



HARMONIZE

clim4health

HARMONIZE toolkit course, November 2025

Emily Ball, Alba Llabrés-Brustenga, Carles Milà, Raúl Capellán, Daniela Lührsen,
Ania Kawiecki, Rachel Lowe



This work was supported by the Wellcome Trust grant number 224694/Z/21/Z.

Session Outline



9.30	Presentation
10.00	Hands on tutorial - workflow 1: downscaling and skill assessment
11.00	Coffee(?) break
11.30	Hands on tutorial - workflow 2 and 3: spatio-temporal aggregation, calibration and masking
12.10	Feedback
12.30	End of session



Emily Ball



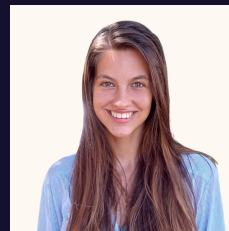
Alba Llabrés



Carles Milà



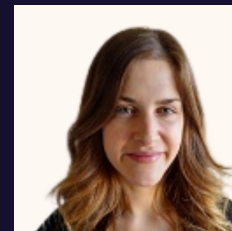
Raúl Capellán



Daniela Lühsen



Ania Kawiecki



Rachel Lowe

Aims of the tutorial

1. Understand sources of climate data
2. Learn how to load climate data into R using clim4health
3. Learn methods to postprocess climate data and apply them using clim4health
4. Learn how to assess forecast quality
5. Learn how to plot climate data using clim4health

Aims of the tutorial

1. Understand sources of climate data
2. Learn how to load climate data into R using clim4health
3. Learn methods to postprocess climate data and apply them using clim4health
4. Learn how to assess forecast quality
5. Learn how to plot climate data using clim4health



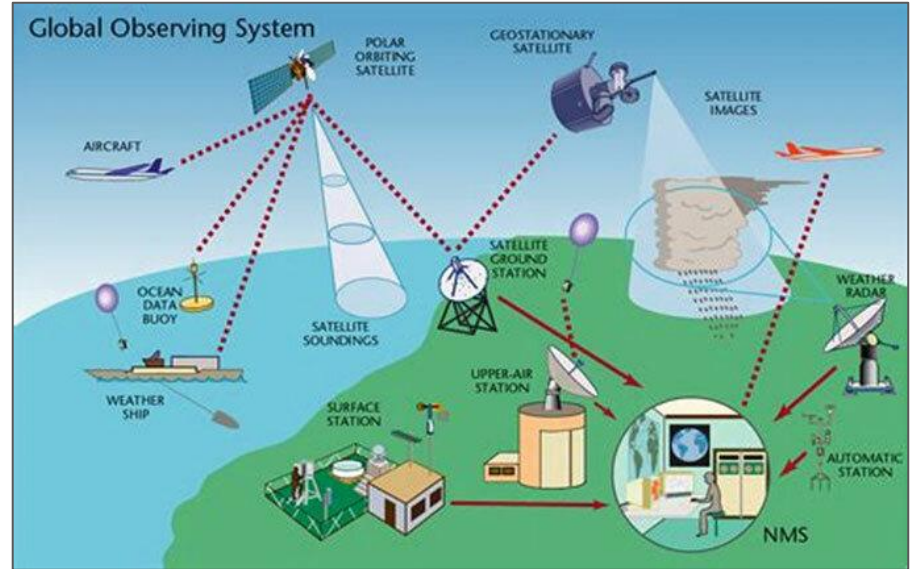
Aims of the tutorial

1. Understand sources of climate data
2. Learn how to load climate data into R using clim4health
3. Learn methods to postprocess climate data and apply them using clim4health
4. Learn how to assess forecast quality
5. Learn how to plot climate data using clim4health

Direct observations come from many sources:

- Satellites
- Radar
- Meteorological stations
- Aircraft and balloons
- Buoys
- ... more!

Observations

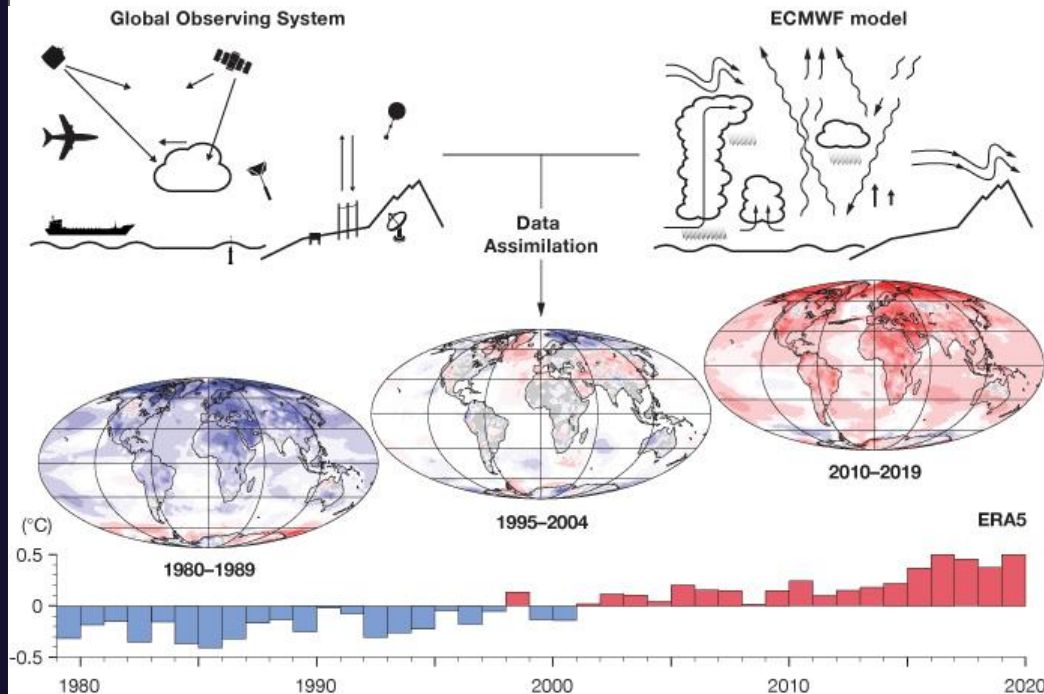


Reanalysis is a type of “observation”:

- We run a climate model and “nudge” (adjust) it towards historical observations
- physically consistent
- temporally/spatially gridded
- can provide estimates of variables that were not directly observed

Note: in regions where direct observations are limited, reanalyses are more model-driven.

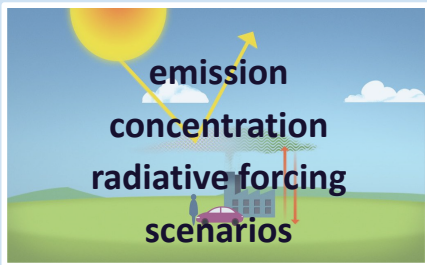
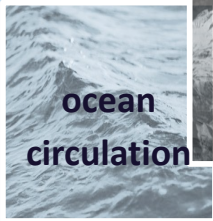
Reanalysis



Predictions and Projections

These are model-based predictions of *future* climate.

- They are available at different timescales.
- They contain ***ensemble members*** that help us to capture some of the uncertainty in the predictions.
- Predictability comes from different sources...



Sources of predictability

now
hours
days

weeks

months

seasons

years

decades

centuries

time

1 - 15 days
Weather forecasts

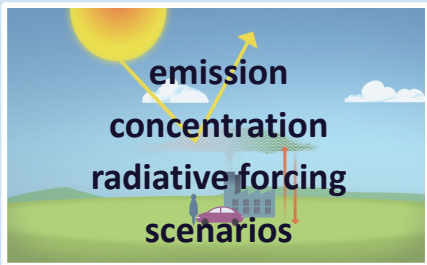
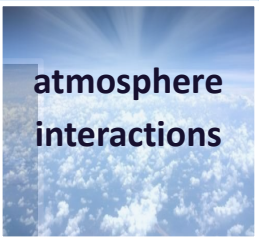
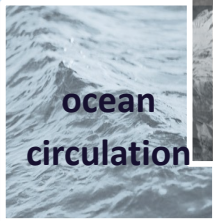
10 - 32 days
Sub-seasonal
1 - 15 months
Seasonal

1 - 10 years
Decadal

Climate predictions

20 - 100 years
Climate change projections

Time Scales



Sources of predictability

now
hours
days

weeks

months

seasons

years

decades

centuries

time

1 - 15 days
Weather forecasts

10 - 32 days
Sub-seasonal
1 - 15 months
Seasonal
1 - 10 years
Decadal
Climate predictions

20 - 100 years
Climate change projections

Time Scales

Seasonal forecasts available from Copernicus:

- Spatial resolution: $1^\circ \times 1^\circ$ (~100km)
- Temporal frequency: 6h, 12h, 24h and monthly statistics
- Hindcast period: 1993-2016
- System: ECMWF-s5

Reanalysis available from Copernicus:

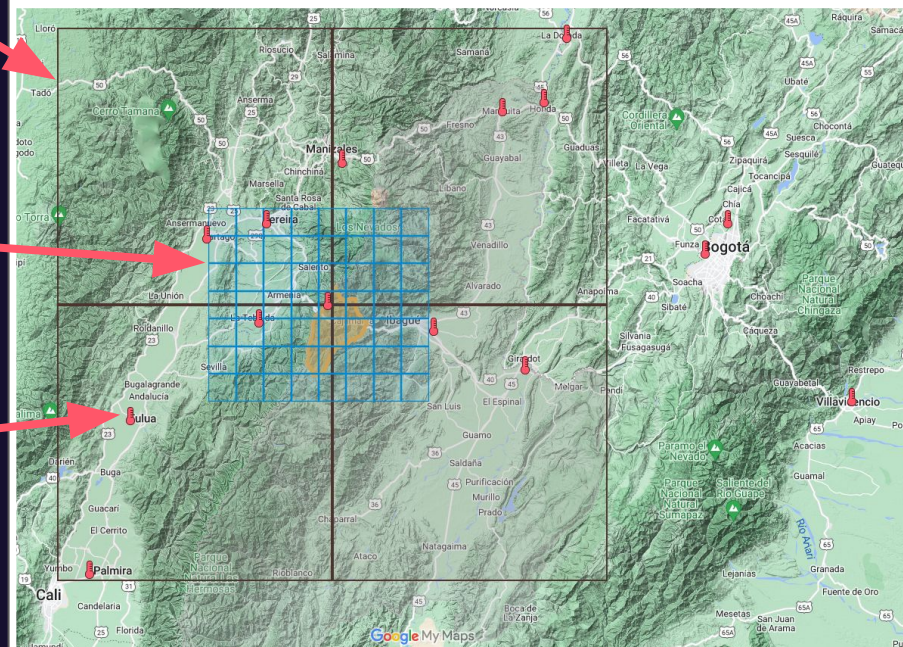
- ERA5 1979-present $0.25^\circ \times 0.25^\circ$
- ERA5-Land 1950-present $0.1^\circ \times 0.1^\circ$ (~10km)

Ground-based observations:

- Global Hourly - Integrated Surface Database (ISD): point data, several variables, 1901-present

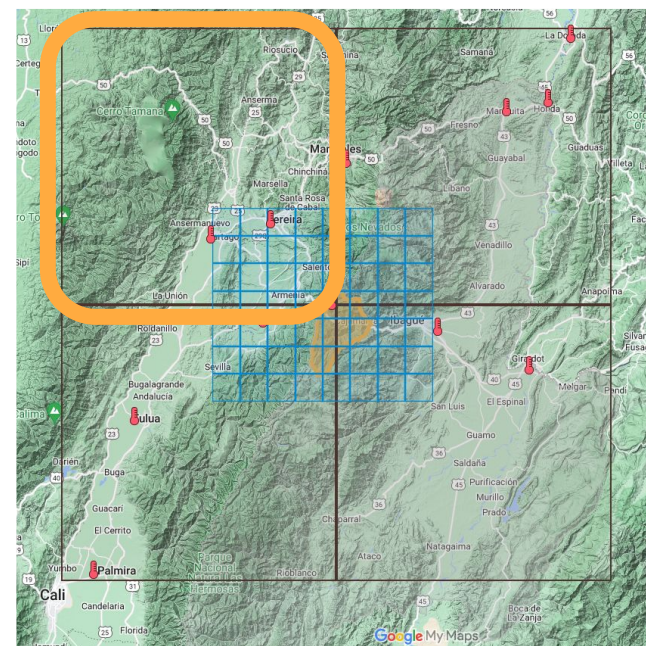
Health studies often require data at a fine spatial scale

Example sources of climate data



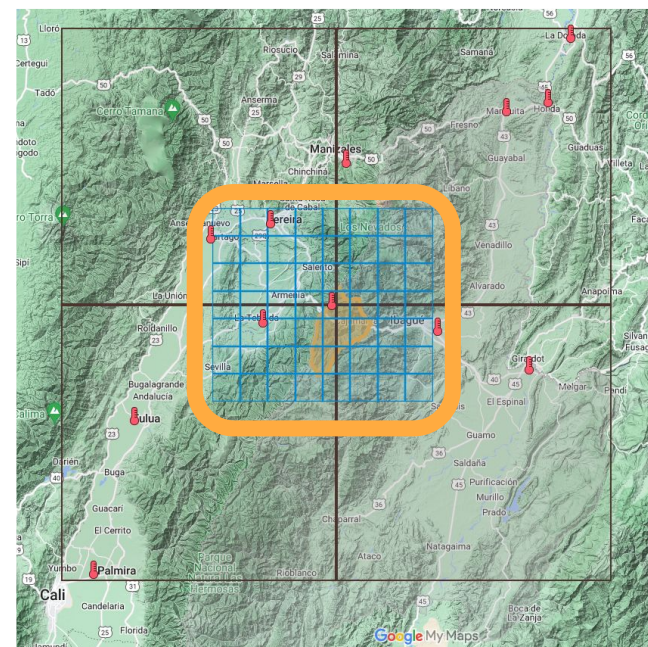
Example sources of climate data

	Seasonal forecast	Reanalysis		Ground-based observations
		ERA5	ERA5Land	
Centre/System	ECMWF-SEAS5.1	Copernicus	Copernicus	Global Integrated Surface Database (ISD)
Spatial resolution	1° x 1° (~100km)	0.25° x 0.25°	0.1° x 0.1° (~10km)	Point data
Temporal frequency	6h, 12h, 24h, monthly statistics	Hourly, daily or monthly statistics	Hourly, daily or monthly statistics	Hourly
Temporal range	1981 - present (hindcast period = 1994-2016)	1979 - present	1950 - present	1901 - present



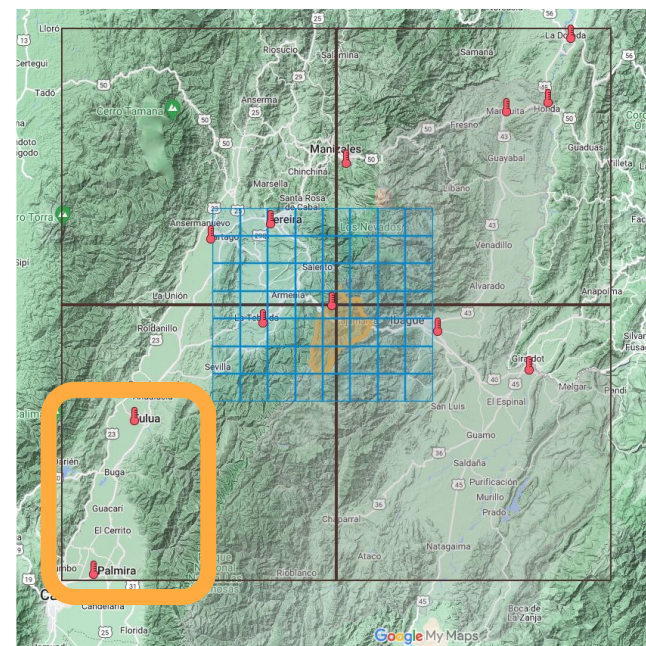
Example sources of climate data

	Seasonal forecast	Reanalysis		Ground-based observations
		ERA5	ERA5Land	
Centre/System	ECMWF-SEAS5.1	Copernicus	Copernicus	Global Integrated Surface Database (ISD)
Spatial resolution	1° x 1° (~100km)	0.25° x 0.25°	0.1° x 0.1° (~10km)	Point data
Temporal frequency	6h, 12h, 24h, monthly statistics	Hourly, daily or monthly statistics	Hourly, daily or monthly statistics	Hourly
Temporal range	1981 - present (hindcast period = 1994-2016)	1979 - present	1950 - present	1901 - present



Example sources of climate data

	Seasonal forecast	Reanalysis		Ground-based observations
		ERA5	ERA5Land	
Centre/System	ECMWF-SEAS5.1	Copernicus	Copernicus	Global Integrated Surface Database (ISD)
Spatial resolution	1° x 1° (~100km)	0.25° x 0.25°	0.1° x 0.1° (~10km)	Point data
Temporal frequency	6h, 12h, 24h, monthly statistics	Hourly, daily or monthly statistics	Hourly, daily or monthly statistics	Hourly
Temporal range	1981 - present (hindcast period = 1994-2016)	1979 - present	1950 - present	1901 - present



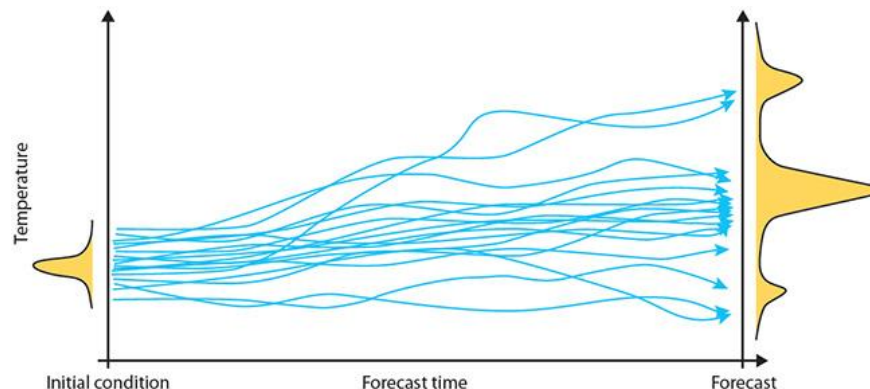
Hindcasts (past forecasts) are forecasts initialised in the past.

- Used to compare how well the forecast model predicts actual observed values.
- Allow us to quantify model **skill** - we do this by comparing the hindcast to the observations.

Ensemble members capture the envelope of uncertainty.

- Climate models are run with multiple slightly different initial conditions.

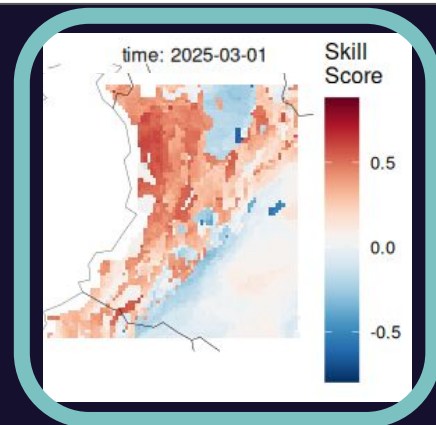
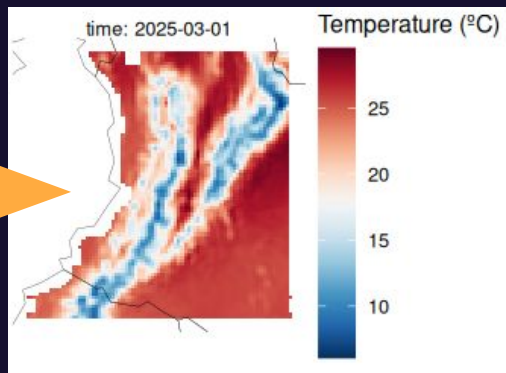
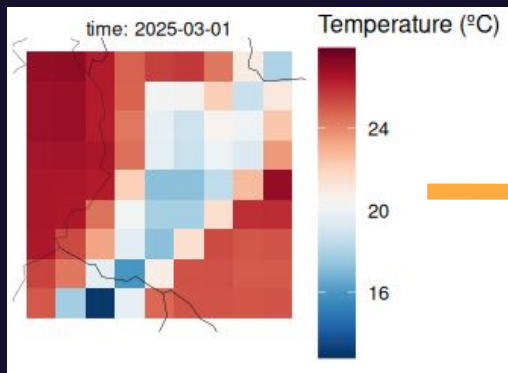
Hindcasts and ensembles



Downscaling and verification

We can leverage observations and hindcasts to **downscale** a forecast to **finer spatial resolution**

We can identify statistical relationships between observations and hindcasts to **assess the quality** of a forecast model



clim4health package structure



obtain input data

transform

prepare outputs

c4h_get

optional use

download climate data (reanalysis, observations and seasonal forecasts) from the Copernicus Climate Data Store

c4h_load

load data into R object and start to work with the data

c4h_space

spatial aggregation

apply one or several in the desired order

c4h_time

temporal aggregation (from daily to weekly to monthly to annual)

c4h_downscale

calibration/downscaling

c4h_postprocess

verification and quality assessment

c4h_index

calculation of simple indices (threshold based)

c4h_convert

transform s2dv to common formats

c4h_plot

visualise and save png

c4h_save

save as csv or grid

UTILS

additional helper functions

Aims of the tutorial

1. Understand sources of climate data
2. Learn how to load climate data into R using clim4health
3. Learn methods to postprocess climate data and apply them using clim4health
4. Learn how to assess forecast quality
5. Learn how to plot climate data using clim4health



Loading data

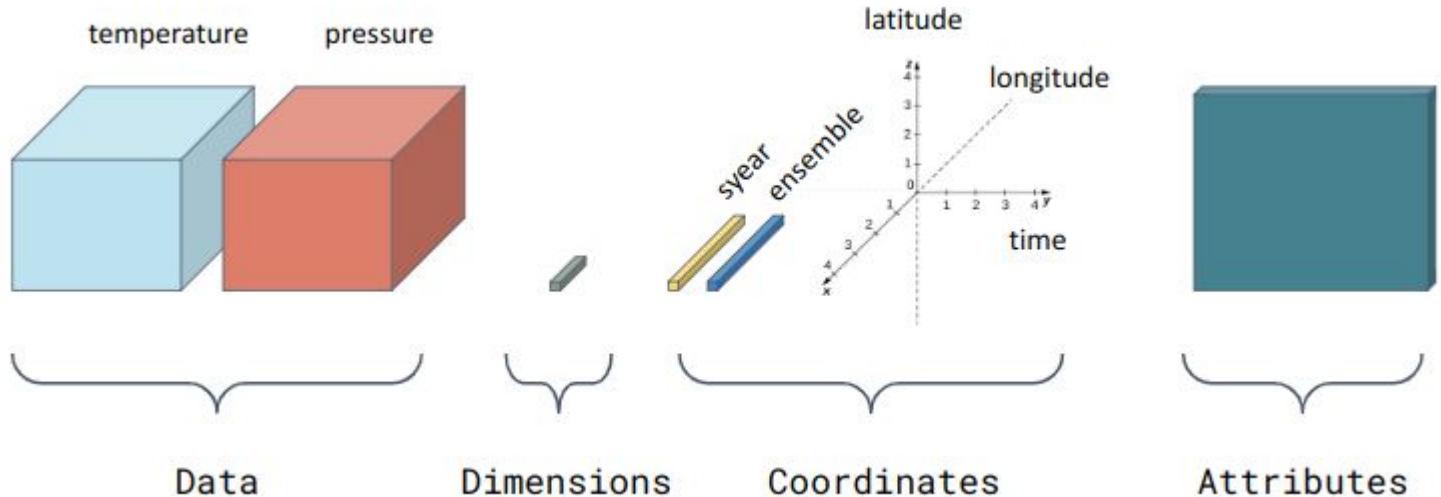
c4h_load: load climate data from netCDF or csv file format into an s2dv_cube object

```
{r}|  
#| include: false  
fcst <- clim4health::c4h_load(path = fcst_path,  
                             variable = "t2m",      # variable to load  
                             year = 2025,           # forecast initialisation year  
                             month = 1,             # forecast initialisation month  
                             leadtime_month = 1:3,  # load leadtimes 1-3  
                             ext = "nc")            # file extension
```

- Key point: specify the dates and times using arguments “year”, “month”, “day”, and “time”
- Specify forecast leadtimes with “leadtime_month”
- We can use further parameters such as “bbox” to specify the region we want to load
- Output: an **s2dv_cube**

The s2dv_cube object

- Climate data is highly dimensional - often includes *latitude*, *longitude*, *time*, as well as potentially others such as *ensemble*, *height*
- It's a list containing the climate variables, their dimensions, the coordinates, and any additional information



Exploring the data

- **dataset** — experiments, usually = 1
- **var** — loaded variables
- **sdate** — (forecast) initialisation time
- **time** — (forecast) lead time
- **ensemble** — model ensemble member
- **spatial dimension(s)**
 - latitude + longitude (gridded data)
 - location (point data)
 - area (polygon data)

**clim4health always works with
climate data in these dimensions**

Exploring the data

Useful commands

`str(fcst)` — prints extended information about the elements of the `s2dv_cube`

`dim(fcst$data)` or `fcst$dims` — prints the dimensions of the data

```
print("print fcst class")
class(fcst)
print("print dimensions of the stored data")
dim(fcst$data)
print("print the names of the list elements in fcst")
names(fcst)
print("print a summary of the data stored in fcst")
summary(fcst$data)
print("print extended information about the list elements in fcst")
str(fcst)
```

```
[1] "print fcst class"
[1] "s2dv_cube"
[1] "print dimensions of the stored data"
  dataset      var  sdate      time  ensemble  latitude longitude
      1         1      1         3         51         5         5
[1] "print the names of the list elements in fcst"
[1] "data"  "dims"  "coords" "attrs"
```

How to specify time in c4h_load()

- year = 1994:2016
- month = 1
- day = 1
- leadtime_month = 1:3



chooses all the possible start (initialisation) dates. These will be the first elements in the time dimension

chooses how many months should be loaded in the time dimension

sdate = forecast
start **date**

sdate

time



time 			
	1994-01-01	1994-02-01	1994-03-01
	1995-01-01	1995-02-01	1995-03-01
	1996-01-01	1996-02-01	1996-03-01

	2016-01-01	2016-02-01	2016-03-01

- year = 1994:2016
- month = 1:2
- day = 1
- leadtime_month = 1:3

chooses all the possible start (initialisation) dates. These will be the first elements in the time dimension

chooses how many months should be loaded in the time dimension

time 			
sdate 	1994-01-01	1994-02-01	1994-03-01
	1994-02-01	1994-03-01	1994-04-01
	1995-01-01	1995-02-01	1995-03-01
	1995-02-01	1995-03-01	1995-04-01

	2016-01-01	2016-02-01	2016-03-01
	2016-02-01	2016-03-01	2016-04-01

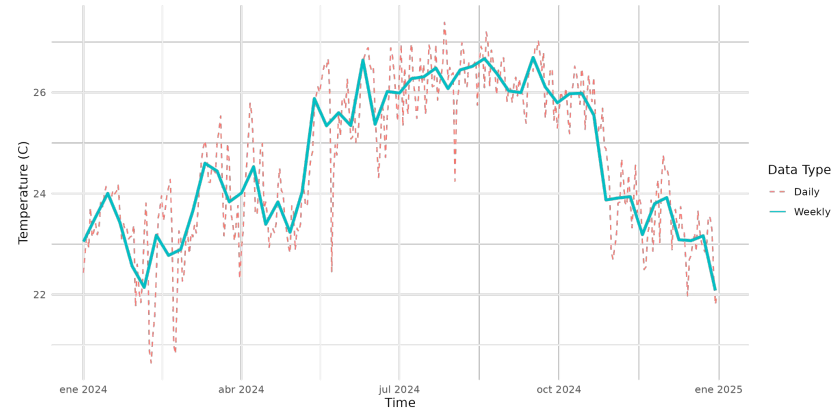
Aims of the tutorial

1. Understand sources of climate data
2. Learn how to load climate data into R using clim4health
3. Learn methods to postprocess climate data and apply them using clim4health
4. Learn how to assess forecast quality
5. Learn how to plot climate data using clim4health



Temporal aggregation

c4h_time:
aggregate climate
data to coarser
temporal resolutions
(e.g. hourly to daily,
daily to weekly or
monthly)



```
{r}  
obs_daily_mean <- clim4health::c4h_time(data = obs,  
                                         time_aggregation = "daily",  
                                         fun = "mean",  
                                         dim_aggregation = "time")
```

```
dim(obs_daily_mean$data)
```

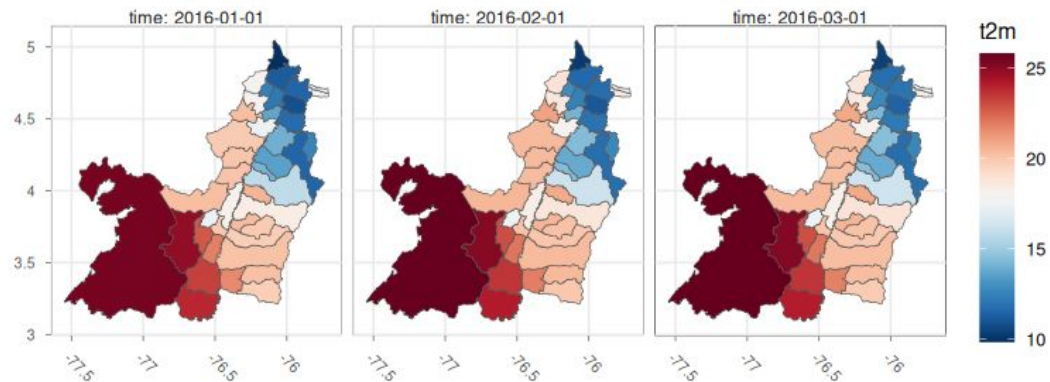
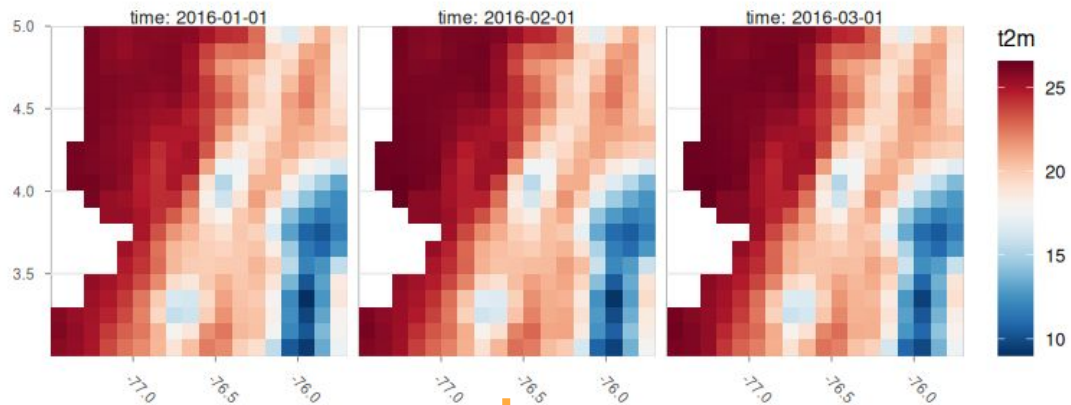
```
print(obs_daily_mean$attrs$Dates[1:10])
```

```
dataset    var    sdate    time ensemble latitude longitude  
      1      1      1     93      1      11       11  
[1] "2011-01-01 UTC" "2011-01-02 UTC" "2011-01-03 UTC" "2011-01-04 UTC" "2011-01-05 UTC"  
"2011-01-06 UTC"  
[7] "2011-01-07 UTC" "2011-01-08 UTC" "2011-01-09 UTC" "2011-01-10 UTC"
```

Spatial aggregation

c4h_space: aggregate gridded climate data to spatial polygons

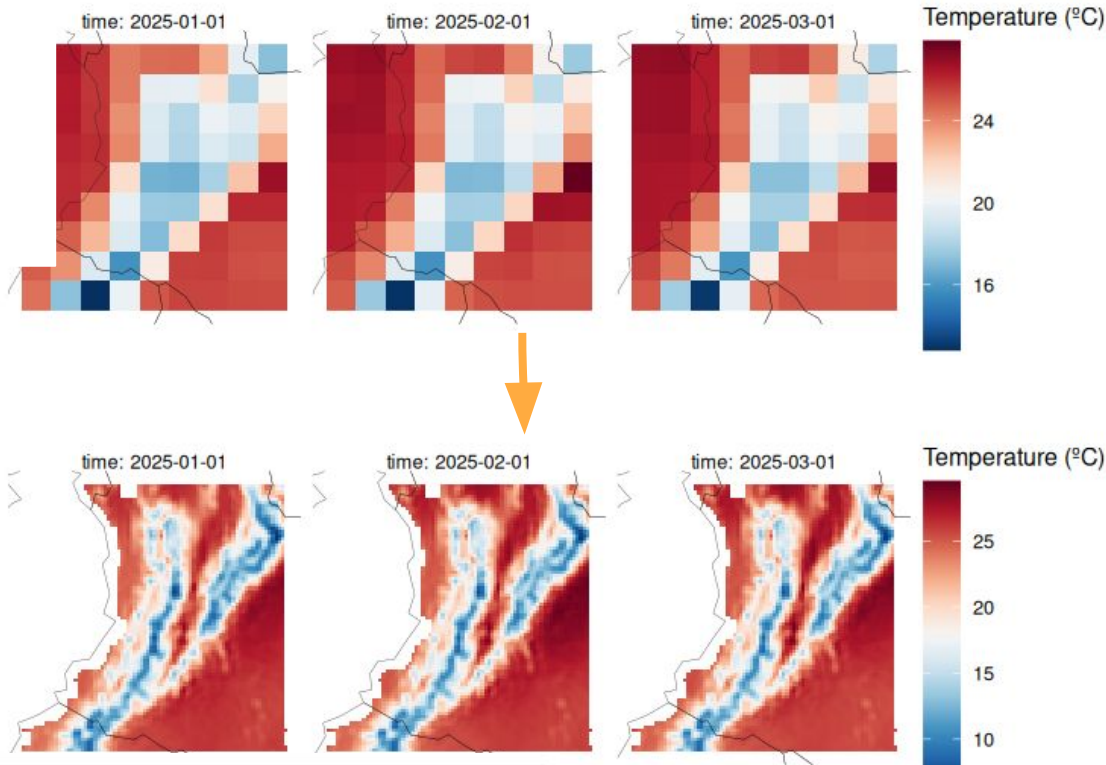
```
{r}  
obs_agg <- c4h_space(obs,  
  areas = shp_path,  
  fun = "mean",  
  weighting = "none",  
  areas_id = "munip_code")
```



Downscaling

c4h_downscale: downscale climate data to finer spatial resolutions (often using observations to adjust the data)

- there are 4 main methods included in **c4h_downscale**
- it is designed to provide helpful notes and messages when it is used



```
{r}  
dwn <- c4h_downscale("Intbc", exp = hcst, obs = obs,  
  bc_method = "evmos",  
  method_remap = "bilinear")
```

Note: If using 'Intbc' and 'points' is not specified, downscaling is to a grid. 'target_grid' has not been specified. Downscaling will be done to the grid of 'obs'.

The key parameter is *downscale_function*, where you must specify the type of downscaling and calibration to be performed — **detailed information is available in the function vignette**

Aims of the tutorial

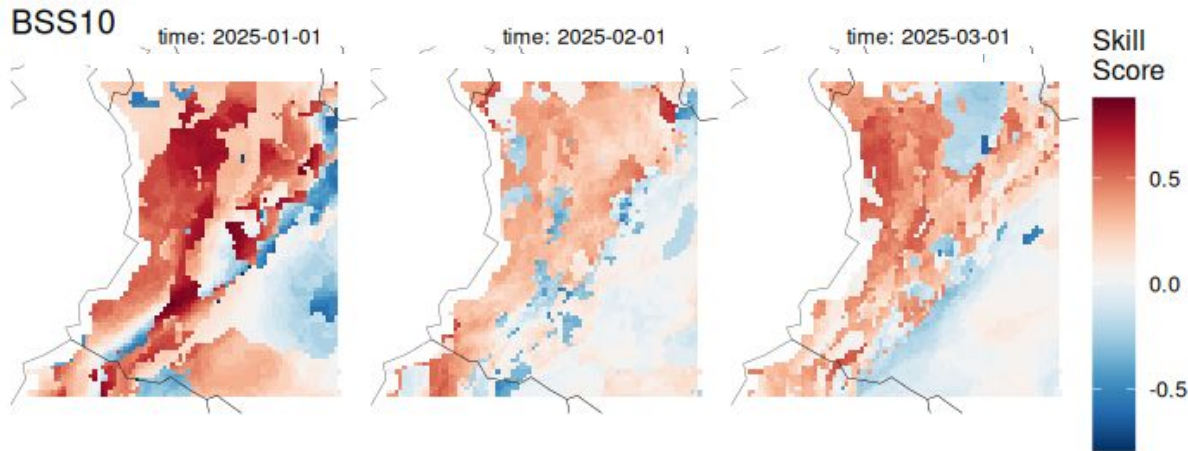
1. Understand sources of climate data
2. Learn how to load climate data into R using clim4health
3. Learn methods to postprocess climate data and apply them using clim4health
4. Learn how to assess forecast quality
5. Learn how to plot climate data using clim4health



Verification

c4h_postprocess: calculate a variety of metrics to assess the quality of a forecast

Key idea: compare the hindcast and observations - how much do we trust our forecast?



```
{r}  
skill <- c4h_postprocess(exp = hcst, |  
                        obs = obs,  
                        metrics = c("BSS", "CRPSS"),  
                        brier_thresholds = c(0.1, 0.9))
```

The key parameter is *metrics*, where you can specify a list of all the metrics to be calculated in the skill assessment — **detailed information is available in the function vignette**

Aims of the tutorial

1. Understand sources of climate data
2. Learn how to load climate data into R using clim4health
3. Learn methods to postprocess climate data and apply them using clim4health
4. Learn how to assess forecast quality
5. Learn how to plot climate data using clim4health



Plotting

c4h_plot: plot your climate data and skill assessments!

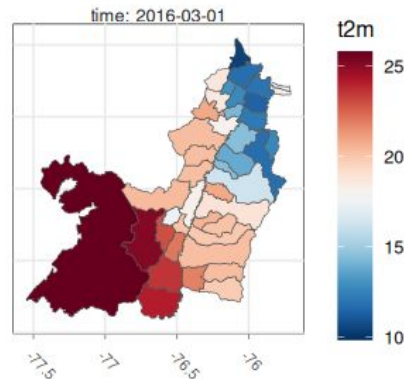
```
{r}  
c4h_plot(fcst, time = 1:3, ensemble = 1:3)|
```

`c4h_plot(data)` — simply plot all the data (this could be many dimensions!)

Add additional parameters to:

- slice dimensions
- take an ensemble mean
- choose your colour palette
- add the boundaries of the region you are interested in
- and more...

```
{r}  
aoi <- sf::st_read(paste0(clim4health_path, "/inst/extdata/areas/munip_vallecauca.gpkg"))  
c4h_plot(fcst, time = 1:3, ensemble = TRUE, boundaries = aoi, coordgrid = TRUE,  
          palette = "Viridis")
```



Additional functions

c4h_get

download climate data (reanalysis, observations and seasonal forecasts) from the Copernicus Climate Data Store

c4h_index

calculation of simple indices (threshold based)

c4h_convert_units

convert variable units

c4h_convert

transform s2dv to common formats

c4h_save

save as csv or grid

Additional functions

c4h_get

download climate data (reanalysis, observations and seasonal forecasts) from the Copernicus Climate Data Store

c4h_index

calculation of simple indices (threshold based)

c4h_convert_units

convert variable units

c4h_convert

transform s2dv to common formats

c4h_save

save as csv or grid



We'll use these in the tutorial!

Additional functions

c4h_get

download climate data (reanalysis, observations and seasonal forecasts) from the Copernicus Climate Data Store

c4h_index

calculation of simple indices (threshold based)

c4h_convert_units

convert variable units

We'll use these in the tutorial!



c4h_convert

transform s2dv to common formats

c4h_save

save as csv or grid

Coming soon!



Let's get started

1. We can now open the Docker - remember:
 - a. Run the container either through the terminal/powershell or from Docker Desktop - don't forget to add the optional settings
 - b. Go to **`http://localhost:8080`**
2. Please follow **HARMONIZE_training.Rmd**
3. Let us know if something doesn't work or doesn't make sense!
 - a. Any unclear error messages?
 - b. Any bugs?
 - c. Anything you would like to see added?

Thanks!



emily.ball@bsc.es

Let's get started

1. We can now open the Docker - remember:
 - a. Run the container either through the terminal/powershell or from Docker Desktop - don't forget to add the optional settings
 - b. Go to **`http://localhost:8080`**
2. Please follow **HARMONIZE_training.Rmd**
3. Let us know if something doesn't work or doesn't make sense!
 - a. Any unclear error messages?
 - b. Any bugs?
 - c. Anything you would like to see added?

Useful commands

`str(fcst)` — prints extended information about the elements of the `s2dv_cube` - useful to see where all the information is stored within the `s2dv_cube`

`dim(fcst$data)` or **`fcst$dims`** — print the dimensions of the data

`summary(fcst$data)` — print a summary of the data

emily.ball@bsc.es

Workflow 2

We have found a bug!

If you run into an error “MergeDims” please run:

```
library(CSTools)
```

Please DON'T use quantile mapping when requested...

Instead, try a simple bias correction - instead of `bc_method = “evmos”` you will need `bc_method = “bias”`

emily.ball@bsc.es

Workflow 3

We have found errors in the solutions!

When loading the data, use **obs_hourly_path**,
not **obs_path**

When aggregating to weekly, use the
obs_hourly s2dv_cube, not **obs**

When asked to make a threshold, try
thresholding the data at 25°, not 30°.

emily.ball@bsc.es

How can clim4health improve?

Today you have used the very first version of clim4health.

To keep improving and making the tool more useful to a bigger audience, we would love to get feedback from users.

Please fill in this form in much detail as possible.

<https://tinyurl.com/HARMONIZE-feedback>

