

icclim a python library to compute climate indices

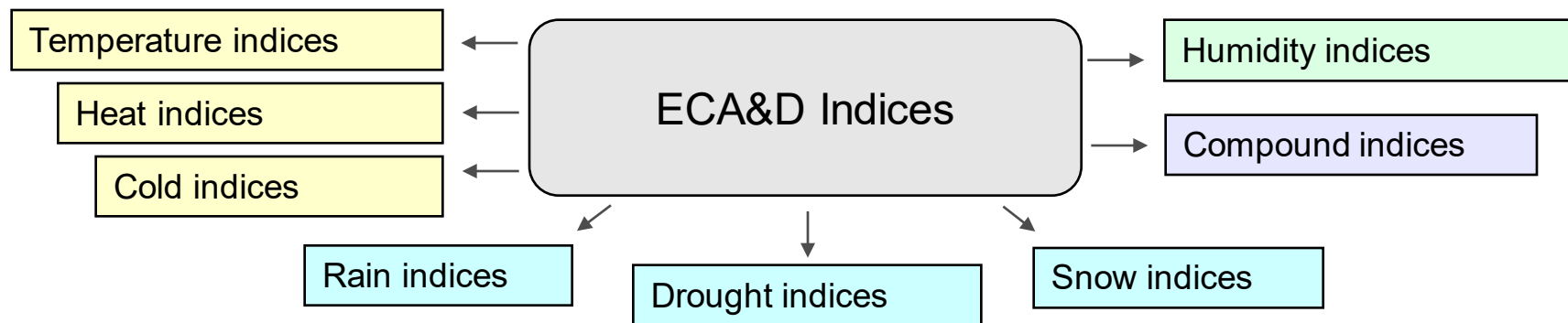


Christian Pagé
CERFACS, Toulouse,
France

Abel Aoun



icclim: Index Calculation for Climate



- Intra-period extreme temperature range [°C] - **ETR**
- Warm days (days with mean temperature > 90th percentile of daily mean temperature) - **TG90p**
- Summer days (days with max temperature > 25 °C) - **SU**
- ...

- Python code developed at Cerfacs since September 2013
- Funded by EU FP7 IS-ENES2, FP7 CLIPC and H2020 IS-ENES3
- Generic and modular approach, can be reused in other environments
- New V5 completely rewritten and using underlying xclim functions, based on xarray and dask
- I/O interface is structured for optimal performance
- Implement the proper percentile indices calculations when calculation period overlaps reference period (called bootstrapping method)
- Available indices: https://icclim.readthedocs.io/en/latest/explanation/climate_indices.html#icclim-capabilities



<https://github.com/cerfacs-globc/icclim>

ci-build passing pypi v6.1.5 code style black docs passing conda-forge v6.1.5 Coverage 91% DOI 10.5281/zenodo.7382653

Fork 30 Starred 62 1,345 commits

```
import icclim
summer_days = icclim.su("netcdf_files/tasmax_1990-2100.nc", out_file="summer_days.nc")
```

icclim: examples

notebooks: <https://gitlab.com/is-enes-cdi-c4i/notebooks/>

```
## KNMI TX
tx_files = glob.glob(f"netcdf_files/knmi/clean/*tx*.nc")
bp = [datetime.datetime(1901, 1, 1), datetime.datetime(1921, 12, 31)]
icclim.index(index_name='tx90p',
              in_files=tx_files,
              base_period_time_range=bp,
              slice_mode='YS',
              out_file="netcdf_files/output/out.nc")
```

[new in 5.1.0] Compute multiple indices at once

Documentation https://icclim.readthedocs.io/en/latest/how_to/recipes_ecad.html#multi-index-computation

```
bp = [datetime.datetime(1991, 1, 1), datetime.datetime(1999, 12, 31)]
tr = [datetime.datetime(1991, 1, 1), datetime.datetime(2010, 12, 31)]
res = icclim.indices(
    index_group=IndexGroup.HEAT,
    in_files="./netcdf_files/climpact.sampledata.gridded.1991-2010.nc",
    base_period_time_range=bp,
    time_range=tr,
    out_file="pouetpouet.nc",
)
```

[new in 5.2.0] one function per index

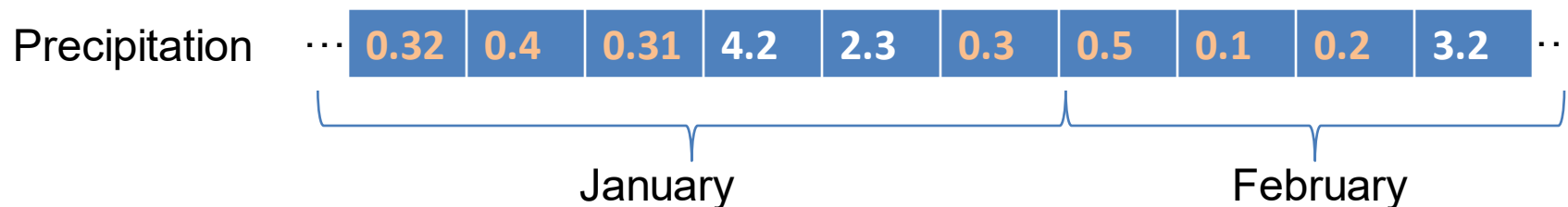
Documentation: https://icclim.readthedocs.io/en/latest/references/ecad_functions_api.html

```
import glob
import icclim
summer_days = icclim.su(
    in_files=glob.glob("netcdf_files/tasmax*.nc"),
    out_file="summer_days.nc",
)
```

[new in 5.3.0] custom season between exact dates

```
result = icclim.su(  
    in_files=xr.open_dataset(climp_file).tmax,  
    slice_mode=["season", ["19-07", "8 Aout"]],  
) .compute()
```

[new in 6.0] Spells starting before the season bounds are properly counted



Say we want to compute cdd on a monthly basis...

CDD: number of consecutive where $pr < 1$ mm/day

```
cdd = icclim.cdd(in_files=precipitation, slice_mode="month").CDD.compute()
```

Should February cdd be 3 days or 4 ?

With icclim 6 we count spells length before resampling into monthly values,
So CDD in February would be 4 here

[new in 6.0] Generic indices – What ?

What's the difference between SU, FD and Tx90p ?

Number of days where $t_x > 25 \text{ degC}$

Number of days where $t_n < 0 \text{ degC}$

Number of days where $t_x > 90\text{th (daily) percentile}$

An operator

A studied variable

A threshold

For our output, we want the right computation to be executed and to have specific metadata to clearly identify :

- What was the computation configuration
- What was the initial data used

[new in 6.0] Generic indices – How ?

```
icclim.index(in_files=climp_file,  
             Operatorindex_name="count_occurrences",  
Studied variable var_name="tmax",  
             Threshold threshold="≥ 20 deg_C",  
             ).compute()
```

Not CF proof!

standard_name: number_of_days_with_maximum_air_temperature_above_threshold

long_name: Number of days when maximum air temperature is greater or equal to 293.15 K for each year.

units: d

cell_methods: time: sum over days



```
thresh = Threshold("≥ 75 period_per",  
                    threshold_min_value="1 mm/day",  
                    reference_period=['1991-01-01', "31 décembre 2000" ])
```

R75p (ECAD)

```
r75p = icclim.index(climp_file,  
pr_thresh = Threshold("≥ 75 doy_per", threshold_min_value="1 mm/day")  
temp_thresh = Threshold("≥ 75 doy_per")  
c_w = icclim.index(climp_file,
```

CW (ECAD)

```
pr_thresh = Threshold("≥ 75 doy_per", threshold_min_value="1 mm/day")  
temp_thresh = Threshold("≥ 75 doy_per")  
wind_thresh = Threshold("≥ 95 doy_per")  
c_w_w = icclim.index(ds,
```

Cold, wet and windy
(Bretagne?)

```
    index_name="count_occurrences",  
    var_name=["precip", "tmax", "sfcWind"],  
    threshold=[pr_thresh, temp_thresh, wind_thresh])
```

[new in 6.0] Generic indices – Operators

count_occurrences	SU, TR, TG90p, CD, R95p SD50cm ...
max_consecutive_occurrence	CSU, CFD, CDD, CWD
sum_of_spell_lengths	WSDI, CSDI
excess	GD4
deficit	HD17
fraction_of_total	R75pTOT, R95pTOT, r99pTOT
maximum	RX1day, TXx, TNx, custom__maximum
minimum	TXn, TNn, user_index - minimum
average	TG, TN, TX, SDII, SD, custom__mean
sum	PRCPTOT, custom__sum
standard_deviation	**new**
max_of_rolling_sum	RX5day, user_indexrolling_sum
min_of_rolling_sum	custom__rolling_sum
min_of_rolling_average	custom__rolling_mean
max_of_rolling_average	custom__rolling_mean
mean_of_difference	DTR
difference_of_extremes	ETR
mean_of_absolute_one_time_step_difference	vDTR
difference_of_means	custom_anomaly

[new in 6.0] Generic indices – Thresholds

Scalar	Threshold(query=">=", value=25, unit="degC")
Day of year percentiles	Threshold(query=">=", value=99, unit="doy_per")
Period percentiles	Threshold(query=">=", value=99, unit="period_per")
Bounded period percentiles.	Threshold(">= 75 period_per". threshold_min_value="1 mm/day", reference_period=['1991-01-01', "31 décembre 2000"])
Per-grid cell threshold	Threshold(query=">=", value="data.nc", threshold_var_name="tmin", unit="K")
Sequence of scalars	Threshold(query=">=", value=[25, 29], unit="degC")

A F.A.I.R icclim

- Findable:
 - [Open source code](#), licensed under the permissive Apache 2 license.
 - Try to follow [OpenSSF principles](#)
 - Documentation freely available on [readthedocs](#)
 - Dissemination by Christian in conferences (AGU, AMS, EGU...).
 - Integrated in [C4I](#) and C3S.
- Accessible
 - Documentation on readthedocs.
 - Github issues and pull requests.
 - CERFACS support the development.
 - Can be installed with **pip** or **conda-forge**.
- Interoperable
 - Based on well knowns libraries: xarray, xclim, numpy.
 - Support NetCDF and xarray.Dataset input formats.
 - Indices are deriviated from ECA&D and CF conventions standards.
 - icclim custom parsing for non CF complying inputs.
- Reusable
 - NetCDF output, with enriched metadata.
 - Easy to integrate within existing infrastructures.
 - Easy to write small scripts with it.

What could be next ?

- Improve bootstrap performances
- Improve documentation
- Add Provenance metadata
- Create a D.S.L to handle requests such as `count days where tx > 25 °C`

icclim users



Climate4Impact



Researchers

Created by Tippawan Soo

Thanks !



Christian Pagé
Abel Aoun

christian.page@cerfacs.fr

THE CONSORTIUM

Coordinated by CNRS-IPSL, the IS-ENES3 project
gathers 22 partners in 11 countries



*This project has received funding from the European Union's
Horizon 2020 research and innovation programme under grant
agreement N°824084*



Our website
<https://is.enes.org/>



Follow us on Twitter !
@ISENES_RI



Contact us at
is-enes@ipsl.fr



Follow our channel
IS-ENES3 H2020