CLIMADA documentation

Release 1.3.1-dev

CLIMADA contributors

Dec 03, 2019

CONTENTS

This is the documentation for version v1.3.1-dev. In CLIMADA-project you will find CLIMADA's contributors, repository and scientific publications.

CHAPTER

ONE

INTRODUCTION

CLIMADA implements a fully probabilistic risk assessment model. According to the [IPCC2014], natural risks emerge through the interplay of climate and weather-related hazards, the exposure of goods or people to this hazard, and the specific vulnerability of exposed people, infrastructure and environment. The unit chosen to measure risk has to be the most relevant one in a specific decision problem, not necessarily monetary units. Wildfire hazard might be measured by burned area, exposure by population or replacement value of homes and hence risk might be expressed as number of affected people in the context of evacuation, or repair cost of buildings in the context of property insurance.

Risk has been defined by the International Organization for Standardization as the "effect of uncertainty on objectives" as the potential for consequences when something of value is at stake and the outcome is uncertain, recognizing the diversity of values. Risk can then be quantified as the combination of the probability of a consequence and its magnitude:

$$risk = probability \times severity$$

In the simplest case, × stands for a multiplication, but more generally, it represents a convolution of the respective distributions of probability and severity. We approximate the $severity$ as follows:

$$severity = F(hazard\ intensity, exposure, vulnerability) = exposure * f_{imp}(hazard\ intensity)$$

where f_{imp} is the impact function which parametrizes to what extent an exposure will be affected by a specific hazard. While 'vulnerability function' is broadly used in the modelers community, we refer to it as 'impact function' to explicitly include the option of opportunities (i.e. negative damages). Using this approach, CLIMADA constitutes a platform to analyse risks of different hazard types in a globally consistent fashion at different resolution levels, at scales from multiple kilometres down to meters, depending on the purpose.

CHAPTER

TWO

INSTALLATION

Please execute the instructions of the following text boxes in a Terminal or Anaconda Prompt.

2.1 Download

Download the last CLIMADA release available in climada releases as a zip or tar.gz file. Uncompress it to your local computer. Hereinafter `climada_python-x.y.z` refers to the downloaded folder of CLIMADA version x.y.z.

2.2 Unix Operating System

2.2.1 Install environment with Anaconda

1. **Anaconda**: Download or update to the latest version of Anaconda. Execute it.
2. **Install dependencies**: In the *Environments* section, use the *Import* box to create a new virtual environment from a yml file. A dialogue box will ask you for the location of the file. Provide first the path of climada's `climada_python-x.y.z/requirements/env_climada.yml`. The default name of the environment, *climada_env*, appears. Click the *Import* button to start the installation.

   The installation of the packages will take some minutes. No dialogue box should appear in the meantime. If an error happens, try to solve it looking into the details description.

   Finally, set the `climada_python-x.y.z` folder path into the environment using the following command:

   ```
   source activate climada_env
   conda develop /your/path/to/climada_python-x.y.z/
   conda deactivate
   ```

3. **Test installation**: Before leaving the *Environments* section of Anaconda, make sure that the climada environment, *climada_env* is selected. Go to the *Home* section of Anaconda and install and launch Spyder (or your preferred editor). Open the file containing all the installation tests, `tests_install.py` in `climada_python-x.y.z` folder and execute it. If the installation has been successful, an OK will appear at the end (the execution should last less than 2min).
4. **Run tutorials**: In the *Home* section of Anaconda, with *climada_env* selected, install and launch *jupyter notebook*. A browser window will show up. Navigate to your `climada_python-x.y.z` repository and open `doc/tutorial/1_main_climada.ipynb`. This is the tutorial which will guide you through all climada's functionalities. Execute each code cell to see the results, you might also edit the code cells before executing. See *Tutorial* for more information.

2.2.2 Install environment with Miniconda

1. **Miniconda**: Download or update to the latest version of Miniconda.
2. **Install dependencies**: Create the virtual environment *climada_env* with climada's dependencies:

   ```
   cd climada_python-x.y.z
   conda env create -f requirements/env_climada.yml --name climada_env
   ```

   Finally, set the `climada_python-x.y.z` folder path into the environment using the following command:

   ```
   source activate climada_env
   conda develop /your/path/to/climada_python-x.y.z/
   conda deactivate
   ```

3. **Test installation**: Activate the environment, execute the installation tests and deactivate the environment when finished using climada:

   ```
   source activate climada_env
   python3 tests_install.py
   source deactivate
   ```

   If the installation has been successful, an OK will appear at the end (the execution should last less than 2min).

4. **Run tutorials**: Install and launch *jupyter notebook* in the same environment:

   ```
   source activate climada_env
   conda install jupyter
   jupyter notebook --notebook-dir /path/to/climada_python-x.y.z
   ```

   A browser window will show up. Open `climada_python-x.y.z/doc/tutorial/1_main_climada.ipynb`. This is the tutorial which will guide you through all climada's functionalities. Execute each code cell to see the results, you might also edit the code cells before executing. See *Tutorial* for more information.

2.3 Windows Operating System

2.3.1 Install environment with Anaconda

See *Install environment with Anaconda*.

2.4 FAQs

- ModuleNotFoundError; climada libraries are not found. Try to include *climada_python-x.y.z* path in the environment *climada_env* path as suggested in Section 2 of *Install environment with Anaconda*. If it does not work you can always include the path manually before executing your code:

  ```
  import sys
  sys.path.append('path/to/climada_python-x.y.z')
  ```

- ModuleNotFoundError; some python library is not found. It might happen that the pip dependencies of *env_climada.yml* (the ones specified after `pip:`) have not been installed in the environment *climada_env*. You can then install them manually one by one as follows:

  ```
  source activate climada_env
  pip install library_name
  ```

  where `library_name` is the missing library.

- Conda right problems in macOS Mojave: try the solutions suggested here https://github.com/conda/conda/issues/8440.

CHAPTER

THREE

TUTORIAL

The main tutorial walks you through all the functionalities of this version of CLIMADA. There, you will find the links to additional tutorials for specific features of CLIMADA, such as different hazard and exposure models. You can execute it by opening `climada_python-x.y.z/doc/tutorial/1_main_climada.ipynb` with Jupyter Notebook and the CLIMADA environment (*climada_env*) activated (i.e. CLIMADA needs to be installed as in *Installation*).

Navigate through the tutorial here:

3.1 CLIMADA features

The functionality of climada is gathered in the following classes:

- **Entity**: socio-economic models:
- Exposures: exposed values
  - BlackMarble: regional economic model from nightlight intensities and economic indicators (GDP, income group)
  - LitPop: regional economic model using nightlight and population maps together with several economic indicators
- ImpactFuncSet: collection of impact functions per hazard
  - ImpactFunc: one adjustable impact function
  - IFTropCyclone: definition of impact functions for tropical cyclones
- DiscRates: discount rates per year
- MeasureSet: collection of measures for adaptation
  - Measure: one configurable measure
- **Hazard**: meteorological models:
- TropCyclone: tropical cyclone events
- **Impact**: impacts of the Hazard and Entity interaction.
- **CostBenefit**: adaptation options appraisal.
- **Add-ons**: OpenStreetMap and Google Earth Engine routines.

3.2 Risk assessment

3.2.1 Entity

The entity class is just a container for the exposures, impact functions, discount rates and measures. It can be directly filled from an excel file following climada's template or from MATLAB files of the climada MATLAB version. The excel template can be found in `climada_python/data/system/entity_template.xlsx`.

[1]:

```python
from climada.entity import Entity
from climada.util.constants import ENT_DEMO_TODAY

# absolute path of file following template.
ent_file = ENT_DEMO_TODAY
ent_fl = Entity()
ent_fl.read_excel(ent_file)
```

```
2019-11-13 12:12:01,631 - climada - DEBUG - Loading default config file: /Users/
↪aznarsig/Documents/Python/climada_python/climada/conf/defaults.conf
```

Every class has a `check()` method. This verifies that the necessary data to compute the impact is correctly provided and logs the optional variables that are not present. Use it always after filling an instance.

[2]:

```python
ent_fl.check() # checks exposures, impact functions, discount rates and measures
```

```
2019-11-13 12:12:06,648 - climada.entity.exposures.base - INFO - crs set to default␣
↪value: {'init': 'epsg:4326', 'no_defs': True}
2019-11-13 12:12:06,649 - climada.entity.exposures.base - INFO - ref_year metadata␣
↪set to default value: 2018
2019-11-13 12:12:06,650 - climada.entity.exposures.base - INFO - value_unit metadata␣
↪set to default value: USD
2019-11-13 12:12:06,651 - climada.entity.exposures.base - INFO - meta metadata set to␣
↪default value: None
2019-11-13 12:12:06,652 - climada.entity.exposures.base - INFO - centr_ not set.
2019-11-13 12:12:06,654 - climada.entity.exposures.base - INFO - category_id not set.
2019-11-13 12:12:06,655 - climada.entity.exposures.base - INFO - region_id not set.
2019-11-13 12:12:06,656 - climada.entity.exposures.base - INFO - geometry not set.
```

Exposures

The `Entity`'s `exposures` attribute contains geolocalized values of anything exposed to the hazard, let it be monetary value of assets or number of human lifes, for example. It is of type `Exposures`.

See Exposures tutorial to learn how to fill and use exposures.

See LitPop to model economic exposures using night-time light and population densitites. See BlackMarble to model economic exposures based only on night-time light intensities. To combine your exposure with OpenStreetMap's data see OSM.

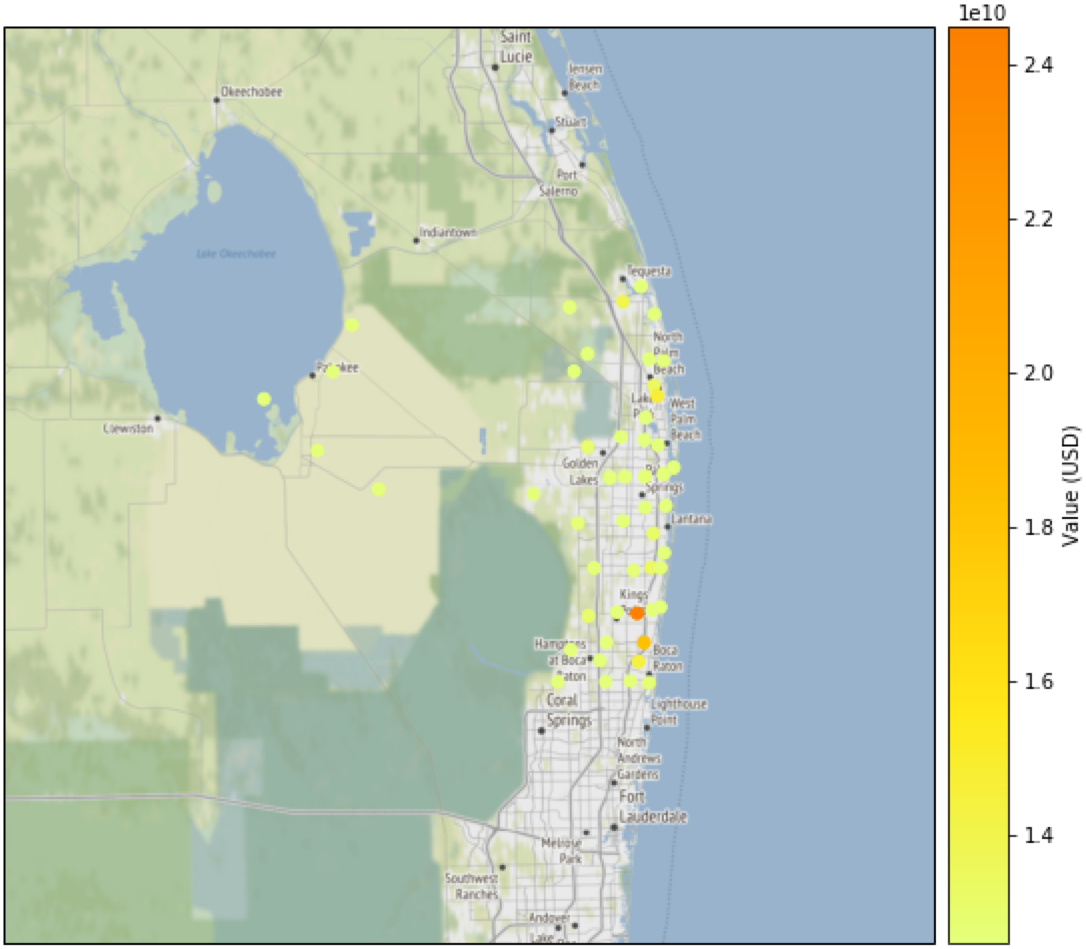
[3]:

```python
%matplotlib inline
ent_fl.exposures.plot_basemap(buffer=50000.0); # exposures in Florida
```

```
2019-11-13 12:12:06,668 - climada.util.coordinates - INFO - Setting geometry points.
2019-11-13 12:12:06,681 - climada.entity.exposures.base - INFO - Setting latitude and␣
↪longitude attributes.
```

```
/Users/aznarsig/Documents/Python/climada_python/climada/util/plot.py:318:␣
↪UserWarning: Tight layout not applied. The left and right margins cannot be made␣
↪large enough to accommodate all axes decorations.
  fig.tight_layout()
```

```
2019-11-13 12:12:07,878 - climada.entity.exposures.base - INFO - Setting latitude and␣
↪longitude attributes.
```

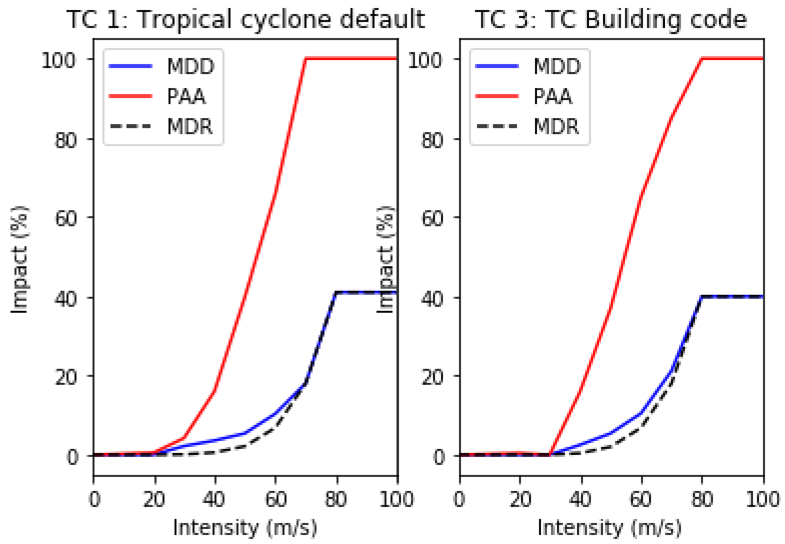


Impact Functions

The `impact_funcs` attribute is of type `ImpactFuncSet`. As such, it contains impact functions for different hazards.

See Impact Functions tutorial to learn how to handle this class.

```
[4]: ent_fl.impact_funcs.plot('TC'); # tropical cyclone impact functions
```



Adaptation Measures

The `measures` attribute is of type `MeasureSet`. This class is a container of `Measure` instances, similarly to `ImpactFuncSet` containing several `ImpactFunc`. Adaptation measures aim to decrease hazards impacts and are subjected to a cost.

See Adaptation Measures to learn to handle measures.

```
[5]: # print measures names
     print(ent_fl.measures.get_names())
```

```
{'TC': ['Mangroves', 'Beach nourishment', 'Seawall', 'Building code']}
```

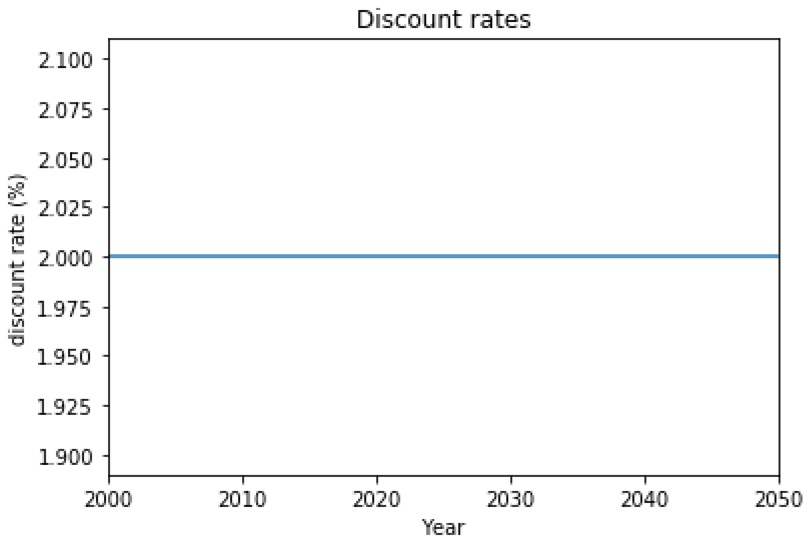
Discount Rates

The `disc_rates` attribute is of type `DiscRates`. This class contains the discount rates for the following years and computes the net present value for given values.

See Discount Rates.

```
[6]: ent_fl.disc_rates.plot()
```

```
[6]: <matplotlib.axes._subplots.AxesSubplot at 0x1c26178860>
```



3.2.2 Hazard

Hazards are characterized by their frequency of occurrence and the geographical distribution of their intensity. A `Hazard` instance collects events of the same hazard type (e.g. tropical cyclone, flood, drought, . . .) over the same centroids. They might be historical events or synthetic.

See Hazard to learn how to handle hazards.

See TropCyclone to learn to model tropical cyclones. TCSurge implements an approximation on tropical cyclones surges.

To use satellite images in your models follow the tutorial Google Earth Engine.

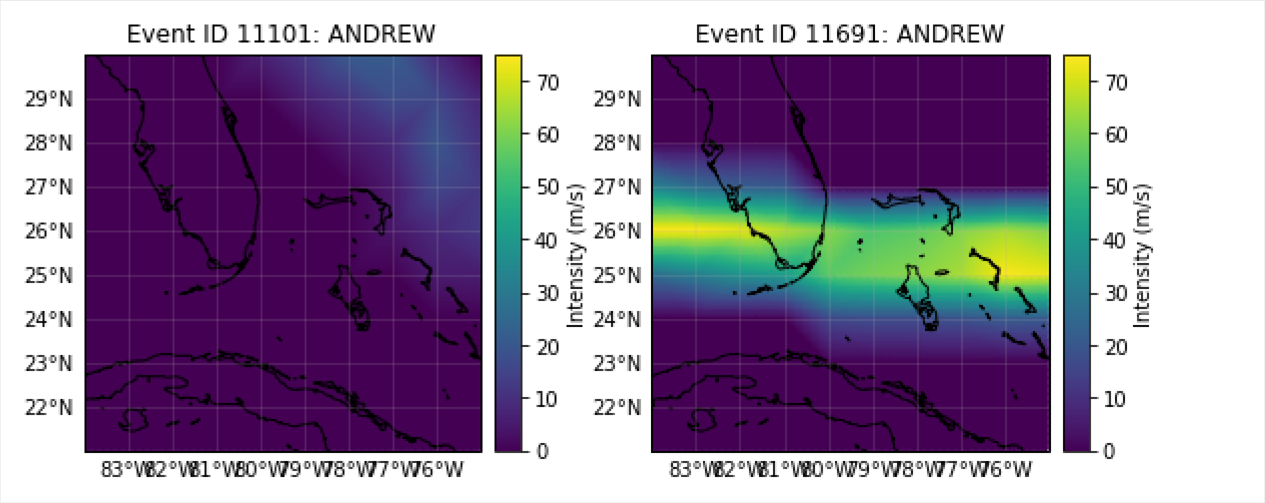
A complete set of tropical cyclones events in Florida can be found in file `HAZ_DEMO_MAT`. This contains 1445 historical events from year 1851 to 2011 and 9 synthetic events for each historical one.

[7]:
```
from climada.hazard import Hazard
from climada.util import HAZ_DEMO_MAT
tc_fl = Hazard('TC')
tc_fl.read_mat(HAZ_DEMO_MAT, 'Historic and synthetic tropical cyclones in Florida␣
↪from 1851 to 2011.')
tc_fl.plot_intensity('ANDREW') # plot intensity of hurricanes Andrew
print('Two hurricanes called Andrew happened in ', tc_fl.get_event_date('ANDREW'))
```

```
2019-11-13 12:12:09,224 - climada.hazard.base - INFO - Reading /Users/aznarsig/
↪Documents/Python/climada_python/data/demo/atl_prob.mat
2019-11-13 12:12:09,267 - climada.hazard.centroids.centr - INFO - Reading /Users/
↪aznarsig/Documents/Python/climada_python/data/demo/atl_prob.mat
```

```
/Users/aznarsig/Documents/Python/climada_python/climada/util/plot.py:318:␣
↪UserWarning: Tight layout not applied. The left and right margins cannot be made␣
↪large enough to accommodate all axes decorations.
  fig.tight_layout()
```

```
Two hurricanes called Andrew happened in  ['1986-06-05', '1992-08-16']
```



3.2.3 Impact

The impact of hazard events over an entity can be computed easily from the previously explained classes. By computing the impact for each event (historical and synthetic), the `Impact` class provides different risk measures, as the expected annual impact per exposure, the probable maximum impact for different return periods and the total average annual impact.

Let us compute the impact of tropical cyclones over the exposures selected in Florida.

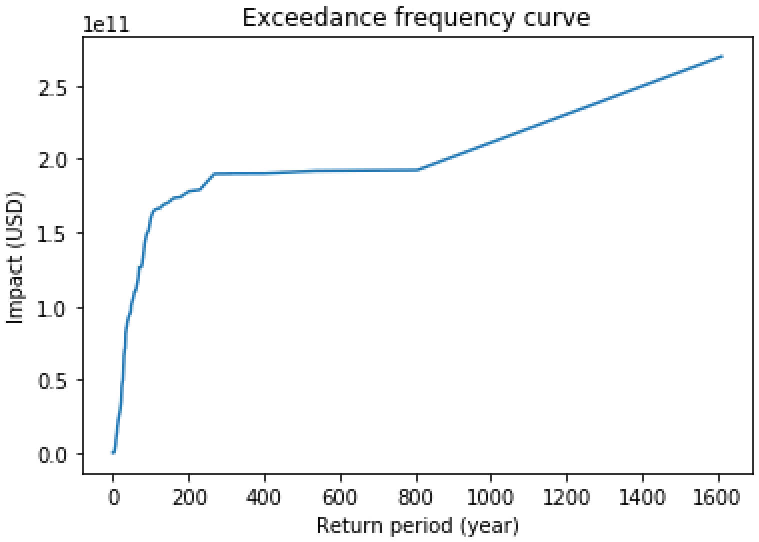
The configurable parameter `MAX_SIZE` controls the maximum matrix size contained in a chunk. You can decrease its value if you are having memory issues when using the `Impact`'s `calc` method. A high value will make the computation fast, but increase the memory use. The configuration file is located at `climada_python/climada/conf/defaults.conf`.

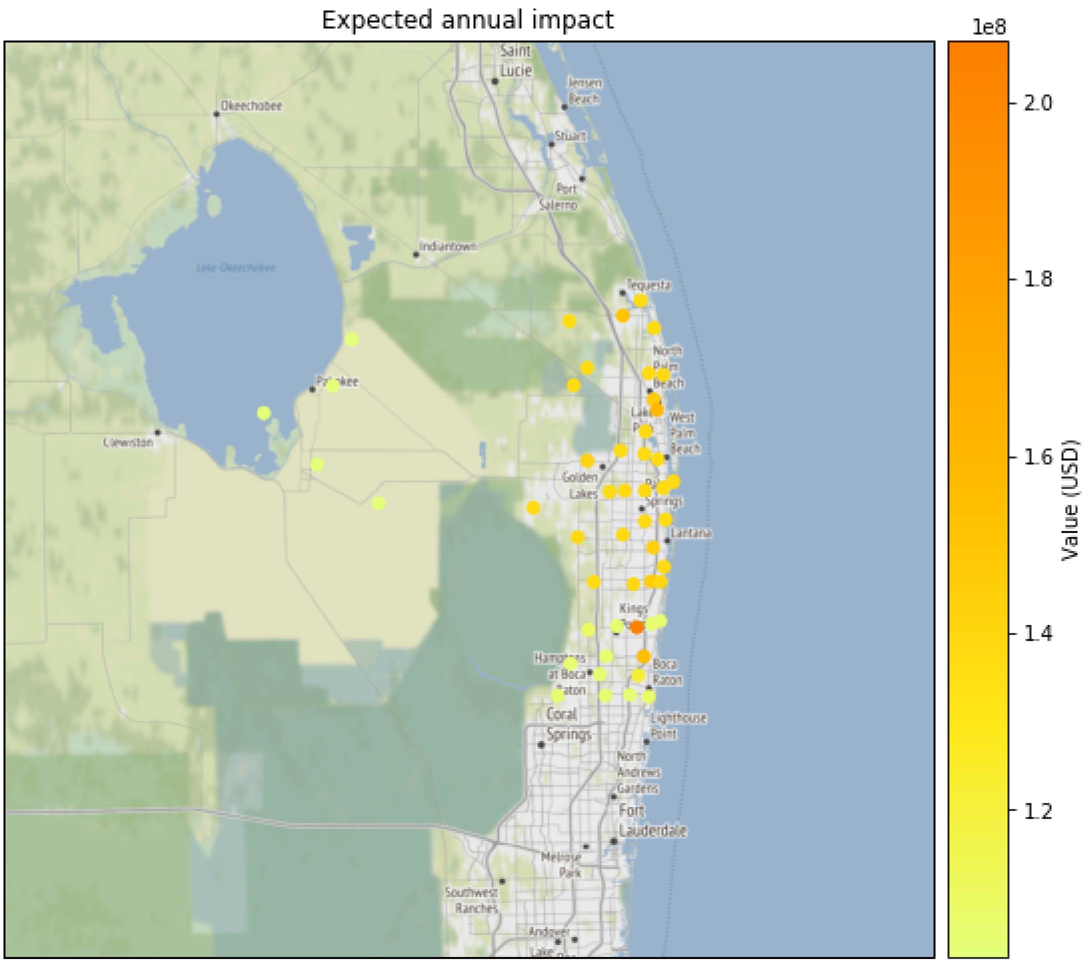
```
[8]: from climada.engine import Impact

imp_fl = Impact()
imp_fl.calc(ent_fl.exposures, ent_fl.impact_funcs, tc_fl)

freq_curve_fl = imp_fl.calc_freq_curve() # impact exceedence frequency curve
freq_curve_fl.plot();

print('Expected average annual impact: {:.3e} USD'.format(imp_fl.aai_agg))

imp_fl.plot_basemap_eai_exposure(buffer=50000.0); # average annual impact at each␣
↪exposure
```

```
2019-11-13 12:12:19,541 - climada.entity.exposures.base - INFO - Matching 50␣
↪exposures with 100 centroids.
2019-11-13 12:12:19,545 - climada.engine.impact - INFO - Calculating damage for 50␣
↪assets (>0) and 14450 events.
Expected average annual impact: 6.512e+09 USD
2019-11-13 12:12:19,595 - climada.util.coordinates - INFO - Setting geometry points.
2019-11-13 12:12:19,602 - climada.entity.exposures.base - INFO - Setting latitude and␣
↪longitude attributes.
```

```
/Users/aznarsig/Documents/Python/climada_python/climada/util/plot.py:318:␣
↪UserWarning: Tight layout not applied. The left and right margins cannot be made␣
↪large enough to accommodate all axes decorations.
  fig.tight_layout()
```

```
2019-11-13 12:12:20,603 - climada.entity.exposures.base - INFO - Setting latitude and␣
↪longitude attributes.
```





We can save our variables in pickle format using the `save` function and load them with `load`. This will save your results in the folder specified in the configuration file. The default folder is a `results` folder which is created in the current path (see default configuration file `climada/conf/defaults.conf`). However, we recommend to use CLIMADA's writers in `hdf5` or `csv` whenever possible.

[9]:

```
import os
from climada.util import save, load
save('impact_florida.p', imp_fl)

# Later, the data can be read as follows:
abs_path = os.path.join(os.getcwd(), 'results/impact_florida.p') # absolute path
data = load(abs_path)
print('Data read:', type(data))
```

```
2019-11-13 12:12:21,244 - climada.util.save - INFO - Written file /Users/aznarsig/
↪Documents/Python/climada_python/doc/tutorial/results/impact_florida.p
Data read: <class 'climada.engine.impact.Impact'>
```

`Impact` also has `write_csv()` and `write_excel()` methods to save the impact variables, and `write_sparse_csr()` to save the impact matrix (impact per event and exposure). Use the class doc to get more information about these functions.

See Impact to learn more about impact calculations.

3.3 Adaptation options appraisal

The adaptation measures defined before can be valued by estimating its cost-benefit ratio. This is done in the class `CostBenefit`.

Let us suppose that the socioeconomic and climatoligical conditions remain the same in 2040. We then compute the cost and benefit of every adaptation measure as follows:

[10]:

```
from climada.engine import CostBenefit

cost_ben = CostBenefit()
cost_ben.calc(tc_fl, ent_fl, future_year=2040) # prints costs and benefits
cost_ben.plot_cost_benefit() # plot cost benefit ratio and averted damage of every␣
↪exposure
cost_ben.plot_event_view() # plot averted damage of each measure for every return␣
↪period
```

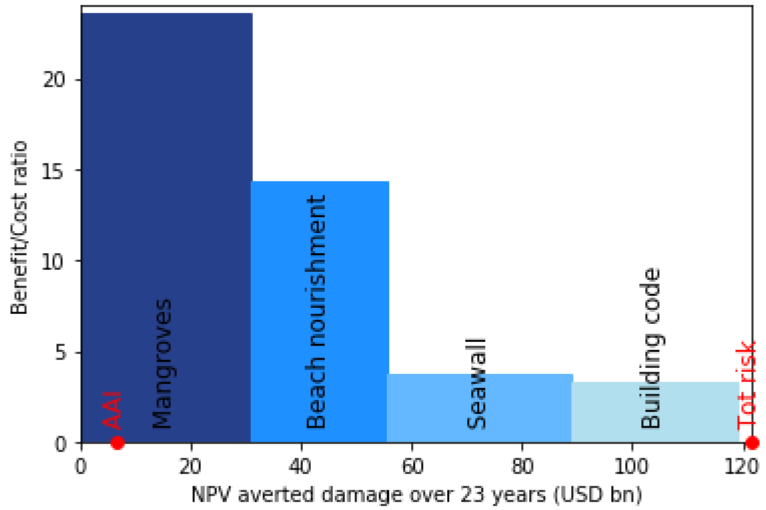
```
2019-11-13 12:12:21,260 - climada.engine.impact - INFO - Exposures matching centroids␣
↪found in centr_TC
2019-11-13 12:12:21,263 - climada.engine.impact - INFO - Calculating damage for 50␣
↪assets (>0) and 14450 events.
2019-11-13 12:12:21,292 - climada.engine.impact - INFO - Exposures matching centroids␣
↪found in centr_TC
2019-11-13 12:12:21,294 - climada.engine.impact - INFO - Calculating damage for 50␣
↪assets (>0) and 14450 events.
2019-11-13 12:12:21,365 - climada.engine.impact - INFO - Exposures matching centroids␣
↪found in centr_TC
2019-11-13 12:12:21,367 - climada.engine.impact - INFO - Calculating damage for 50␣
↪assets (>0) and 14450 events.
2019-11-13 12:12:21,583 - climada.engine.impact - INFO - Exposures matching centroids␣
↪found in centr_TC
2019-11-13 12:12:21,586 - climada.engine.impact - INFO - Calculating damage for 50␣
↪assets (>0) and 14450 events.
2019-11-13 12:12:21,644 - climada.engine.impact - INFO - Exposures matching centroids␣
↪found in centr_TC
```

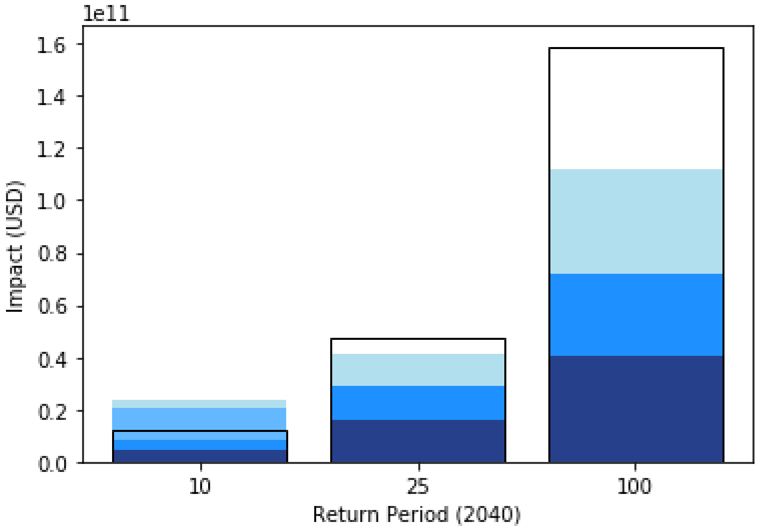
(continues on next page)

(continued from previous page)

```
2019-11-13 12:12:21,646 - climada.engine.impact - INFO - Calculating damage for 50␣
↪assets (>0) and 14450 events.
2019-11-13 12:12:21,720 - climada.engine.impact - INFO - Exposures matching centroids␣
↪found in centr_TC
2019-11-13 12:12:21,721 - climada.engine.impact - INFO - Calculating damage for 50␣
↪assets (>0) and 14450 events.
2019-11-13 12:12:21,802 - climada.engine.cost_benefit - INFO - Computing cost benefit␣
↪from years 2018 to 2040.

Measure              Cost  (USD bn)    Benefit  (USD bn)    Benefit/Cost
-----------  ----------  ------------  ---------
Mangroves                  1.31177             31.0058        23.6367
Beach nourishment          1.728               24.6898        14.2881
Seawall                    8.87878             33.133          3.7317
Building code              9.2                 30.3762         3.30177

-------------  -----  -----
Total climate risk:   121.505   (USD bn)
Average annual risk:    6.5122  (USD bn)
Residual risk:          2.3001  (USD bn)
-------------  -----  -----
Net Present Values
```

```
[10]: <matplotlib.axes._subplots.AxesSubplot at 0x1c26f49940>
```





Let us now assume that the exposure evolves according to `ENT_DEMO_FUTURE` in 2040 and that the intensity of the hazards increase uniformly due to climate change.

[11]:
```
import copy
from climada.util.constants import ENT_DEMO_FUTURE

# future conditions
ent_future = Entity()
ent_future.read_excel(ENT_DEMO_FUTURE)
ent_future.check()
ent_future.exposures.ref_year = 2040

haz_future = copy.deepcopy(tc_fl)
haz_future.intensity.data += 15  # increase uniformly the intensity

cost_ben = CostBenefit()
cost_ben.calc(tc_fl, ent_fl, haz_future, ent_future, save_imp=True)
cost_ben.plot_cost_benefit() # plot cost benefit ratio and averted damage of every␣
↪exposure
cost_ben.plot_event_view() # plot averted damage of each measure for every return␣
↪period
ax = cost_ben.plot_waterfall(tc_fl, ent_fl, haz_future, ent_future) # plot expected␣
↪annual impact
ax.set_title('Expected Annual Impact in 2015 and 2040')
ax = cost_ben.plot_waterfall_accumulated(tc_fl, ent_fl, ent_future) # plot␣
↪accumulated impact from present to future
cost_ben.plot_arrow_averted(ax, accumulate=True, combine=True, disc_rates=ent_fl.disc_
↪rates) # plot total averted damages
```

```
2019-11-13 12:12:22,380 - climada.entity.exposures.base - INFO - crs set to default␣
↪value: {'init': 'epsg:4326', 'no_defs': True}
2019-11-13 12:12:22,381 - climada.entity.exposures.base - INFO - ref_year metadata␣
↪set to default value: 2018
2019-11-13 12:12:22,381 - climada.entity.exposures.base - INFO - value_unit metadata␣
↪set to default value: USD
2019-11-13 12:12:22,382 - climada.entity.exposures.base - INFO - meta metadata set to␣
↪default value: None
2019-11-13 12:12:22,383 - climada.entity.exposures.base - INFO - centr_ not set.
```

(continues on next page)

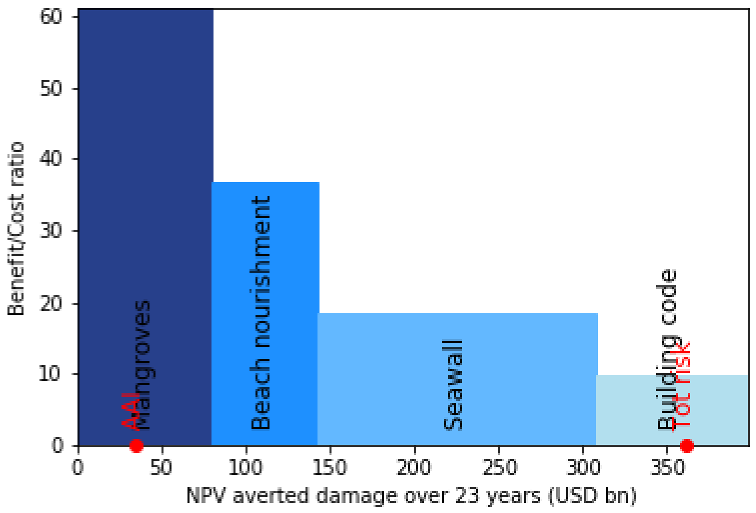
(continued from previous page)

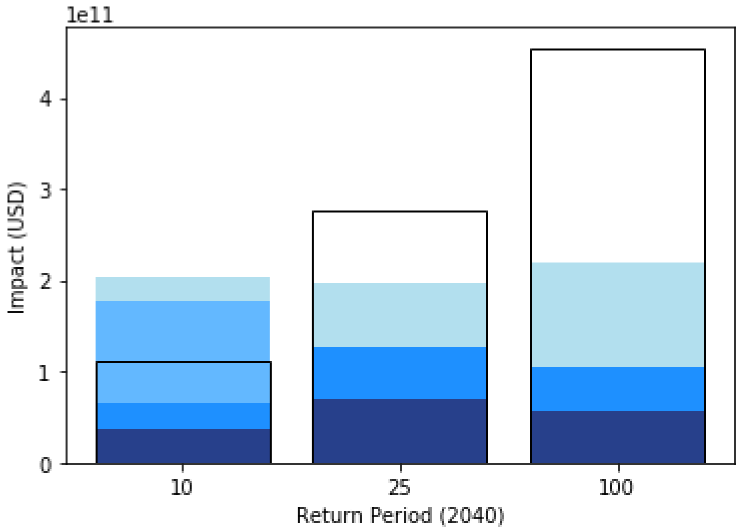
```
2019-11-13 12:12:22,385 - climada.entity.exposures.base - INFO - category_id not set.
2019-11-13 12:12:22,387 - climada.entity.exposures.base - INFO - region_id not set.
2019-11-13 12:12:22,388 - climada.entity.exposures.base - INFO - geometry not set.
2019-11-13 12:12:22,402 - climada.engine.impact - INFO - Exposures matching centroids␣
↪found in centr_TC
2019-11-13 12:12:22,405 - climada.engine.impact - INFO - Calculating damage for 50␣
↪assets (>0) and 14450 events.
2019-11-13 12:12:22,435 - climada.engine.impact - INFO - Exposures matching centroids␣
↪found in centr_TC
2019-11-13 12:12:22,438 - climada.engine.impact - INFO - Calculating damage for 50␣
↪assets (>0) and 14450 events.
2019-11-13 12:12:22,512 - climada.engine.impact - INFO - Exposures matching centroids␣
↪found in centr_TC
2019-11-13 12:12:22,513 - climada.engine.impact - INFO - Calculating damage for 50␣
↪assets (>0) and 14450 events.
2019-11-13 12:12:22,579 - climada.engine.impact - INFO - Exposures matching centroids␣
↪found in centr_TC
2019-11-13 12:12:22,581 - climada.engine.impact - INFO - Calculating damage for 50␣
↪assets (>0) and 14450 events.
2019-11-13 12:12:22,629 - climada.engine.impact - INFO - Exposures matching centroids␣
↪found in centr_TC
2019-11-13 12:12:22,631 - climada.engine.impact - INFO - Calculating damage for 50␣
↪assets (>0) and 14450 events.
2019-11-13 12:12:22,774 - climada.engine.impact - INFO - Exposures matching centroids␣
↪found in centr_TC
2019-11-13 12:12:22,778 - climada.engine.impact - INFO - Calculating damage for 50␣
↪assets (>0) and 14450 events.
2019-11-13 12:12:22,871 - climada.entity.exposures.base - INFO - Matching 50␣
↪exposures with 100 centroids.
2019-11-13 12:12:22,877 - climada.engine.impact - INFO - Calculating damage for 50␣
↪assets (>0) and 14450 events.
2019-11-13 12:12:22,905 - climada.engine.impact - INFO - Exposures matching centroids␣
↪found in centr_TC
2019-11-13 12:12:22,908 - climada.engine.impact - INFO - Calculating damage for 50␣
↪assets (>0) and 14450 events.
2019-11-13 12:12:22,995 - climada.engine.impact - INFO - Exposures matching centroids␣
↪found in centr_TC
2019-11-13 12:12:22,997 - climada.engine.impact - INFO - Calculating damage for 50␣
↪assets (>0) and 14450 events.
2019-11-13 12:12:23,080 - climada.engine.impact - INFO - Exposures matching centroids␣
↪found in centr_TC
2019-11-13 12:12:23,082 - climada.engine.impact - INFO - Calculating damage for 50␣
↪assets (>0) and 14450 events.
2019-11-13 12:12:23,142 - climada.engine.impact - INFO - Exposures matching centroids␣
↪found in centr_TC
2019-11-13 12:12:23,144 - climada.engine.impact - INFO - Calculating damage for 50␣
↪assets (>0) and 14450 events.
2019-11-13 12:12:23,301 - climada.engine.impact - INFO - Exposures matching centroids␣
↪found in centr_TC
2019-11-13 12:12:23,303 - climada.engine.impact - INFO - Calculating damage for 50␣
↪assets (>0) and 14450 events.
2019-11-13 12:12:23,394 - climada.engine.cost_benefit - INFO - Computing cost benefit␣
↪from years 2018 to 2040.

Measure               Cost (USD bn)    Benefit (USD bn)    Benefit/Cost
-----------  ----------  ------------  ---------
Mangroves                   1.31177           80.0097        60.9938
```

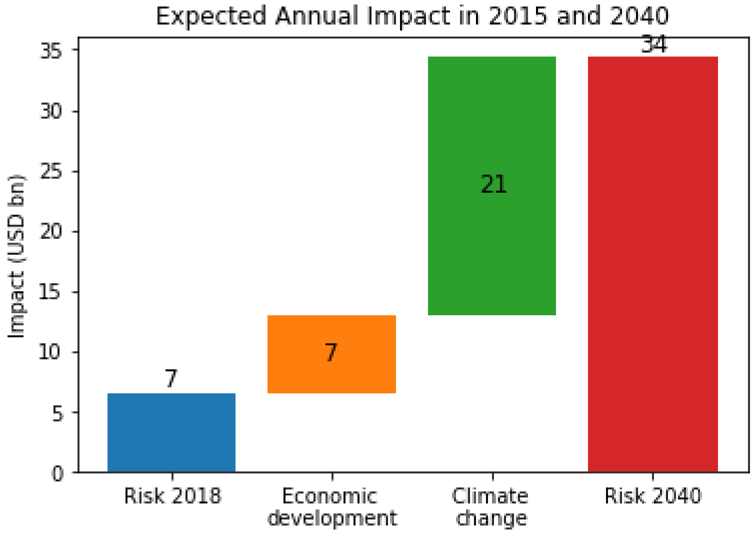
(continues on next page)

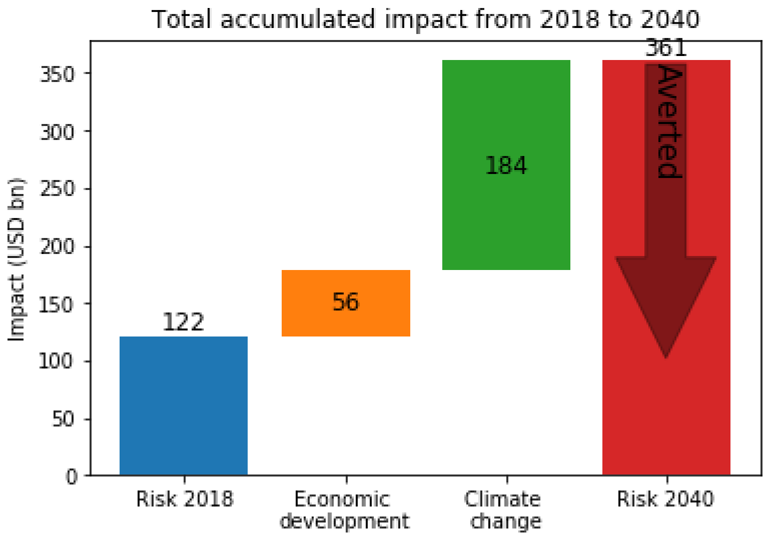
(continued from previous page)

```
Beach nourishment          1.728               63.3336        36.6514
Seawall                    8.87878            164.132         18.4858
Building code              9.2                 90.2786         9.81289

-------------  -----  -----
Total climate risk:   361.115   (USD bn)
Average annual risk:   34.3977  (USD bn)
Residual risk:        -36.6389  (USD bn)
-------------  -----  -----
Net Present Values
2019-11-13 12:12:23,451 - climada.engine.impact - INFO - Exposures matching centroids␣
↪found in centr_TC
2019-11-13 12:12:23,452 - climada.engine.impact - INFO - Calculating damage for 50␣
↪assets (>0) and 14450 events.
2019-11-13 12:12:23,481 - climada.engine.impact - INFO - Exposures matching centroids␣
↪found in centr_TC
2019-11-13 12:12:23,483 - climada.engine.impact - INFO - Calculating damage for 50␣
↪assets (>0) and 14450 events.
2019-11-13 12:12:23,521 - climada.engine.cost_benefit - INFO - Risk at 2018: 6.512e+09
2019-11-13 12:12:23,523 - climada.engine.impact - INFO - Exposures matching centroids␣
↪found in centr_TC
2019-11-13 12:12:23,526 - climada.engine.impact - INFO - Calculating damage for 50␣
↪assets (>0) and 14450 events.
2019-11-13 12:12:23,552 - climada.engine.cost_benefit - INFO - Risk with development␣
↪at 2040: 1.302e+10
2019-11-13 12:12:23,553 - climada.engine.cost_benefit - INFO - Risk with development␣
↪and climate change at 2040: 3.440e+10
2019-11-13 12:12:23,568 - climada.engine.cost_benefit - INFO - Current total risk at␣
↪2040: 1.215e+11
2019-11-13 12:12:23,569 - climada.engine.impact - INFO - Exposures matching centroids␣
↪found in centr_TC
2019-11-13 12:12:23,571 - climada.engine.impact - INFO - Calculating damage for 50␣
↪assets (>0) and 14450 events.
2019-11-13 12:12:23,606 - climada.engine.cost_benefit - INFO - Total risk with␣
↪development at 2040: 1.775e+11
2019-11-13 12:12:23,608 - climada.engine.cost_benefit - INFO - Total risk with␣
↪development and climate change at 2040: 3.611e+11
2019-11-13 12:12:23,639 - climada.engine.cost_benefit - INFO - Combining measures␣
↪['Mangroves', 'Beach nourishment', 'Seawall', 'Building code']

Measure      Cost (USD bn)    Benefit (USD bn)    Benefit/Cost
------  ----------  ------------  ---------
combine           21.1185             262.731        12.4408

-------------  -----  -----
Total climate risk:   361.115   (USD bn)
Average annual risk:   34.3977  (USD bn)
Residual risk:         98.3835  (USD bn)
-------------  -----  -----
Net Present Values
```


Benefit/Cost ratio
0
10
20
30
40
50
60
Mangroves
AAI
Beach nourishment
Seawall
Building code
Tot risk
0
50
100
150
200
250
300
350
NPV averted damage over 23 years (USD bn)


1e11
Impact (USD)
0
1
2
3
4
10
25
100
Return Period (2040)


Expected Annual Impact in 2015 and 2040
Impact (USD bn)
0
5
10
15
20
25
30
35
7
7
21
34
Risk 2018
Economic development
Climate change
Risk 2040



Check what happens when different parameters are changed, such as the `imp_time_depen` and `risk_func` in CostBenefit.calc() (and plot_waterfall(), plot_waterfall_accumulated())

3.4 Your case

1. Build an entity. It might be one from your previous runs in MATLAB. Make sure i'is saved in version > v7.3 if it's a MATLAB file. If it's not, you'll get an error message. Then, you can save it again in MATLAB like that: `save('file_name.mat','variable_name', '-v7.3')`
2. Build a hazard. It might also come from a previous run in MATLAB. This file might already contain the centroids. If not, define the centroids as well and use them in your calculations.
3. Compute the impact.
4. Visualization. Plot:

- the damage functions for the hazard
- the entity values map
- the strongest event intensity
- the maximum hazard intensity of all the events in Zürich (47.38, 8.54)
- the impact exceedence frequency curve

```
[12]: # Put your code here
```

```
[13]: # SOLUTION: example: winter storms in europe
from climada.util import DATA_DIR
import pandas as pd
from climada.hazard import Hazard
```

(continues on next page)

(continued from previous page)

```
from climada.entity import Exposures, ImpactFuncSet
from climada.engine import Impact

# Put any absoulte path for your files or set up the configuration variable
↪"repository"
FILE_HAZARD = DATA_DIR + '/demo/WS_ERA40.mat'
FILE_ENTITY = DATA_DIR + '/demo/WS_Europe.xls'

# Define hazard type
HAZ_TYPE = 'WS'

# 1. Entity: we only need impact functions and exposures to compute the impact
# Exposures
exp_ws_eu = pd.read_excel(FILE_ENTITY)
exp_ws_eu = Exposures(exp_ws_eu)
exp_ws_eu.check()

# Impact functions
impf_ws_eu = ImpactFuncSet()
impf_ws_eu.read_excel(FILE_ENTITY, 'Impact functions for winter storms in EU.')

# 2. Hazard
haz_ws_eu = Hazard(HAZ_TYPE)
haz_ws_eu.read_mat(FILE_HAZARD, 'WS EU ERA 40')

# 3. Impact
imp_ws_eu = Impact()
imp_ws_eu.calc(exp_ws_eu, impf_ws_eu, haz_ws_eu)

# 4.
# the damage functions for the hazard
impf_ws_eu.plot()

# the exposures values map
exp_ws_eu.plot_hexbin(pop_name=False)

# the strongest event
haz_ws_eu.plot_intensity(-1) # might be better to use an other earth projection?

# the impact exceedence frequency curve
imp_exc_curve = imp_ws_eu.calc_freq_curve()
imp_exc_curve.plot()
```

```
2019-11-13 12:12:24,344 - climada.entity.exposures.base - INFO - crs set to default␣
↪value: {'init': 'epsg:4326', 'no_defs': True}
2019-11-13 12:12:24,344 - climada.entity.exposures.base - INFO - tag metadata set to␣
↪default value:  File:
 Description:
2019-11-13 12:12:24,345 - climada.entity.exposures.base - INFO - ref_year metadata␣
↪set to default value: 2018
2019-11-13 12:12:24,346 - climada.entity.exposures.base - INFO - value_unit metadata␣
↪set to default value: USD
2019-11-13 12:12:24,346 - climada.entity.exposures.base - INFO - meta metadata set to␣
↪default value: None
2019-11-13 12:12:24,348 - climada.entity.exposures.base - INFO - centr_ not set.
2019-11-13 12:12:24,349 - climada.entity.exposures.base - INFO - category_id not set.
2019-11-13 12:12:24,349 - climada.entity.exposures.base - INFO - region_id not set.
```

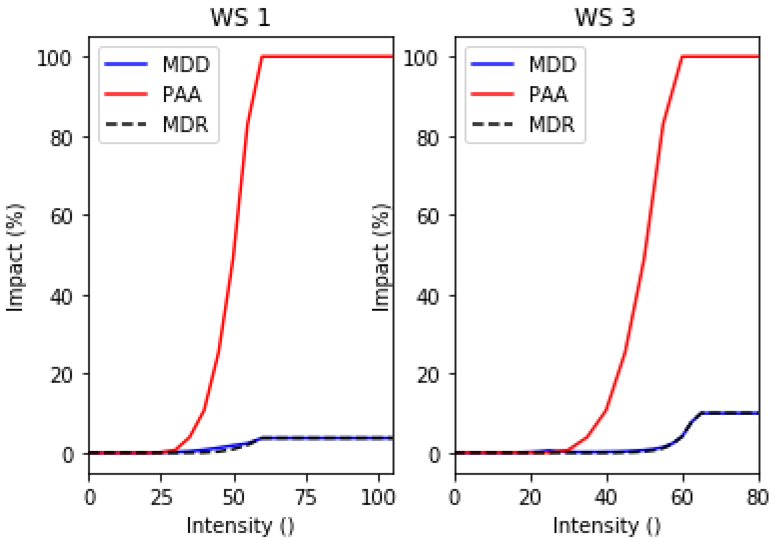
(continues on next page)

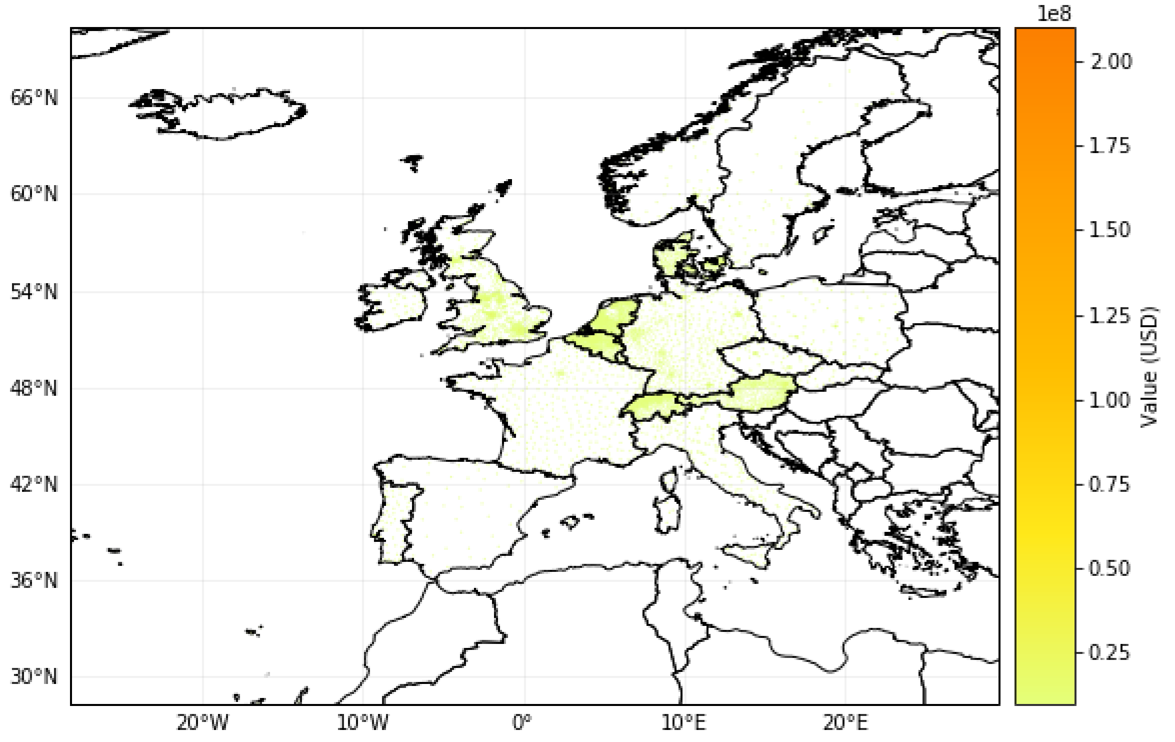
(continued from previous page)

```
2019-11-13 12:12:24,350 - climada.entity.exposures.base - INFO - geometry not set.
2019-11-13 12:12:24,579 - climada.hazard.base - INFO - Reading /Users/aznarsig/
↪Documents/Python/climada_python/data/demo/WS_ERA40.mat
2019-11-13 12:12:24,804 - climada.hazard.centroids.centr - INFO - Reading /Users/
↪aznarsig/Documents/Python/climada_python/data/demo/WS_ERA40.mat
2019-11-13 12:12:25,122 - climada.entity.exposures.base - INFO - Matching 6186␣
↪exposures with 6331 centroids.
2019-11-13 12:12:25,811 - climada.engine.impact - INFO - Calculating damage for 6186␣
↪assets (>0) and 1755 events.
```

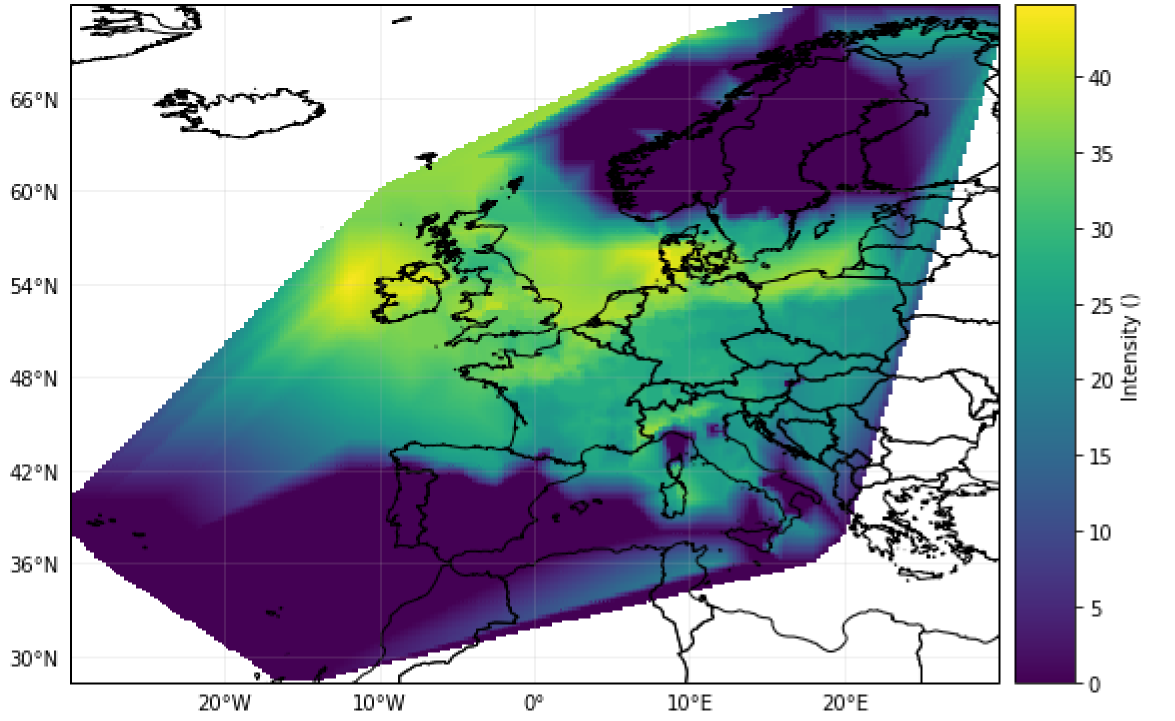
```
/Users/aznarsig/Documents/Python/climada_python/climada/util/plot.py:318:␣
↪UserWarning: Tight layout not applied. The left and right margins cannot be made␣
↪large enough to accommodate all axes decorations.
  fig.tight_layout()
```

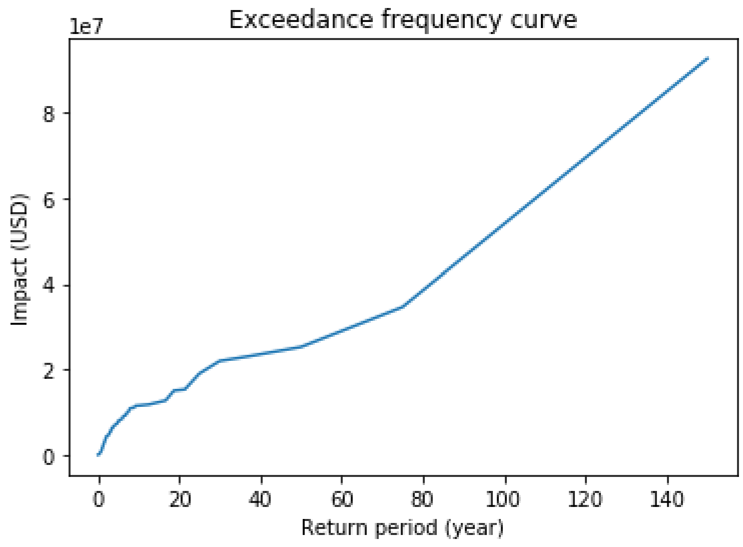
```
[13]: <matplotlib.axes._subplots.AxesSubplot at 0x1c28595940>
```





1-largest Event. ID 1714: 1714




Exceedance frequency curve
1e7
Impact (USD)
0
2
4
6
8
0
20
40
60
80
100
120
140
Return period (year)

CHAPTER

FOUR

DATA DEPENDENCIES

4.1 Web APIs

CLIMADA relies on open data available through web APIs such as those of the World Bank, Natural Earth, NASA and NOAA. You might execute the test `climada_python-x.y.z/test_data_api.py` to check that all the APIs used are active. If any is out of service (temporarily or permanently), the test will indicate which one.

4.2 Manual download

As indicated in the software and tutorials, other data might need to be downloaded manually by the user. The following table shows these last data sources, their version used, its current availabilty and where they are used within CLIMADA:

Availability	Name	Version	Link	CLIMADA class	CLIMADA version	CLIMADA tutorial reference
OK	Fire Information for Resource Management System	•	FIRMS	BushFire	> v1.2.5 | climada_hazard_BushFire.ipynb	
OK	Gridded Population of the World (GPW)	v4.11	GPW v4.11	LitPop	> v1.2.3	climada_entity_LitPop.ipynb
FAILED	Gridded Population of the World (GPW)	v4.10	GPW v4.10	LitPop	>= v1.2.0	climada_entity_LitPop.ipynb

CHAPTER

FIVE

CONFIGURATION OPTIONS

CLIMADA searches for a local configuration file located in the current working directory. A static default configuration file is supplied by the package and used as fallback. The local configuration file needs to be called `climada.conf`. All other files will be ignored.

The climada configuration file is a JSON file and consists of the following values:

- `config`
- `local_data`
- `global`
- `trop_cyclone`

A minimal configuration file looks something like this:

```
{
    "config":
    {
        "env_name": "climada_env"
    },

    "local_data":
    {
        "save_dir": "./results/"
    },

    "global":
    {
        "log_level": "INFO",
        "max_matrix_size": 1.0e8
    },

    "trop_cyclone":
    {
        "random_seed": 54
    }
}
```

5.1 config

Configuration parameters related with configuration settings such as paths.

Option	Description	Default
`env_name`	Name given to CLIMADA's virtual environement. Used for checks of paths of libraries.	"climada_env"

5.2 local_data

Configuration parameters related to local data location.

Option	Description	Default
`save_dir`	Folder were the variables are saved through the `save` command when no absolute path provided.	"./results"

5.3 global

Configuration parameters with global scope within climada's code.

Option	Description	Default
`log_level`	Minimum log level showed by logging: DEBUG, INFO, WARNING, ERROR or CRITICAL.	"INFO"
`max_matrix_size`	Maximum matrix size that can be used. Set a lower value if memory issues.	1.0E8

5.4 trop_cyclone

Configuration parameters related to tropical cyclones.

Option	Description	Default
`random_seed`	Seed used for the stochastic tracks generation.	54

CHAPTER

SIX

CONTRIBUTING

Contributions are very welcome! Please follow these steps:

0. **Install** Git and Anaconda (or Miniconda).

1. **Fork** the project on GitHub:

   ```
   git clone https://github.com/CLIMADA-project/climada_python.git
   ```

2. **Install the packages** in `climada_python/requirements/env_climada.yml` and `climada_python/requirements/env_developer.yml` (see *Installation*). You might need to install additional environments contained in `climada_python/requirements` when using specific functionalities.

3. You might make a new **branch** if you are modifying more than one part or feature:

   ```
   git checkout -b feature_branch_name
   ```

   About branches.

4. Write small readable methods, classes and functions. Make well commented and clean **commits** to the repository:

   ```
   git pull
   git stats          # use it to see your locally modified files
   git add climada/modified_file.py climada/test/test_modified_file.py
   git commit -m "new functionality of .. implemented"
   ```

5. Make unit and integration **tests** on your code, preferably during development:

   - Unit tests are located in the `test` folder located in same folder as the corresponding module. Unit tests should test all methods and functions using fake data if necessary. The whole test suit should run in less than 20 sec. They are all executed after each push in Jenkins.

   - Integration tests are located in `climada/test/`. They test end-to-end methods and functions. Their execution time can be of minutes. They are executed once a day in Jenkins.

6. Perform a **static code analysis** of your code using `pylint` with CLIMADA's configuration `.pylintrc`. Jenkins executes it after every push. To do it locally, you might use the Interface provided by *Spyder*. To do so, search first for *static code analysis* in *View* and then *Panes*.

7. Add new **data dependencies** used in *Data dependencies* and write a **tutorial** if a new class has been introduced (see *Tutorial*).

8. Add your name to the **AUTHORS** file.

9. **Push** the code or branch to GitHub. To push without a branch (to master) do so:

```
git push
```

To push to your branch `feature_branch_name` do:

```
git push origin feature_branch_name
```

When the branch is ready, create a new **pull request** from the feature branch. About pull requests.

6.1 Notes

6.1.1 Update CLIMADA's environment

Remember to regularly update your code as well as climada's environment. You might use the following commands to update the environments:

```
cd climada_python
git pull
source activate climada_env
conda env update --file requirements/env_climada.yml
conda env update --file requirements/env_developer.yml
```

If any problem occurs during this process, consider reinstalling everything from scratch following the :doc:install instructions. You can find more information about virtual environments with conda here.

CHAPTER

SEVEN

SOFTWARE DOCUMENTATION

Documents functions, classes and methods:

7.1 Software documentation per package

7.1.1 climada.engine package

climada.engine.impact module

climada.engine.cost_benefit module

7.1.2 climada.entity package

climada.entity.disc_rates package

climada.entity.disc_rates.base module

climada.entity.exposures package

climada.entity.exposures.base module

climada.entity.exposures.black_marble module

climada.entity.exposures.litpop module

climada.entity.exposures.open_street_map module

climada.entity.impact_funcs package

climada.entity.impact_funcs.base module

climada.entity.impact_funcs.impact_func_set module

climada.entity.impact_funcs.trop_cyclone module

climada.entity.measures package

climada.entity.measures.base module

climada.entity.measures.measure_set module

climada.entity.entity_def module

climada.entity.tag module

class climada.entity.tag.**Tag**(*file_name=''*, *description=''*)

Bases: `object`

Source data tag for Exposures, DiscRates, ImpactFuncSet, MeasureSet.

file_name

name of the source file

Type str

description

description of the data

Type str

__init__(*file_name=''*, *description=''*)

Initialize values.

Parameters

- **file_name** (*str, optional*) – file name to read
- **description** (*str, optional*) – description of the data

append(*tag*)

Append input Tag instance information to current Tag.

7.1.3 climada.hazard package

climada.hazard.centroids package

climada.hazard.centroids.centr module

climada.hazard.base module

climada.hazard.tag module

climada.hazard.trop_cyclone module

climada.hazard.tc_tracks module

7.1.4 climada.util package

climada.util.checker module

climada.util.config module

climada.util.config.**setup_logging**(*log_level='DEBUG'*)
Setup logging configuration

climada.util.config.**setup_conf_user**()
Setup climada configuration

climada.util.config.**setup_environ**()
Parse binary environment and correct if necessary

climada.util.constants module

climada.util.constants.**SOURCE_DIR = '/home/docs/checkouts/readthedocs.org/user_builds/clima**
climada directory

climada.util.constants.**DATA_DIR = '/home/docs/checkouts/readthedocs.org/user_builds/climada**
Folder containing the data

climada.util.constants.**SYSTEM_DIR = '/home/docs/checkouts/readthedocs.org/user_builds/clima**
Folder containing the data used internally

climada.util.constants.**GLB_CENTROIDS_NC = '/home/docs/checkouts/readthedocs.org/user_builds**
Global centroids nc.

climada.util.constants.**GLB_CENTROIDS_MAT = '/home/docs/checkouts/readthedocs.org/user_buil**
Global centroids.

climada.util.constants.**ENT_TEMPLATE_XLS = '/home/docs/checkouts/readthedocs.org/user_builds**
Entity template in xls format.

climada.util.constants.**NAT_REG_ID = '/home/docs/checkouts/readthedocs.org/user_builds/clima**
Look-up table ISO3 codes

climada.util.constants.**HAZ_DEMO_FLDDPH = '/home/docs/checkouts/readthedocs.org/user_builds/**
NetCDF4 Flood depth from isimip simulations

climada.util.constants.**HAZ_DEMO_FLDFRC = '/home/docs/checkouts/readthedocs.org/user_builds/**
NetCDF4 Flood fraction from isimip simulations

climada.util.constants.**HAZ_DEMO_MAT = '/home/docs/checkouts/readthedocs.org/user_builds/cl**
hurricanes from 1851 to 2011 over Florida with 100 centroids.

Type Hazard demo from climada in MATLAB

climada.util.constants.**HAZ_DEMO_H5 = '/home/docs/checkouts/readthedocs.org/user_builds/cli**
ibtracs from 1975 to 2011 over Florida with 2500 centroids.

Type Hazard demo in h5 format

climada.util.constants.**DEMO_GDP2ASSET = '/home/docs/checkouts/readthedocs.org/user_builds/c**
Exposure demo file for GDP2Asset

climada.util.constants.**WS_DEMO_NC = ['/home/docs/checkouts/readthedocs.org/user_builds/cli**
Winter storm in Europe files. These test files have been generated using the netCDF kitchen sink: ncks -d latitude,50.5,54.0 -d longitude,3.0,7.5 ./file_in.nc ./file_out.nc

climada.util.constants.**EXP_DEMO_H5 = '/home/docs/checkouts/readthedocs.org/user_builds/cli**
Exposures over Florida

climada.util.constants.**TC_ANDREW_FL = '/home/docs/checkouts/readthedocs.org/user_builds/cl**
Tropical cyclone Andrew in Florida

`climada.util.constants.`**`ONE_LAT_KM = 111.12`**

Mean one latitude (in degrees) to km

`climada.util.constants.`**`EARTH_RADIUS_KM = 6371`**

Earth radius in km

climada.util.coordinates module

climada.util.files_handler module

climada.util.hdf5_handler module

climada.util.interpolation module

climada.util.plot module

climada.util.save module

- genindex
- modindex

CHAPTER EIGHT

LICENSE

BIBLIOGRAPHY

[IPCC2014] IPCC: Climate Change 2014: Impacts, Adaptation and Vulnerability. Part A: Global and Sectoral Aspects. Contribution of Working Group II to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change, edited by C. B. Field, V. R. Barros, D. J. Dokken, K. J. Mach, M. D. Mastrandrea, T. E. Bilir, M. Chatterjee, K. L. Ebi, Y. O. Estrada, R. C. Genova, B. Girma, E. S. Kissel, A. N. Levy, S. MacCracken, P. R. Mastrandrea, and L. L. White, Cambridge University Press, United Kingdom and New York, NY, USA., 2014.

PYTHON MODULE INDEX

INDEX