climada module **country risk** 15 July 2017

https://github.com/davidnbresch/climada_module_country_risk
david.bresch@gmail.com
melanie.bieli@bluewin.ch (2014)

This module runs all (available) perils for one country or for a list of countries. First, it allows generating the country assets (10 or 1 km resolution, based on night light intensity[1]), second to generate the hazard event sets and second to run all damage calculations.

Currently implemented is the automatic generation of earthquake (EQ), volcano (VQ), tropical cyclone (TC), torrential rain (TR), storm surge (TS) and European winter storm (WS) hazard event sets. The core function is `country_risk_calc`, which does it all in one go[2]. Instead of whole countries, one can also analyze single states/provinces, see `country_admin1_risk_calc`. The module does contain a series of support functions to calibrate country results – they are named `cr_*`, while the top-level functions of this module are named `country_*` or `country_risk_*`, see function reference below. The batch code `selected_countries_all_in_one` does provide the reference example about how to use all these functions.

Further, the module calculates the economic loss (i.e. the full range of economic costs in the wake of a natural disaster) associated with the hazard event sets[3].

Content

Table of Contents

[1] I.e. the module allows the generate centroids and entities for each country on these (high) spatial resolution, see `climada_nightligth_entity` (described further below).
[2] The routine `climada_country_risk` allows for processing a list or even all countries. As always, use e.g. `help climada_country_risk` to get a detailed description on the options.
[3] See appendix for details on the calculation of economic loss based on the damages in the hazard event set.

Country risk module – basics

The whole module introduces the structure country_risk, the core output of `country_risk_calc` and `country_admin1_risk_calc` which contains country results in a standardized way and is therefore input to a series of functions of this module. `country_risk(i).res.hazard(j)` contains the EDS for country i and hazard j (`country_risk(i).res.hazard(j).EDS`), the name of the peril in `country_risk(i).res.hazard(j).peril_ID`, the name of the entity in `country_risk(i).res.hazard(j).entity_file` and the name of the hazard event set in `country_risk(i).res.hazard(j).hazard_set_file`.

country_risk_calc

Basic procedure implemented in `country_risk_calc` is as follows:
1) generate centroids for the country (uses `climada_create_GDP_entity`[4] or `climada_nightlight_entity`)
2) figure which hazards affect the country
3) create the hazard event sets, using
   - climada_tc_hazard_set (tropical cyclone wind[5])
   - climada_tr_hazard_set (tropical cyclone rain[6])
   - climada_ts_hazard_set (tropical cyclone surge[7])
   - eq_global_hazard_set (earthquake[8])
   - European winter storm (hazard not generated, just assigned[9])
4) run the risk calculation for all hazards

Next steps are `country_risk_report`, `country_risk_EDS_combine` and `country_risk_EDS2YDS` plus `cr_plot_DFC` or `cr_plot_DFC_aggregate` to plot the resulting country damage frequency curves.

In essence, you define the country and the code runs the generation of centroids, default assets (from nightlight intensity, see climada module GDP_entity) and the EQ, TC, TR and TS hazard event sets plus checks for WS Europe exposure. It even figures whether the country is exposed to more than one ocean basin and in such a case generates a suite of TC/TS/TR hazard event

[4] See https://github.com/davidnbresch/climada_module_GDP_entity and further below for `climada_nightlight_entity` which allows to generate a high-resolution entity for any country and state/province.
[5] Core climada contains the basic tropical cyclone hazard, but please add the module https://github.com/davidnbresch/climada_module_tropical_cyclone to generate useful probabilistic hazard event sets (see parameter `probabilistic` in `country_risk_calc`). Please consider to run `climada_tc_get_unisys_databases` (climada core) in order to download the latest tropical cyclone databases for all ocean basins (core climate comes with TC Atlantic to start with).
[6] See climada module https://github.com/davidnbresch/climada_module_tropical_cyclone
[7] See climada module https://github.com/davidnbresch/climada_module_tropical_cyclone which also requires the module https://github.com/davidnbresch/climada_module_etopo
[8] See climada module https://github.com/davidnbresch/climada_module_earthquake_volcano
[9] See climada module for European winter storm, which contains the hazard sets https://github.com/davidnbresch/climada_module_storm_europe

sets for each ocean basin. The code is ready for upgrade with additional hazards (usually a new hazard is a new climada module). That's why the code notifies the user if the specific hazard module is missing (even indicates the github location where to get it from).

Simply call e.g. `country_risk_calc('El Salvador')`
If called without any argument, a list dialog to select the country pops up. See code header for details, i.e. `help country_risk_calc`

Behind the scenes, the code `centroids_generate_hazard_sets` does the heavy lifting, i.e. steps 2) and 3) from above. This way one can generate all relevant hazard sets with one call to `centroids_generate_hazard_sets` for any set of centroids (e.g. only a part of a country, a region, a city…[10]):

```
centroids_hazard_info=...
   centroids_generate_hazard_sets(centroids,force_recalc,check_plots)
```

The resulting structure `centroids_hazard_info` contains the names of the generated hazard sets (or the ones generated earlier if just called to check for step 2) in `centroids_hazard_info.res.hazard(i).hazard_set_file` (the somewhat complicated nested structure is due to the flexibility required by `country_risk_calc`).

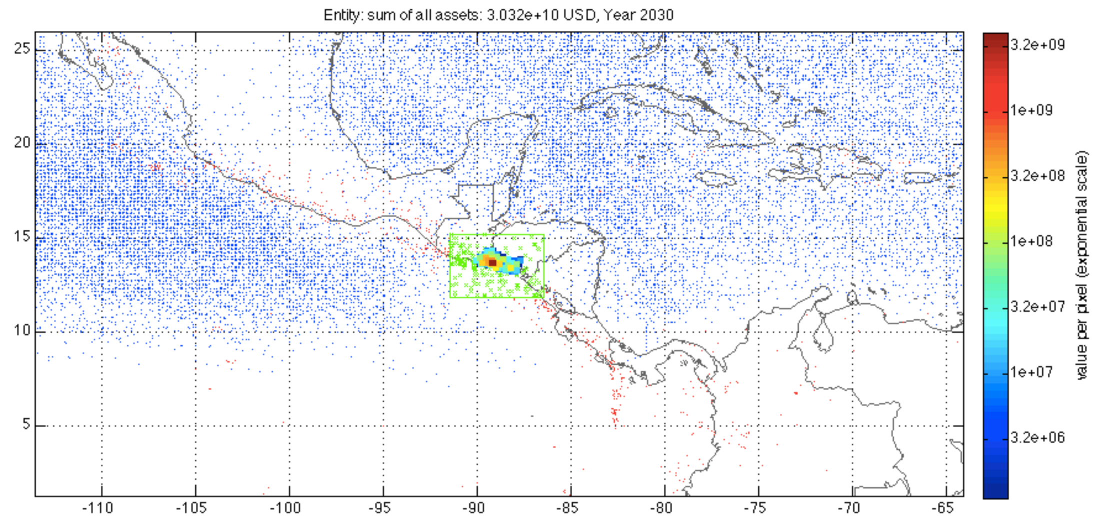

Figure: Step 1 (generate centroids, assets distribution, color scale indicates value per centroid) and step 2 (hazard selection). The green box shows the selection area around the country, the blue dots are all the TC track nodes (historic) and the red dots the epicenters (historic). This figure is generated if check_plot=1 in the call, e.g. `country_risk_calc('El Salvador',0,0,1)`.

[10] See e.g. the code `climada_cut_out_GDP_entity` from https://github.com/davidnbresch/climada_module_GDP_entity and also `country_admin1_risk_calc` and `climada_nightlight_entity` further below (part of module country risk)

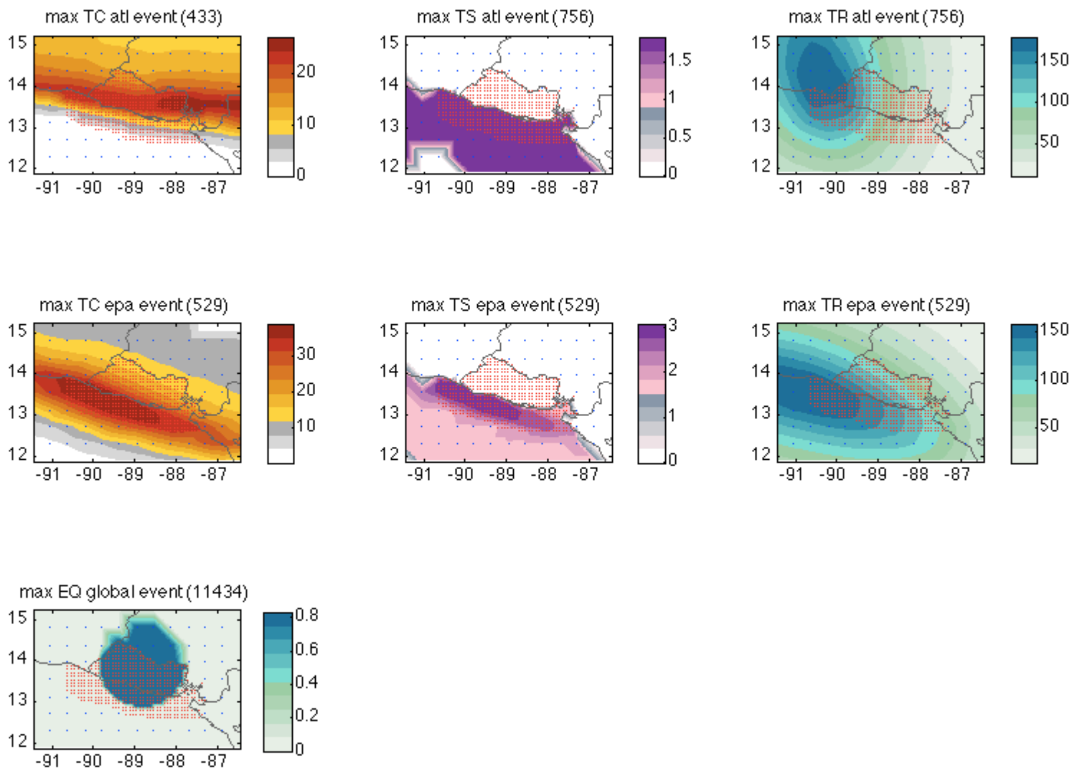

Figure: The most intense single event for all hazard generated for El Salvador (Step 3). Note that El Salvador is both exposed to tropical cyclones from the East and West, that's why there are two hazard events sets for TC/TS/TR, one for the Atlantic side (atl for Atlantic), one for the Pacific side (epa for East Pacific Ocean). The earthquake model is global. Note further the nice feature of hazard (or peril) – dependent color scales; and the coarser resolution of centroids (blue) around the country (with red dots at high-density centroids) to support plotting hazard intensities around the country, too. This figure is generated if `check_plot=1` in the call, e.g. `country_risk_calc('El Salvador',0,0,1)`

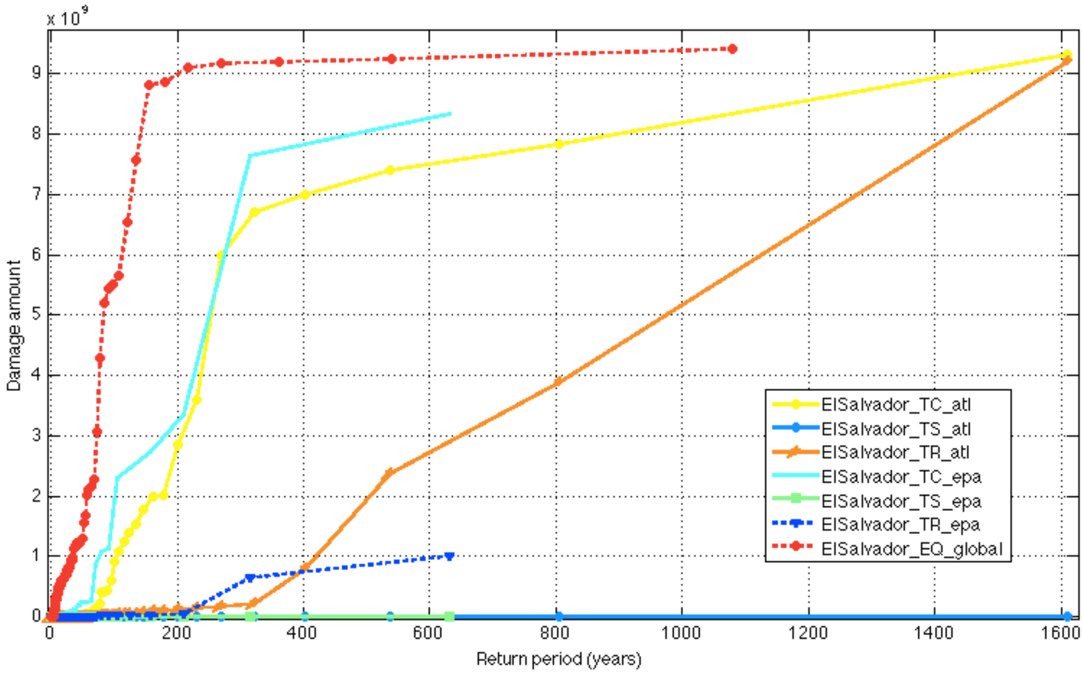

Figure: The resulting damage frequency curves (DFC) for all seven (!) hazards affecting El Salvador (values just for illustration, based on dummy damage functions). This figure is generated with check_plot=1 in the call, e.g.
`country_risk_report(country_risk_calc('El Salvador',1,1),0,1)`

country_risk_report

Comes in handy if one runs some select countries, e.g:

```
country_risk=country_risk_calc('Barbados')
country_risk(2)=country_risk_calc('El Salvador')
country_risk(3)=country_risk_calc('Costa Rica')
```

And then `country_risk_report(country_risk,0)` results in the following output (to stdout, also an Excel or .csv file is written[11]):

```
Barbados (1)
  TR EL=36572051.496470        (8.481508%o)          Barbados_TR_atl
  TC EL=23083330.494007        (5.353308%o)          Barbados_TC_atl
  TS EL=7531.966739            (0.001747%o)          Barbados_TS_atl
  EQ EL=0.000000               (0.000000%o)          Barbados_EQ_global
ElSalvador (2)
  EQ EL=415631535.361110       (17.943889%o)         ElSalvador_EQ_global
  TR EL=141613002.072040       (6.113800%o)          ElSalvador_TR_epa
  TC EL=59386249.565168        (2.563858%o)          ElSalvador_TC_atl
  TC EL=16152772.894979        (0.697357%o)          ElSalvador_TC_epa
  TR EL=621784.438763          (0.026844%o)          ElSalvador_TR_atl
  TS EL=0.000000               (0.000000%o)          ElSalvador_TS_epa
  TS EL=0.000000               (0.000000%o)          ElSalvador_TS_atl
Costa Rica (3)
  EQ EL=523833928.441207       (12.396559%o)         Costa Rica_EQ_global
  TR EL=1530537.767294         (0.036220%o)          Costa Rica_TR_epa
  TC EL=73978.520263           (0.001751%o)          Costa Rica_TC_epa
  TR EL=5765.009179            (0.000136%o)          Costa Rica_TR_atl
  TS EL=1689.347413            (0.000040%o)          Costa Rica_TS_atl
  TC EL=60.830655              (0.000001%o)          Costa Rica_TC_atl
  TS EL=0.000000               (0.000000%o)          Costa Rica_TS_epa
```

[11] The report does contain the annual expected damage (ED) as well as defined return periods (such as 100 and 250 years). In case writing an Excel file fails, a .csv file is written.

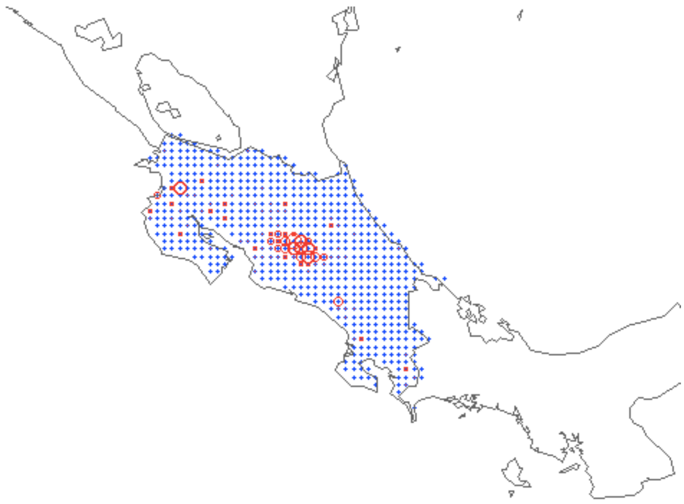

Figure: The local damage for a given peril (here EQ) for one country (here Costa Rica) in spatial resolution (e.g. at each centroid). Produced by the call (following from above resulting structure `country_risk`):

```
country_i=3; hazard_i=7[12];
climada_circle_plot(...
country_risk(country_i).res.hazard(hazard_i).EDS.ED_at_centroid,...
country_risk(country_i).res.hazard(hazard_i).EDS.assets.Longitude,...
country_risk(country_i).res.hazard(hazard_i).EDS.assets.Latitude)
```

country_admin1_risk_calc

Same as country_risk_calc, but for a state or province (admin1 level) of any given country. Run the all (available) perils for one country's admin1 level. Obtains the admin1 boundaries (from www.naturalearthdata.com, shape files already part of the data that comes with the country risk module) and carve out the respective centroids (set Value at all others to zero). Runs the risk calculation for each admin1 for all hazards. In case one would like to skip hazards, just (temporarily) remove the respective {country_IOS3}`_{country_name}_*.mat` hazard event sets or see parameter `peril_ID` in the call to `country_admin1_risk_calc`. ONLY makes sense if `country_risk_calc` has been run for the respective country (we keep it like this, as automatic mode might trigger lots of un-wanted calculations). If not, the code terminates with the respective messages (no entity found, no hazard set(s) found...). But one can run `country_admin1_risk_calc` for more than one country, if the respective countries have been run as `country_risk_calc`. NOTE: Before using this code, make yourself familiar with `country_risk_calc` and `country_risk_report` (same format as `country_risk_calc`).

climada_nightlight_entity

Construct an entity file based on mid (10x10km) or high-res (1x1km!) night light data[13]. Reads an image file with nightlight density[14] and matches it to the local geography. Prompts for country

[12] Note that the number seven here corresponds to the 7th hazard analyzed (EQ). The report to stdout shows EQ as the first result, since `country_risk_report` sorts by descending damage, unless it is called with the second parameter (`print_unsorted`) set to 1.

[13] One can also run it at moderate (10x10km) resolution, see parameter selections, i.e. type `help climada_highres_entity`

[14] The climada module country_risk comes with the .mat file F182012.v4c_web.stable_lights.avg_vis.mat (24MB), since the .tif image is about 700MB. See http://ngdc.noaa.gov/eog/dmsp/downloadV4composites.html#AVSLCFC3 to obtain the file http://ngdc.noaa.gov/eog/data/web_data/v4composites/F182012.v4.tar and unzip the file F182012.v4c_web.stable_lights.avg_vis.tif in there to the /data folder of country_risk module. As the .tif is so much

(admin0) and state/province (admin1), constrains the active centroids (with values>0) to the selected country or admin1 and saves the entity. The original nightlight intensities are first scaled to the range [0..1], then transformed using a polynomial (default to $i=i_{orig}^3$, see Fig below), then scaled such that all values sum up to one (normalized). If admin0 (whole country) is selected, the values are scaled to sum up to country GDP*(income_group+1) as a good proxy for the 'real' asset value (see code `climada_entity_value_GDP_adjust_one`). If admin1, is requested, no automatic scaling or allocation of GDP to centroids is performed.

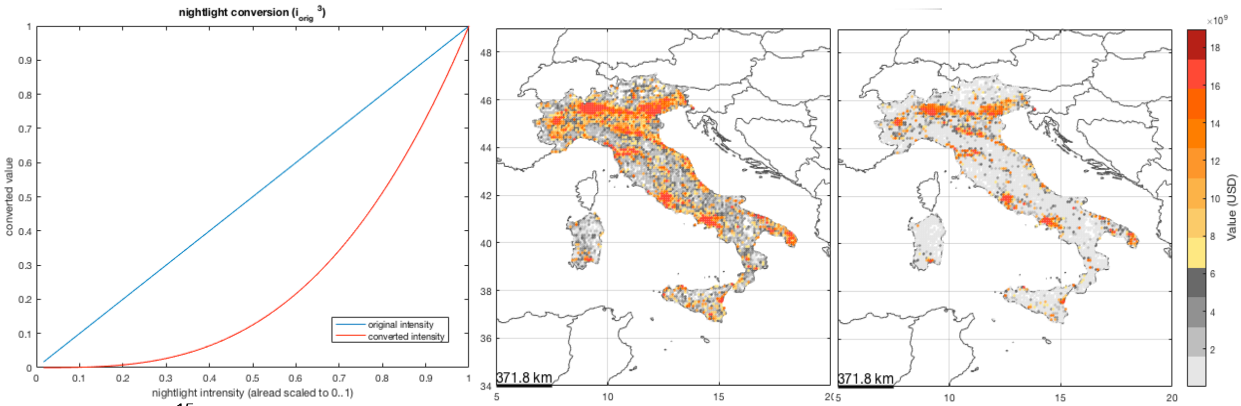

Figure: The old[15] default transformation of nightlight intensity (left) and the effect (middle: linear, no transformation, right: cubic transformation, leading to a concentration of values in more densely populated areas – closer to reality).

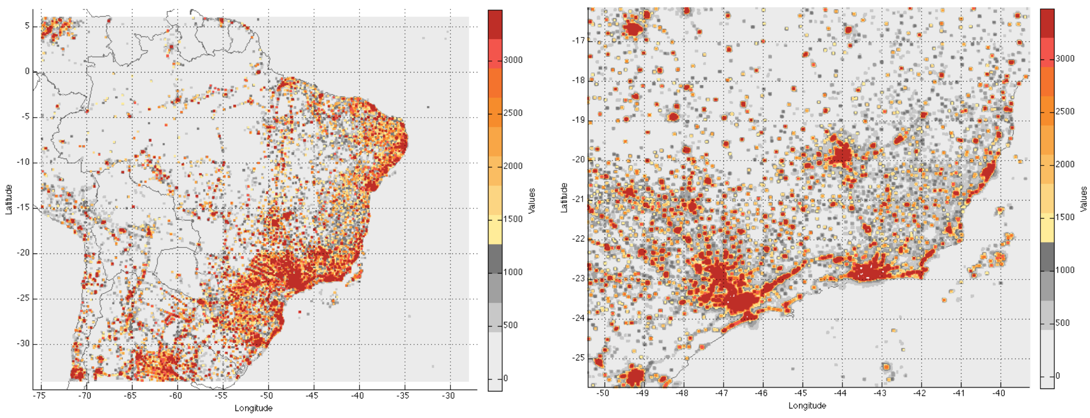

Figure: Brazil value distribution on 1km (!) resolution (left, zoomed in right) as generated by `climada_nightlight_entity('Brazil','')`.

A note on nightlights

Original data from http://ngdc.noaa.gov/eog/data/web_data/v4composites/F182012.v4.tar (see http://ngdc.noaa.gov/eog/dmsp/downloadV4composites.html#AVSLCFC3). Original data range (0..62) scaled to 0..1, the transformed by taking the square (.)^2. Reason for ^2: the intensity of nightlights in cities and densely populated areas does not fully reflect the (much) higher density compared to more rural spots. Hence comparison with other sources (such as insurance

larger, the climada module country_risk comes with the .mat file, but does not contain the original (.tif). Should the .mat file not exist, climada_nightlight_entity creates it on first call.

[15] Until 19 Jan 2017, we used ( )^3, since then we use ( )^2.

portfolio data of housing values) revealed such a scaling to better match with (replacement) values of property on the ground.

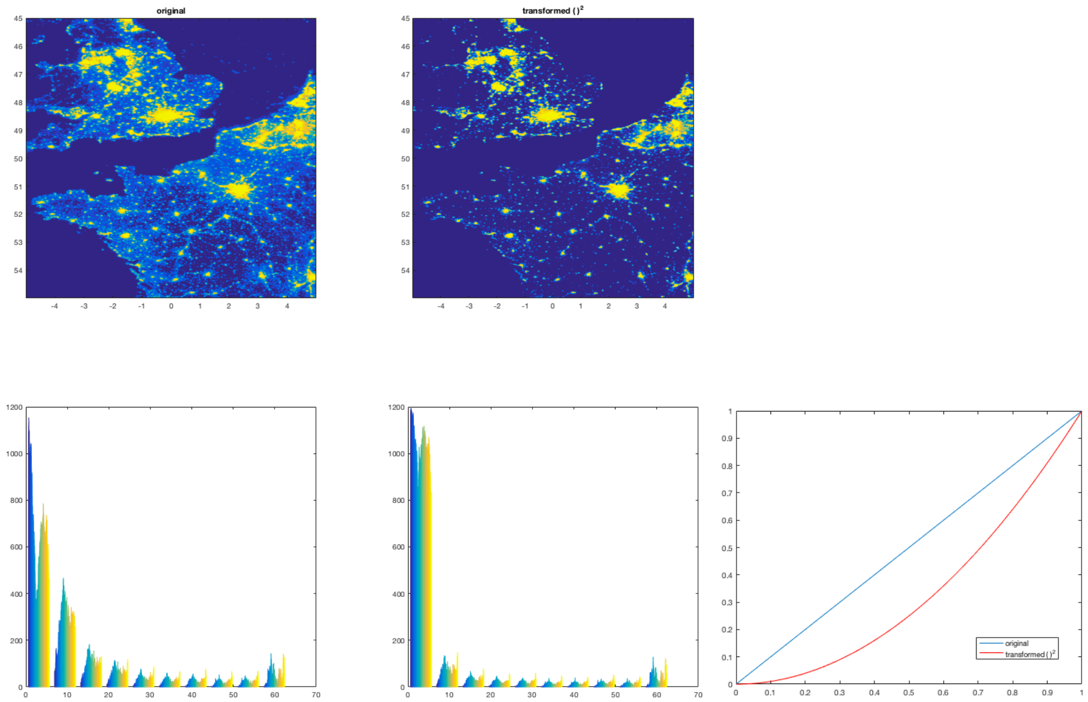

Fig: Nightlight before (left) and after (center) scaling for a region over Europe. Right inset: the transformation. Note that brighter areas do get weighted more heavily this way (as city centers are much more densely populated and built up).

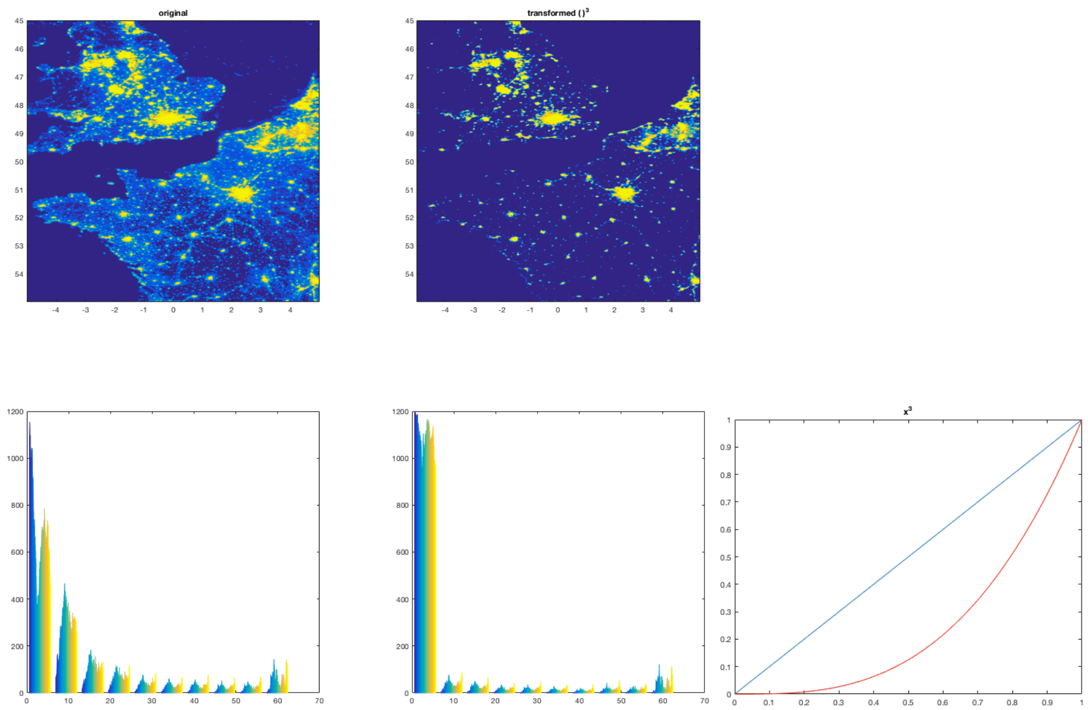

Fig: Same as above, but for ( )^3. Not used, but see parameter `nightlight_transform_poly` in code.

cr_economic_loss_calc

Major natural disasters can and do have severe negative short-run economic impacts, the severity of which depends on the affected country's resilience, or ability to recover. `cr_economic_loss_calc` calculates the economic damages resulting from the simple property damages in the hazard event set, taking into account socio-economic data on the country's financial strength, supply chain risk profile, resilience and preparedness for natural disasters (see appendix for details).

Country and peril calibration

country_risk_calibrate

Calibrate a given country (or a list of countries). Make sure you called country_risk_calc before (not necessarily in the same session, but the country entities need to exist – at least the ones for present ({ISO}_{CountryName}_entity.mat).

The most common use therefore looks like (as an example, generate the asset distributions for listed countries and all the probabilistic hazard sets for all perils that affect any of these countries):

```
country_list={'Colombia','Costa Rica','Dominican Republic','United States'};
country_risk=country_risk_calc(country_list,-3);
country_risk_calibrate(country_list); % calibrate all countries
```

Standard procedure is that the switch statement in the function `country_risk_calibrate` has entries for countries (and lists of countries) and hence performs the specific adjustments, primarily to the damage function(s).

In case you consider editing this function, be careful to check for repetitious application. The code sets the field `entity.calibrated=1` the first time a country is treated, but since one might need to re-calibrate, one should rather assign absolute values to e.g. `entity.damagefunctions.MDD`, since a mere multiplication of existing values might lead to troubles on subsequent calls. Luckily, `country_risk_calibrate` calls `climada_damagefunctions_replace`, which does indeed not replace on repetitious calls if the result would be exactly the same.

See also `cr_country_hazard_test` in order to *test* country calibration.

cr_country_hazard_test

Given a `country_risk` results structure (see above, generated by `country_risk_calc`), experiment with one country and hazard to test different damage function settings etc.

This code is MOST LIKELY to be edited by the user, i.e. to set damagefunction parameters etc. It provides merely a TESTBED for efficient calibration of country results.

Process:

- Run `country_risk_calc` for either one or a set of countries. A set makes particularly sense for e.g. a peril region, such as TC atl, in order to ensure (neighboring) countries in that region have similar damagefunction settings. Hence you might e.g. run

  ```
  country_list={'Colombia','Costa Rica','Dominican Republic','United States'};
  peril_ID=['atl_TC';'atl_TS']; % TC and TS in Atlantic region
  country_risk=country_risk_calc(country_list,-7,0,0,peril_ID);
  ```

- call cr_DFC_plot to get a first overview by country, e.g.

  ```
  cr_DFC_plot(country_risk)
  ```

- call cr_DFC_plot_aggregate to get a first overview or the combined results of all countries, e.g. (note that the code does combine sub-peril EDSs himself)

```
cr_DFC_plot_aggregate(country_risk)
```

- Note that especially the comparison with EM-DAT makes only sense for either larger countries or a group of smaller ones - otherwise it might be too much due to chance whether a country got hit in the past years or not.
- Now, call `cr_country_hazard_test` to test different damagefunction settings (or modifications) for one country and peril, e.g. (for above `country_list` and perils, 2 points to Costa Rica[16], and 1 is the first peril in this country, i.e. TC[17] Atlantic):

```
country_risk=cr_country_hazard_test(country_risk,2,1)
```

Within `cr_country_hazard_test`, the damagefunction for TC is modified see section indicated with `*** edit the damage function below ***` in the code, e.g.

```
% ****** edit the damage function below ******************

[damagefunctions,dmf_info_str]=climada_damagefunction_generate(0:5:120,20,3,0.5,'exp','TC',0);
fprintf('%s TC atl: %s\n',country_name_char,dmf_info_str);
entity=climada_damagefunctions_replace(entity,damagefunctions);

% ****** end edit the damage function  *******************
```

- Within `cr_country_hazard_test` the modified damagefunction is written temporarily to `entity.damagefunctions`[18] and the specific event damage set (EDS) is re-calculated[19]. You now repeat the process (edit damage function parameters, call `cr_country_hazard_test`) until you're happy with the result.
- In case there is a sub-peril (in the present case TS[20]), you might want to repeat the process for the sub-peril (**2** for TS), i.e.

```
country_risk=cr_country_hazard_test(country_risk,2,2)
```

And repeat this call after each edit of the TS damagefunction again in the `*** edit the damage function below ***` section in `cr_country_hazard_test`.
- Note that `cr_country_hazard_test` shows the single-(sub-)peril results (e.g. for TC *or* TS). Therefore, in order to finally compare the combined result (TS *and* TS) with the combined EM-DAT (and, if present comparison model), call[21]

```
cr_DFC_plot(country_risk_EDS_combine(country_risk),2,1)
```

- You might also occasionally use this call while you experiment with e.g. the TC settings to check the combined result. Note that you can also provide damagefunction

[16] `country_list={'Colombia',`**`'Costa Rica'`**`,'Dominican Republic','United States'}`
[17] `peril_ID=[`**`'atl_TC'`**`;'atl_TS'];`
[18] Note that the modified entity is NOT saved, as `climada_damagefunctions_replace` does only return the entity with damagefunction(s) replaced. Only once you are happy with the adjustment, the code `country_risk_calibrate` does store the modified damagefunction(s) back to the entity file.
[19] In the specific case hence `country_risk(2).res.hazard(1).EDS` and the resulting DFC is plotted by `cr_country_hazard_test` calling itself `cr_DFC_plot(country_risk,2,1)`.
[20] i.e. `peril_ID=['atl_TC';`**`'atl_TS'`**`]`
[21] Note that the last parameter ‚1' now refers tot he combined EDS, since `country_risk_EDS_combine` did add the TS damage tot he TC damage, stored into the TC EDS.

modifications for both TC and TS in `cr_country_hazard_test`, e.g.

```
% ****** edit the damage function below ******************

[damagefunctions,dmf_info_str]=climada_damagefunction_generate(0:5:120,20,3,0.5,'exp','TC',0);
fprintf('%s TC atl: %s\n',country_name_char,dmf_info_str);
entity=climada_damagefunctions_replace(entity,damagefunctions);

[damagefunctions,dmf_info_str]=climada_damagefunction_generate(0:16,0.5,2,0.3,'s-shape','TS',0);
fprintf('%s TS atl: %s\n',country_name_char,dmf_info_str);
entity=climada_damagefunctions_replace(entity,damagefunctions);

% ****** end edit the damage function  *******************
```

- And then call it twice[22] (to re-calculate the TC and TS EDSs) and then check the aggregate result with `cr_DFC_plot(country_risk_EDS_combine(country_risk),2,1)`.
- And in case you'd like to see the full basin-wide aggregated result, call

```
cr_DFC_plot_aggregate(country_risk)
```

- This is NOT the same as `cr_DFC_plot(country_risk_EDS_combine(country_risk),2,1)`, as above call aggregates over all countries, while `cr_DFC_plot(…,2,1)` only shows the results for country 2 (here Costa Rica).
- Finally, once you're happy with the combined result, consider adding (or updating) the particular section in `country_risk_calibrate`, e.g. for the above example, this could be[23]:

```
…
switch country_name_char
  case 'Costa Rica'
  [damagefunctions,dmf_info_str]=climada_damagefunction_generate(0:5:120,20,3,0.5,'exp','TC',0);
  fprintf('%s TC atl: %s\n',country_name_char,dmf_info_str);
  entity=climada_damagefunctions_replace(entity,damagefunctions);
  if ~isempty(entity_future)
    entity_future=climada_damagefunctions_replace(entity_future,damagefunctions);end

  [damagefunctions,dmf_info_str]=climada_damagefunction_generate(0:16,0.5,2,0.3,'s-shape','TS',0);
  fprintf('%s TS atl: %s\n',country_name_char,dmf_info_str);
  entity=climada_damagefunctions_replace(entity,damagefunctions);
  if ~isempty(entity_future)
    entity_future=climada_damagefunctions_replace(entity_future,damagefunctions);end
…
```

- Note that one can really copy-paste the specific code segment (here highlighted in green) from `cr_country_hazard_test` and only needs to add the future entity treatment (here in blue).

In order to avoid troubles with GitHub, please make your copy of the function `cr_country_hazard_test`, name it e.g. `cr_country_hazard_mytest`, and experiment with different damage function settings for a given country and region (group of countries). In special cases, you might also consider adjusting hazard event sets.

Next step: put your final adjustments in country_risk_calibrate (and if you're really of the opinion they are an improvement of climada, please check these changes in).

---

[22] i.e. `country_risk=cr_country_hazard_test(country_risk,2,1); country_risk=cr_country_hazard_test(country_risk,2,2)`

[23] Standard procedure is that the switch statement below has entries for countries (and lists of countries) and hence performs the specific actions. Be careful to check for repetitious application. We set the field `entity.calibrated=1` the first time it is treated here, but since one might need to re-calibrate, one should rather assign absolute values (e.g. `damagefunctions.MDD=((1:length(damagefunctions.Intensity))^2)/... (length(damagefunctions.Intensity)^2)`, and avoid statements such as `damagefunctions.MDD=... damagefunctions.MDD*2`, since a mere multiplication of existing values might lead to troubles on subsequent calls. The code `climada_damagefunctions_replace` does indeed not replace on repetitious calls if the result would be exactly the same.

Function reference

Use `help {function name}` to get a detailed description and input/output specification

Top level functions

`country_risk_calc`: generate assets and hazard sets for a given country (i.e. admin0, or a list of countries). This function provides a convenient way to get started for any given country.
`country_admin1_risk_calc`: Once `country_risk_calc` has been run for a country (or a series of), calculate the admin1 (state/province) level results.
`country_risk_EDS_combine`: combine sub-peril EDSs in `country_risk` structure, such as TC and TS – helpful to reduce the complexity of results.
`country_risk_EDS2YDS`: convert event event damage sets to year damage sets
`country_risk_report`: produce a report given output from `country_risk_calc`, `country_admin1_risk_calc` or `cr_economic_loss_calc` (see also `cr_loss_multiplier_plot`).
`country_risk_calibrate`: for many countries, climada has been calibrated (to some extent) and this code does modify the damagefunctions of a given country accordingly. Just call this code with the full list of countries you've processed once with `country_risk_calc` – and climada does automatically apply all calibrations to all countries (and all hazards). This overwrites damagefunctions in the respective entities (`entity.damagefunctions...`), but keeps track of changes (and does not re-apply on subsequent calls, hence one can call `country_risk_calibrate` repeatedly).

`climada_nightlight_entity`: Generate the (high-resolution) asset distribution for any country (admin0) and any state/province (admin1) within.

Plotting functions

cr_DFC_plot: plot damage frequency curves (DFC) for all countries and perils in country_risk
cr_DFC_plot_aggregate: plot combined peril damage frequency curves (DFC) for all countries and also plot the peril region aggregate as well as the global aggregate DFC.

Support-level functions

`cr_economic_loss_calc`: given (property damage) output from `country_risk_calc`, calculate total economic loss
`cr_EDS_emdat_adjust`: given an event damage set, adjust such that it matches best the EM-DAT damage history of a given country and hazard (see also `emdat_read`)
`emdat_read`: read the EM-DAT global damage database (www.emdat.be, by country and peril). See also Appendix.
`cr_country_hazard_test`: given a country_risk results structure, experiment with one country and hazard to test different damage function settings etc. Make your own copy before using this code, as you will need to edit it.

Also useful

`climada_entity_value_GDP_adjust`: given an entity with assets for a country, adjust total asset value to represent country assets (a simple formula based on GDP and country development index).
`climada_damagefunctions_plot`: plot damage functions
`climada_create_GDP_entity`: create an entity (asset distribution) based on night light intensity (climada module GDP_entity).

Useful data sources

Here, we list a couple data sources useful for purposes as provided by this module.

- *http://web.ornl.gov/sci/landscan/: global population distribution, 1 km resolution (30" x 30")*[24]*.*
- http://due.esrin.esa.int/page_globcover.php: Global Land Cover Map, 300 m resolution, 2009.

[24] Only listed for reference, since not open-access. License fee for humanitarian organizations, educational research and commercial organizations determined on a case-by-case basis.

Appendix

A. Calculation of economic damage in `cr_economic_loss_calc`

Starting point for the economic loss calculation is damage(event_i), i.e. the property damage calculated by climada_EDS_calc. The economic loss then also includes secondary losses to an economy including e.g. lost ouput, retail sales, wages, costs to business form rerouting goods and services around the affected area, reduced taxable receipts, etc.
The underlying rationale of the calculation is that a property damage resulting from a natural disaster does not have a major impact on a country's economy as long as the damage is small compared to the country's GDP, and as long as adequate financing and national resources exist. However, if a damage is big, it will be exacerbated depending on how well a country is "in shape" to deal with major shocks.

`cr_economic_loss_calc` intends to estimate the economic damage as it manifests itself about 3-6 months after a disaster occurred. This first-round effect of natural disasters is usually that income and output (GDP) fall. What happens in the next round then depends on the way the country or region responds to the crisis. For example, Japan is a strong economy and has the resources to start rebuilding quickly, while smaller and badly managed countries such as Haiti can suffer severe long-term effects. It should be noted that `cr_economic_loss_calc` only calculates the temporary dip following the first months after a disaster (see Figure).
After that initial dip, different scenarios can lead to no, positive or negative follow-on effects over the long run. In particular, a natural disaster can even positively affect total factor productivity, e.g. when it leads to the replacement of damaged, outdated production facilities and physical infrastructure with state-of-the-art facilities and infrastructure. However, as noted above, `cr_economic_loss_calc` only deals with the calculation of the initial impact, not with the long-term scenarios.

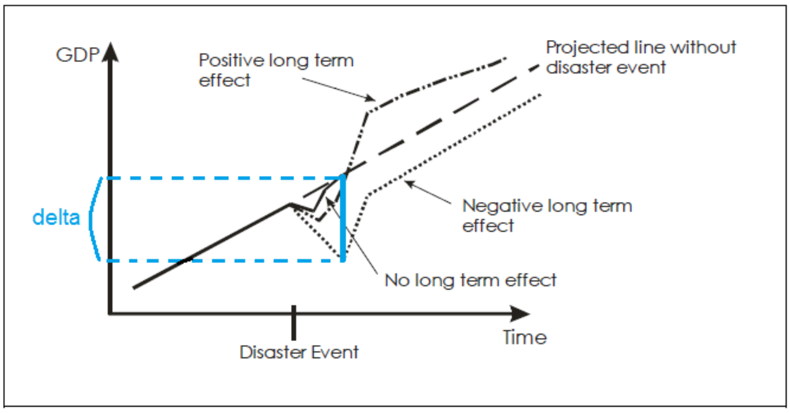

Figure: Possible trajectories of GDP after a disaster[25]. The delta is the initial temporary dip `cr_economic_loss_calc` provides an estimate for.

[25] Source: Hochrainer, 2006
https://openknowledge.worldbank.org/bitstream/handle/10986/4162/WPS4968.pdf?sequence=1

In `cr_economic_loss_calc`, the economic loss caused by the natural disaster is calculated according to:

```
economic_loss(event_i) = damage(event_i) * loss_multiplier
```

where loss_multiplier is defined by:

```
loss_multiplier =
1 + cr_get_damage_weight(damage(event_i)/GDP) * country_damage_factor
```

with:

`cr_get_damage_weight`: function that determines how much weight a damage should be given based on its ratio to GDP

```
country_damage_factor = 1/financial_strength ...
                      + BI_and_supply_chain_risk ...
                      + natural_hazard_economic_exposure ...
                      - disaster_resilience
```

Hence, `country_damage_factor` consists of four terms:

- `financial_strength` measures a country's economic health and ability to finance the recovery.

  ```
  financial_strength = total_reserves/GDP ...
                     + insurance_penetration ...
                     + income_group ...
                     - central_government_debt
  ```

- `BI_and_supply_chain_risk` measures a country's risk of disaster-related business and supply chain interruption

  ```
  BI_and_supply_chain_risk = GDP_industry ...
                           + FM_resilience_index_supply_chain/100
  ```

- `natural_hazard_economic_exposure` assesses which countries have a concentration of their total economic output exposed to natural hazards

  ```
  natural_hazard_economic_exposure = ...
                     1 - Natural_Hazards_Economic_Exposure/10
  ```

- `disaster_resilience` measures the quality of a country's natural hazard risk management, i.e., the country's "preparedness" to deal with the consequences of a disaster

  ```
  disaster_resilience = FM_resilience_index_risk_quality/100 ...
                      + (global_competitiveness_index-1)/6
  ```

See economic_indicators_mastertable.xls (in the data folder of the country_risk module[26]) for more information on the four components of country_damage_factor (and their respective subcomponents).

[26] Download: https://github.com/davidnbresch/climada_module_country_risk

B. The database as read by `emdat_read`

The data as used in the calibration described above comes from EM-DAT (www.emdat.be). It has been retrieved as described in the readme tab in the Excel file containing the data as downloaded[27]. Please note that TWO database queries are needed, since one cannot obtain *Disaster type* and *Disaster subtype* at the same time. The records are then joined together in Excel, in order for climada to read the 'complete'[28] database.

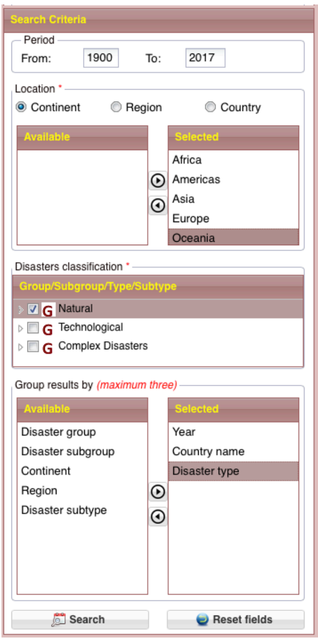

Fig: The screen as used to download the EM-DAT damage data from www.emdat.be

[27] See file ../data/emdat/emdat.xlsx and the tab _readme therein.
[28] There are some remaining issues, as disaster type and subtype are not fully consistently handled in EM-DAT. Hence you might want to inspect the Excel file for special cases.