of interest.*

```
[ ]: min_lon, max_lon = 10, 30
min_lat, max_lat = 30, 45

mask_lon = (ds.longitude >= min_lon) & (ds.longitude <= max_lon)
mask_lat = (ds.latitude >= min_lat) & (ds.latitude <= max_lat)

ds_selection = ds.where(mask_lon & mask_lat, drop=True)
```

```
[ ]: ds_selection.max_wind_gust.plot();
```

*Tip: an xarray dataset can be saved to netcdf with `ds.to_netcdf('filename')`.*

### 1.3.3 Exercise 3 : Compute storm impacts to Switzerland

In this exercise we will compute impacts to Switzerland using the exposure and vulnerability data retrieved in Exercise 1 and the storm data generated in Exercise 2.

Save your script in a jupyter notebook name as `wisc2_YOUR_NAME.ipynb`.

#### Tasks for this exercise:

- Load the `storm_ch.nc` file directly using the CLIMADA `StormEurope.from_footprints(filename)` method. (Yes, for some data, CLIMADA comes with built-in functions that rely on xarray and allow reading the data into a CLIMADA object)
- Based on the code in Exercise 1, compute the impact from this storm.
- Plot the intensity of the event.
- Plot the impact.
- Extract two important values from the impact calculation:
  - The Aggregated Average Annual Impact `aai_agg`: this is the total impact that is expected each year on average.
  - The Total Value `tot_value`: this is the value of all assets.
- Find the name and date of this event. Provide a link to a newspaper article that describes the real damage from this storm. Compare the impact numbers to the ones obtained with CLIMADA.

*Tip: the `StormEurope` class can be imported by doing:*

```
[ ]: from climada.hazard import StormEurope
```



*Tip: The aai\_agg and the tot\_val are obtained directly from the impact object.*

- The aai\_agg is obtained from `impact.aai_agg`.
- The total value is obtained from `impact.tot_value`.

### 1.3.4 Exercise 4 : Compute impacts for different countries

In the following exercise, you will repeat the impact calculation done for Switzerland to four other European countries of your choice. This will require adapting the previous code to work for other countries too. For the hazard data, you can either use the storm data in the `fp_eraint_1999122606_507_0.nc` or just the DATA API.

Save your script to a jupyter notebook `compare_countries_YOUR_NAME.ipynb`.

#### Tasks for this exercise:

- Make a function `storm_impact(country)` that returns the impact to a given country.
- Compute the impact for four European countries of your choice.
- Save the aai\_agg for each country. In addition, save the tot\_val and compute the relative value of the aai\_agg to the tot\_val.
- Save these impact values in a pandas dataframe with columns ('Country name', 'Country ISO', 'AAI AGG', 'Total Value', 'Relative AAI AGG').
- Make a bar plot to compare the AAI AGG and make another bar plot to compare the Relative AAI AGG.
- Save all plots to .png files.
- Provide a brief comment of your results.

*Tip: a smart way to achieve the tasks above is to employ “for loops”*

Example code to construct the required dataframe and make the bar plot:

```
[ ]: import numpy as np
import pandas as pd

cntry_list = ['Switzerland', 'France']
iso_list = ['CHE', 'FRA']
aai_agg = np.array([10e5, 16.3e5])
tot_val = np.array([53e7, 17.2e8])
rel_val = aai_agg / tot_val

df = pd.DataFrame(
    {'Country name' : cntry_list,
     'Country ISO' : iso_list,
     'AAI AGG' : aai_agg,
     'Total Value' : tot_val,
     'Relative AAI AGG' : rel_val
    }
)
```

```
[ ]: df
```

```
[ ]: df['Total Value']
```

```
[ ]: df.plot(x='Country name', y='AAI AGG', kind='bar');
```

## 1.4 4. Elective exercises

All the exercises in this section can be chosen and done in any order. You should have some results for at least one of them.

### 1.4.1 Exercise 1 : Manipulate the impact matrix

Save your script in a jupyter notebook `impmat_YOUR_NAME.ipynb`.

- Extract the impact matrix and the frequency array from the `Impact` object used in the Exercise 1.
- Compute the average annual aggregated impact using this data.
- Compute also the maximum, average and standard deviation of the impact over all exposures points and all events.

*Tip: the impact matrix is obtained as `impact.imp_mat` and the frequency as `impact.frequency`*

*Tip: the impact matrix is stored as a [sparse matrix](#) as implemented in the `scipy` module. The `scipy.sparse.csr_matrix` object can be manipulated much like a normal matrix and can be converted to a `numpy` 2-d array if needed. Sparse matrices are used to reduce the amount of require memory when a lot of entries are identical (for example 0).*

### 1.4.2 Exercise 2 : Use future scenario

Save your script in a jupyter notebook `future_YOUR_NAME.ipynb`.

Based on the hazard, exposures and impact functions from Exercise 1 we shall “simulate” a future scenario.

Simulate climate change. In principle, this requires the use of complex physical models. However, even with the best models the effects of a changing climate on natural hazards is hard to predict. One simple approach is to consider scenarios which might be only loosely based on physical modelling. This allows for ‘what-if’ types of questions: - Multiply the storm intensity with 10%. Multiply the storm frequency with 5%. - How does the impact change? Is this a good way to model climate change? - Find values in the literature that give estimates for the change in intensity AND frequency of European winter storms? Can you implement these changes?

Simulate assets growth. Estimating the change in money-value for the exposures can be proxied by the change in macro-economical indicators such as the GDP.

- Look for long-time GDP change data and estimate the growth until 2050.
- Multiply the exposures asset value by this value. How does the impact change?
- Is GDP a good measure of asset-value growth?

Simulate vulnerability change. This basically means that one adapts (for good or bad) to the hazards. For example, someone that has never seen a tsunami will not recognize it and not move away from the ocean when the water retreats. A village that has experienced a tsunami will have a memory, and thus recognize the early signs of a tsunami and move to safety. Another example

would be the building of avalanche barriers in the mountains which reduces the vulnerability of villages in mountains. Or, a country might be subject to an economic crisis which leads to a decrease in financial means to cope with natural hazards which increase the vulnerability.

- Modify the impact function to a high-vulnerability curve and a low-vulnerability curve
- How does the impact change?

### 1.4.3 Exercise 3 : Use another hazard

Save your script in a jupyter notebook `other_hazard_YOUR_NAME.ipynb`.

Can you, using all the resources available in the CLIMADA documentation, do a climate risk study as showed above for a different country (for example Mozambique, Vietnam, Bolivia, Japan, ...) and a different hazard (for example river flood, tropical cyclones, wildfires, ...) than presented here?

*Tip : Many hazards and exposures exist precomputed on the CLIMADA API.*

- See the documentation on the [API](#)