MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÕES
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

Heat Wave Case: diagnoses and prediction

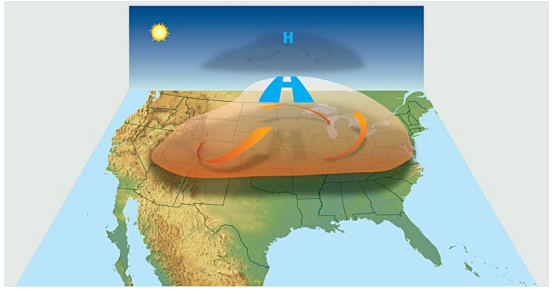
Glícia Ruth Garcia de Araújo//Josefa Morgana Viturino de Almeida

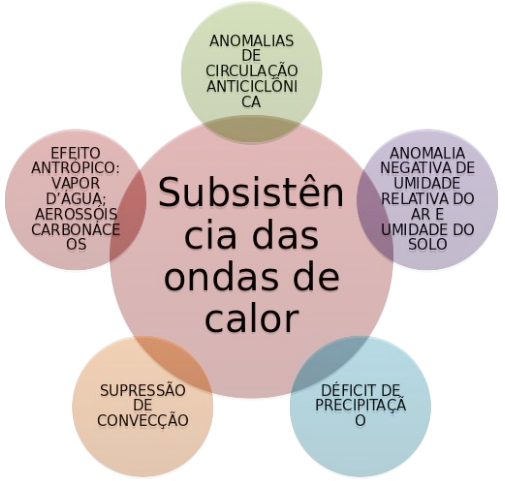
TRAINING COURSE ON WEATHER FORECASTING - AND BEYOND

26 DE Novembro DE 2025

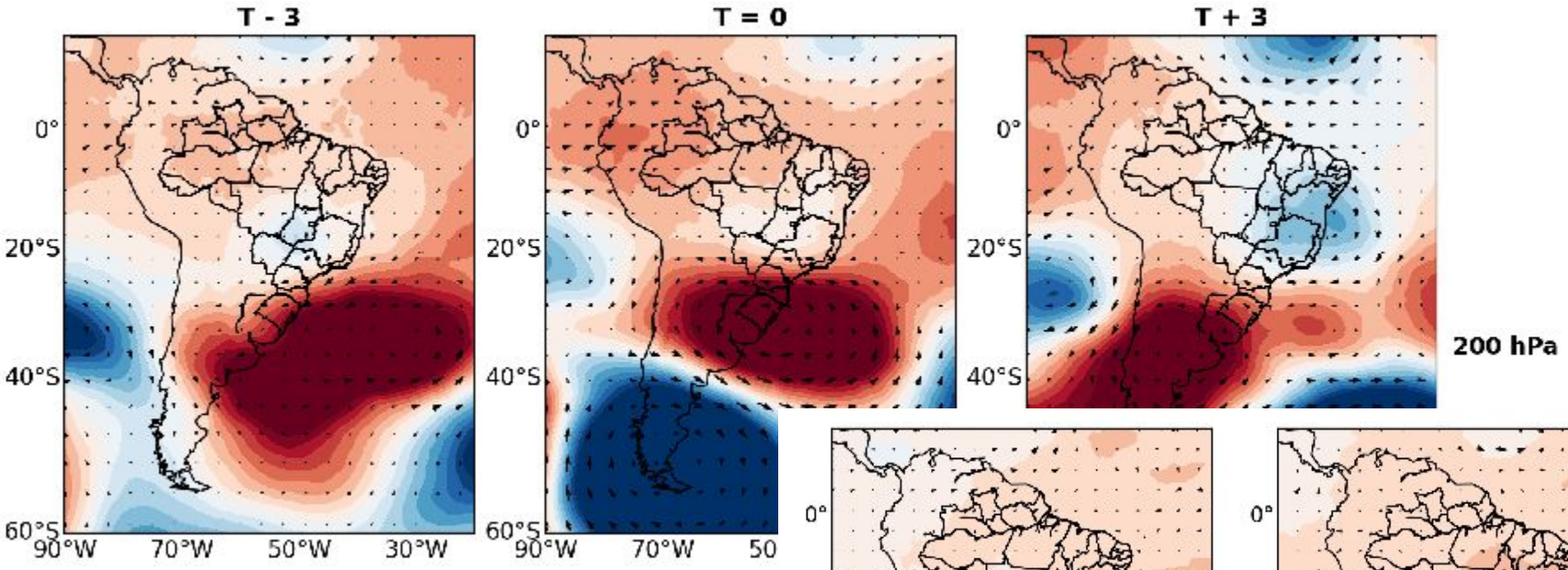
Definition

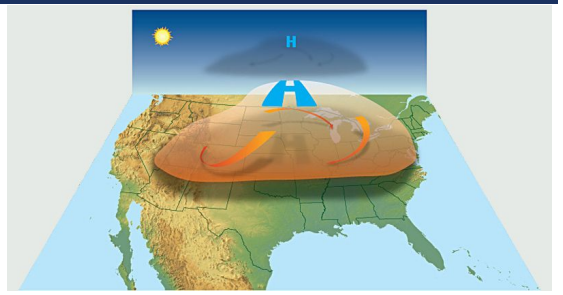
- World Meteorological Organization (WMO) – extended period of maximum and minimum temperatures above an intensity threshold according to the meteorological conditions of a region.
- WMO has used a specific definition for a heatwave that involves a 5 degrees Celsius increase above the average maximum temperature. However…
- There is no consensus in the literature regarding spatial extent, minimum number of days of duration, and intensity threshold.
- Criteria most used in studies: Tmax above the 90th percentile persisting for at least 3 or 5 days.

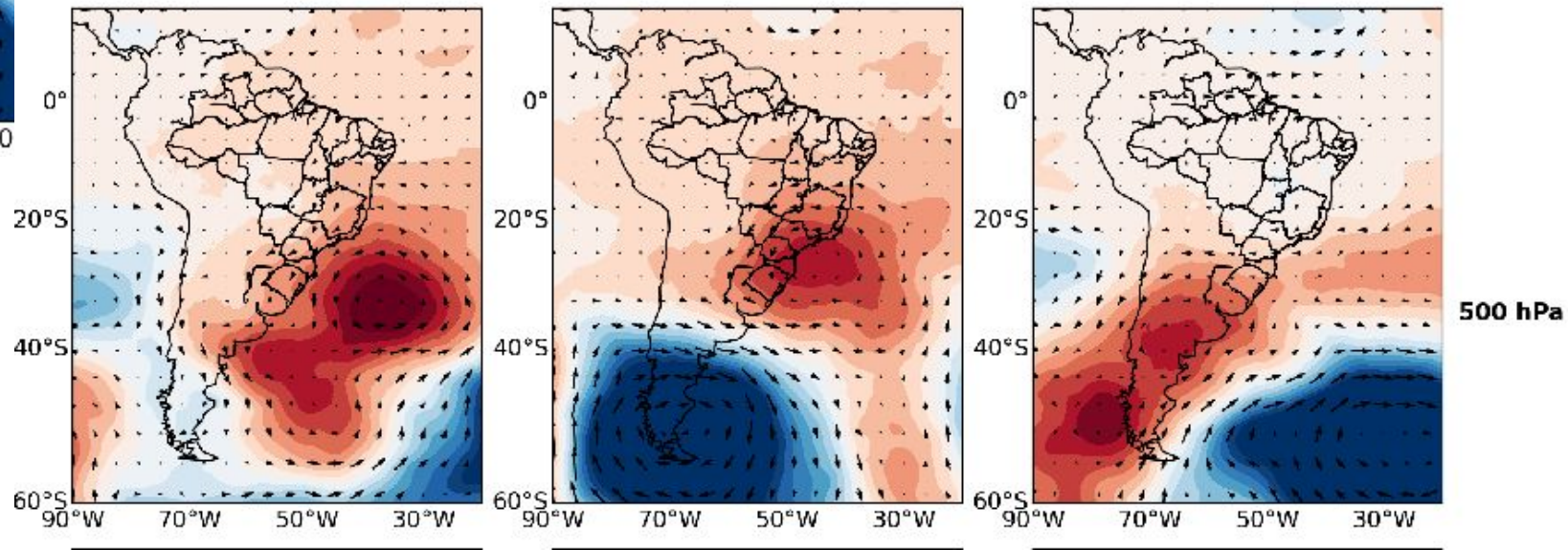




Large-scale situation







Presence of an anticlonic anomaly at medium and high levels.

Source: Araújo (2025)

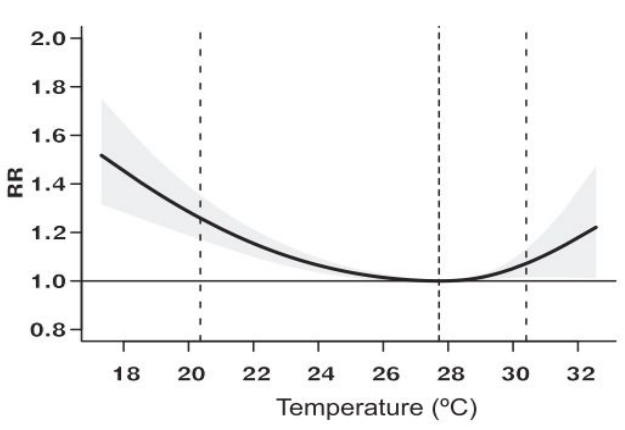
Heat waves and their impacts

Fonte: https://wmo.int/topics/heatwave

Heat waves amplify many risks (economic and/or health-related):

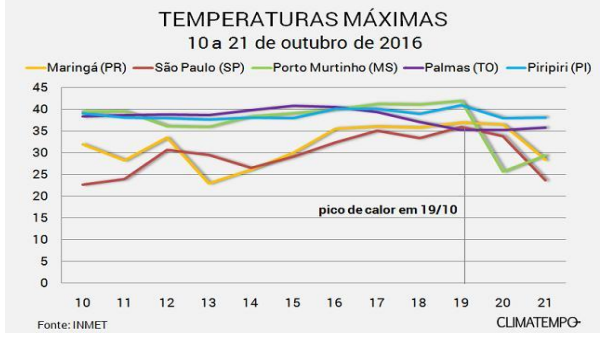
- increased human mortality/hospitalizations
- drought
- water quality
- wildfires
- energy consumption
- and agricultural losses

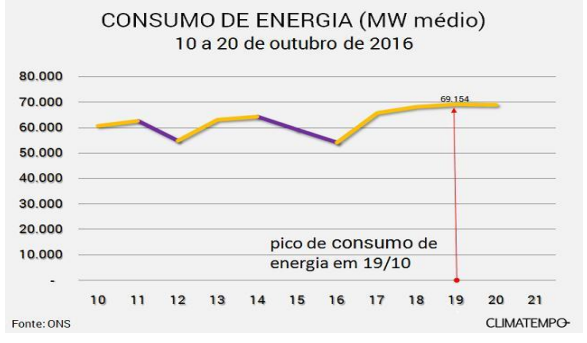
Taxa de mortalidade vs. Temp.



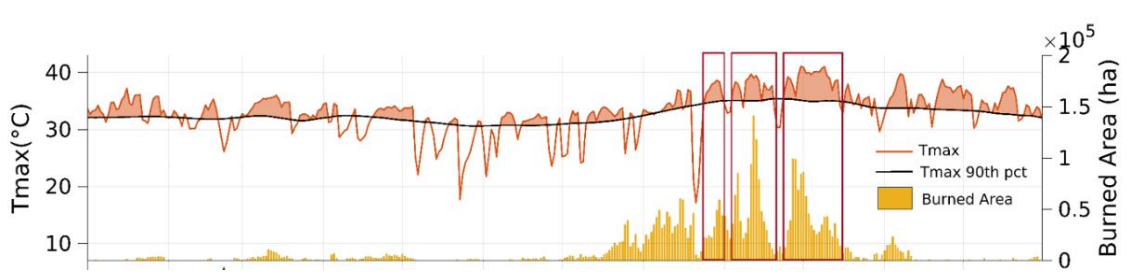
Silveira et al. (2019)

Impactos no setor energético





Aumento de queimadas em períodos extremos de Tmax

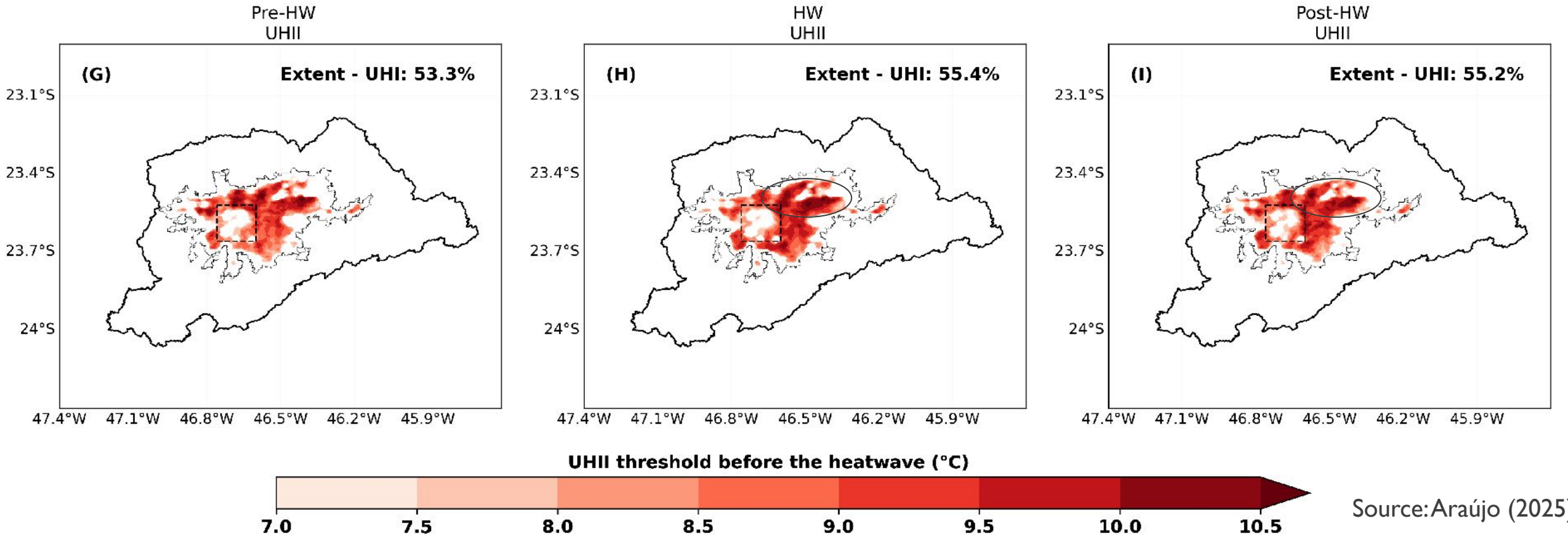


Libonati et al. (2022)

Heat waves and their impacts

Greater impacts in cities: intensification of the urban heat island effect during heat wave events.

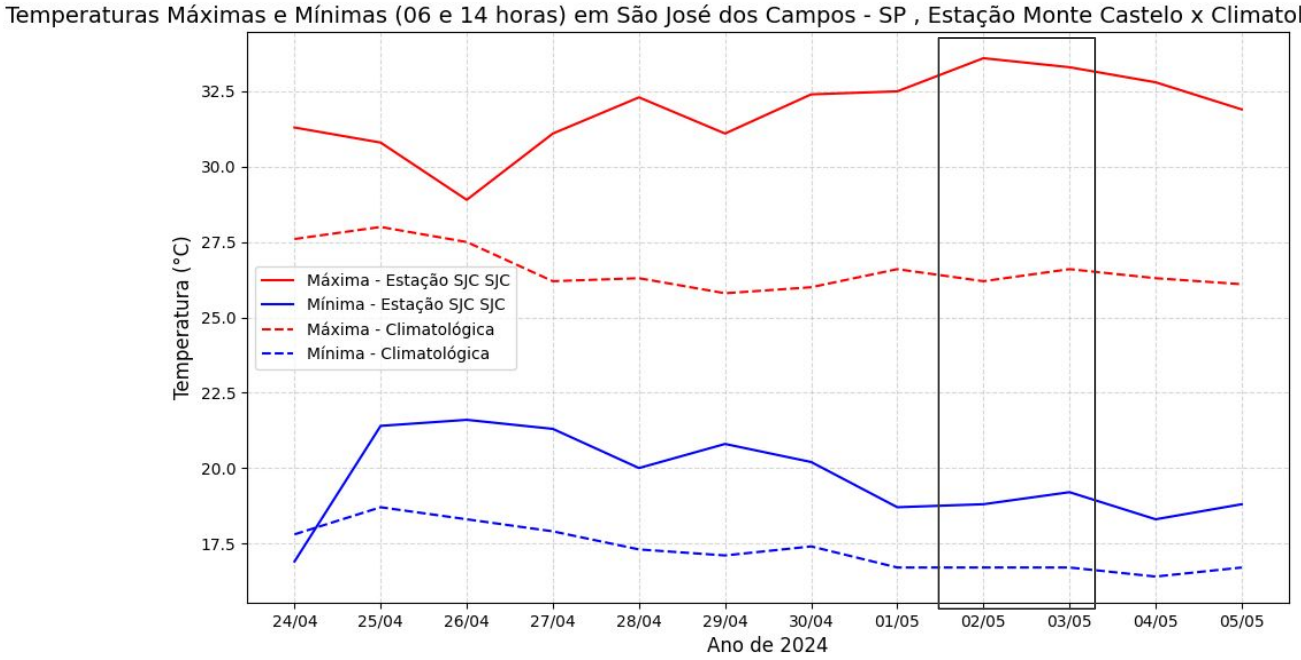
São Paulo Metropolitan Region



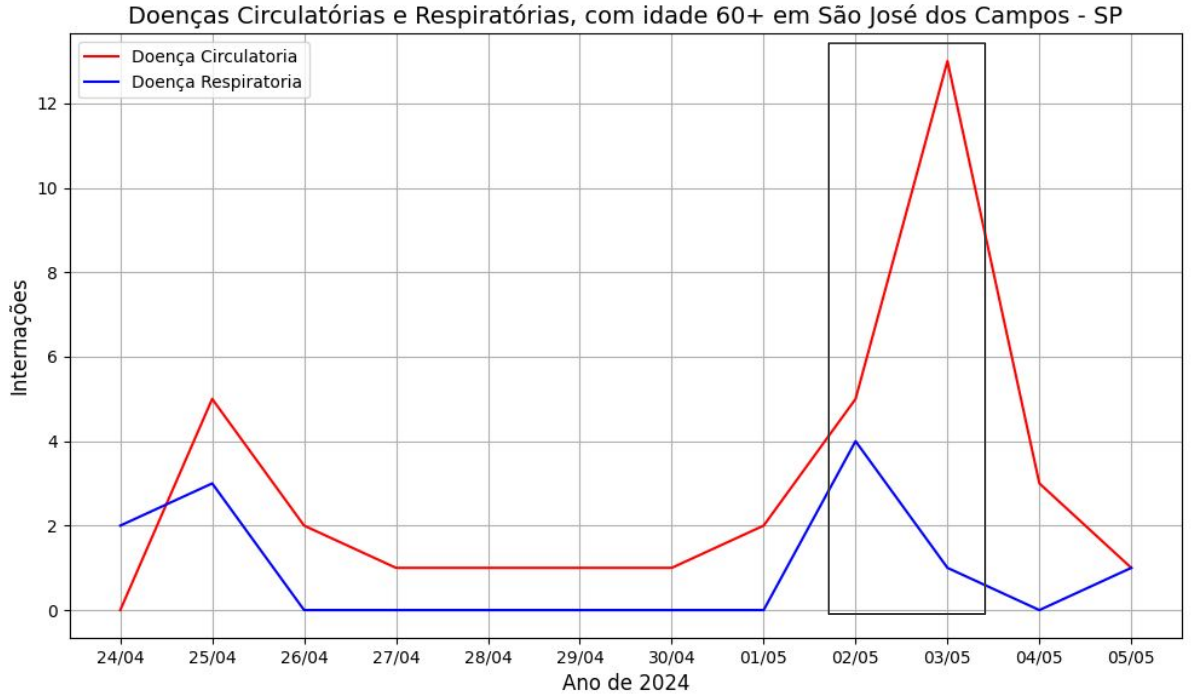
Source: Araújo (2025)

Heat waves and their impacts

Increase in hospitalizations (respiratory and circulatory)

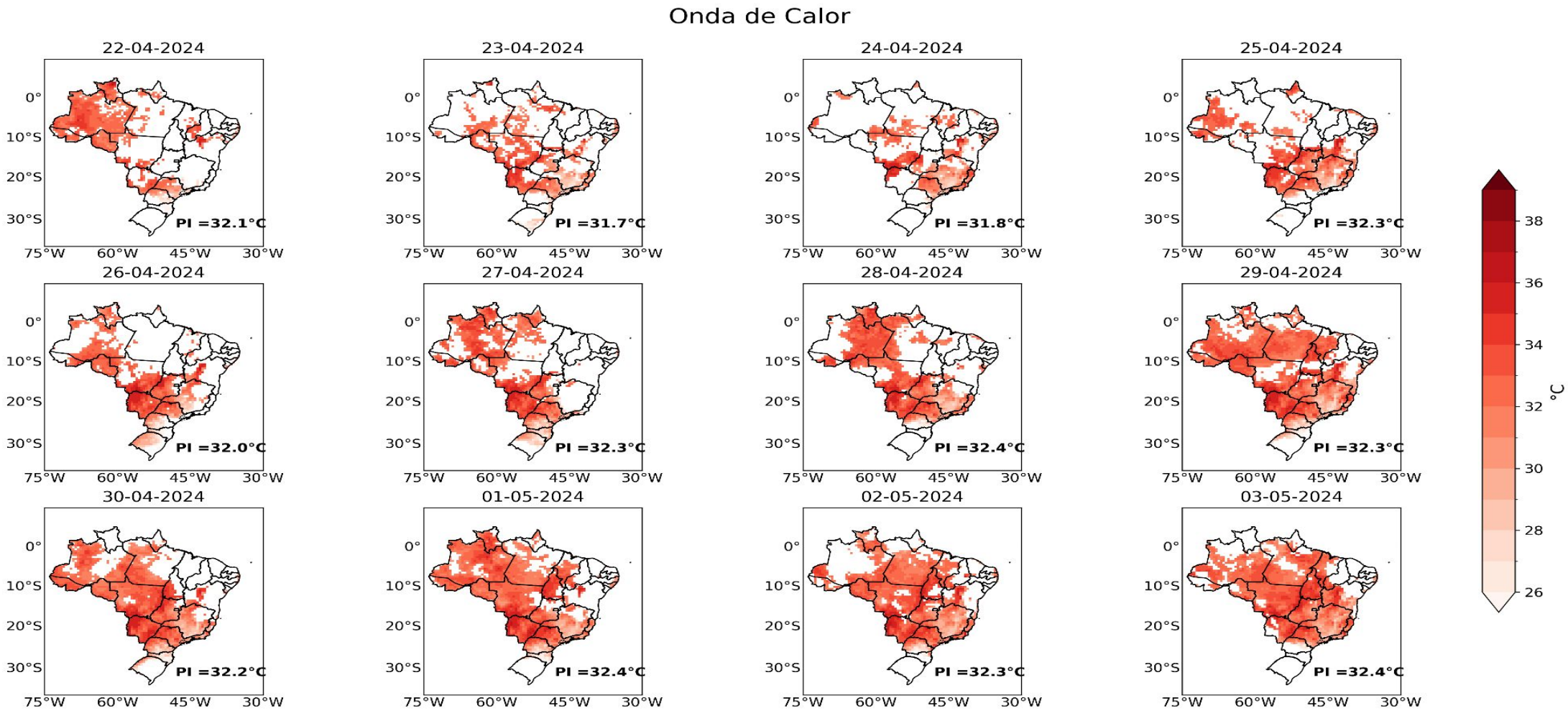


Heat wave between **April 28 and May 5**, 2024 - São José dos Campos - SP

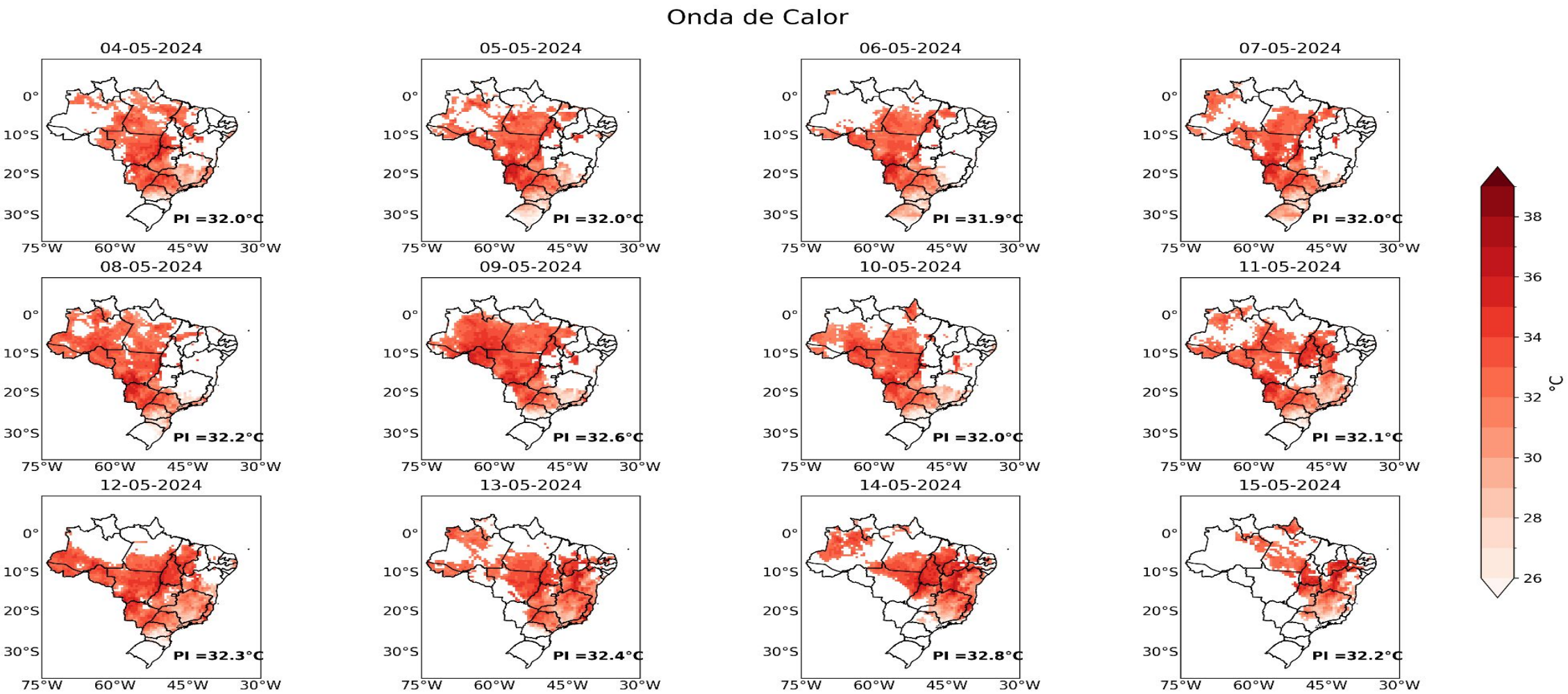


Source: project by student Júlia Teixeira Santos and Ariane Frassoni

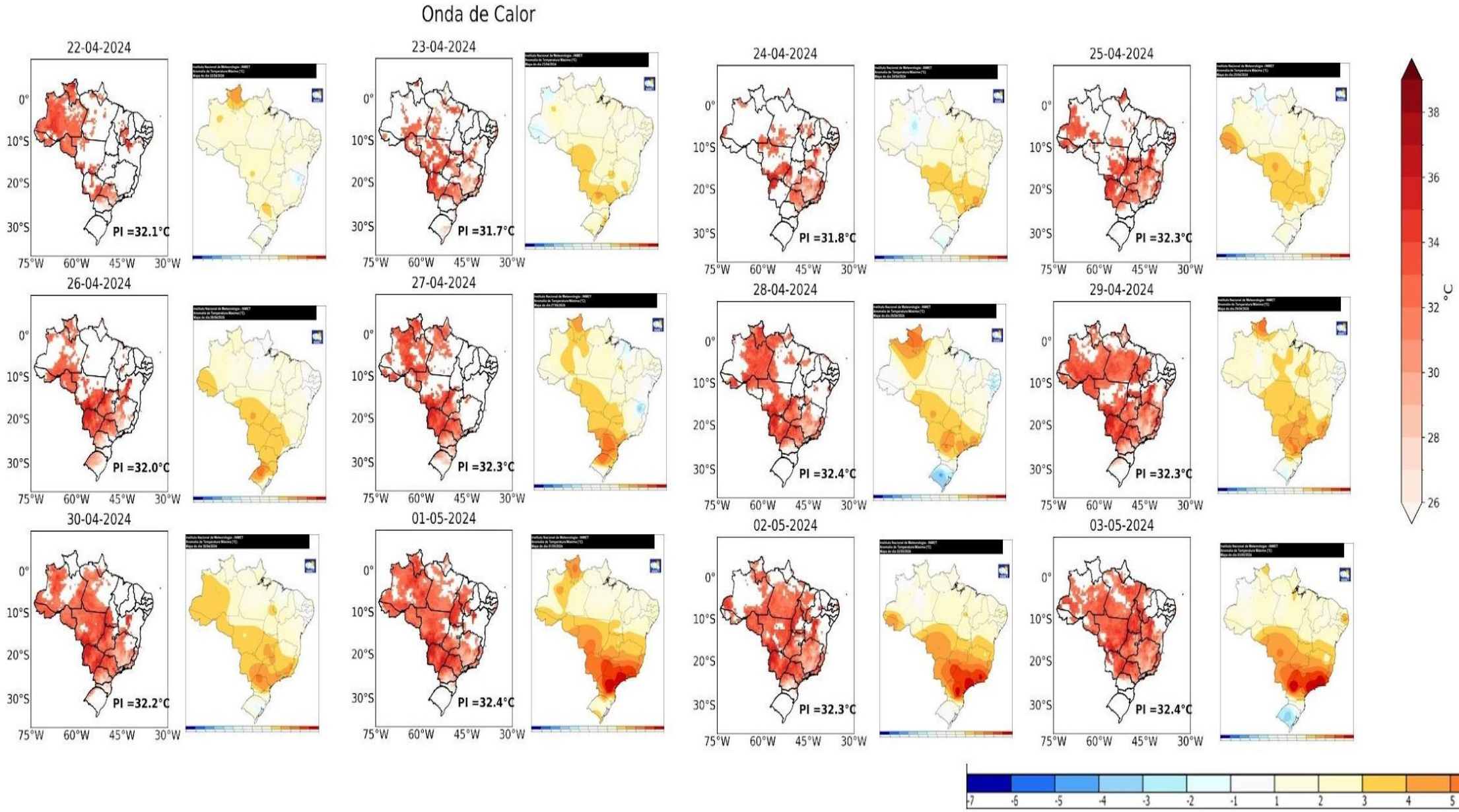
Heat wave case



Heat wave case

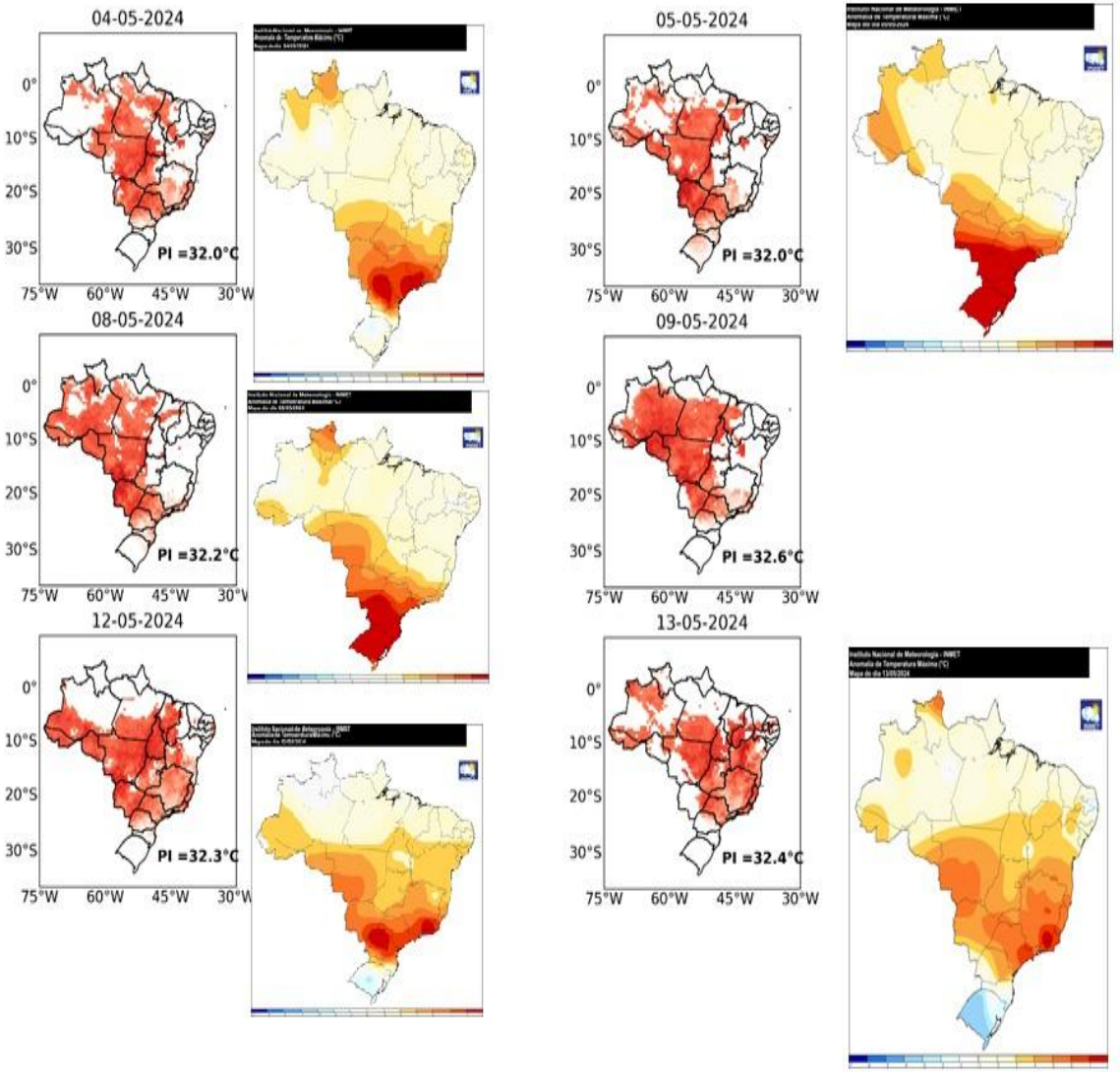


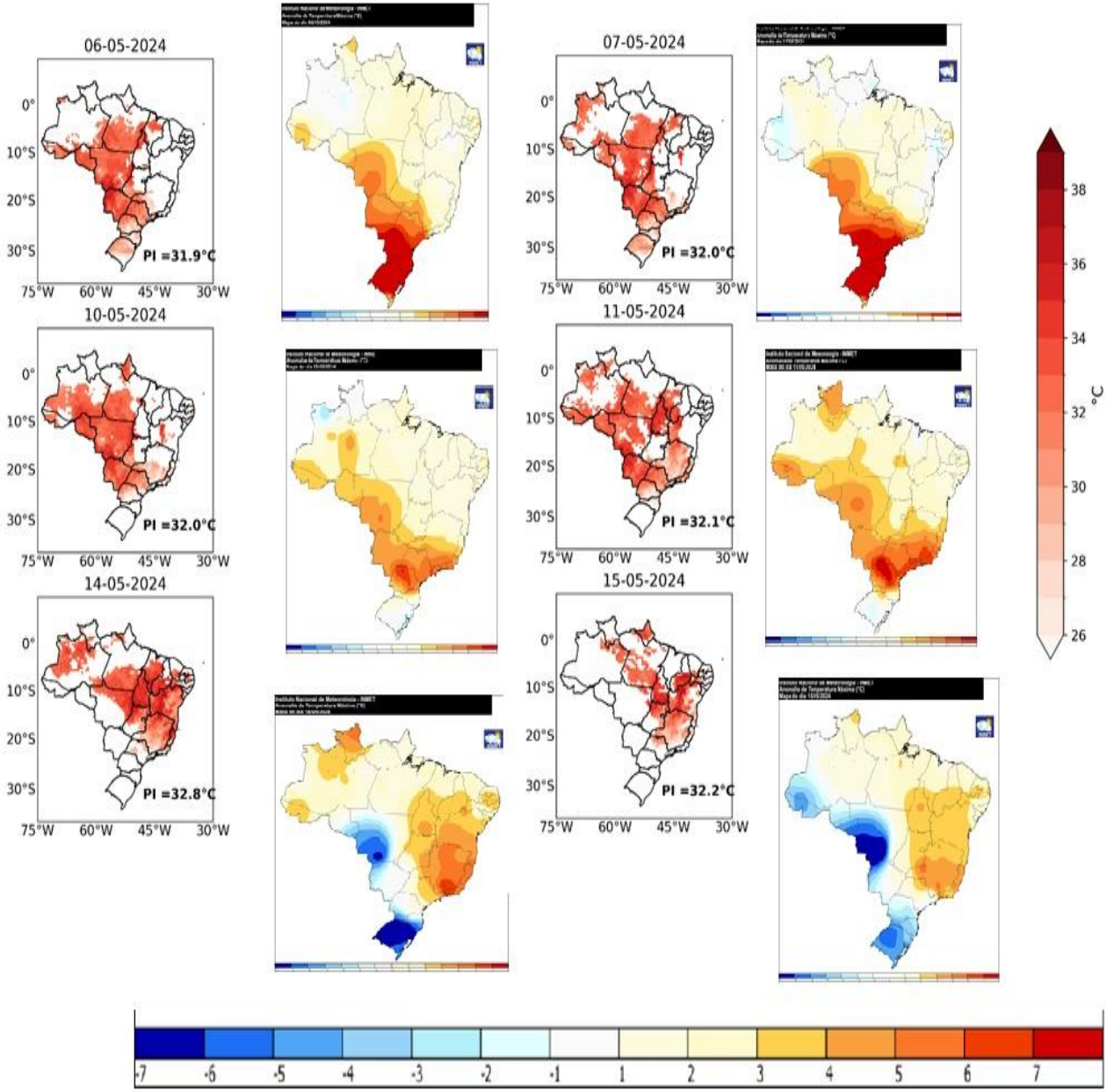
validation


Onda de Calor
22-04-2024
PI =32.1°C
23-04-2024
PI =31.7°C
24-04-2024
PI =31.8°C
25-04-2024
PI =32.3°C
26-04-2024
PI =32.0°C
27-04-2024
PI =32.3°C
28-04-2024
PI =32.4°C
29-04-2024
PI =32.3°C
30-04-2024
PI =32.2°C
01-05-2024
PI =32.4°C
02-05-2024
PI =32.3°C
03-05-2024
PI =32.4°C
0°
10°S
20°S
30°S
75°W
60°W
45°W
30°W
38
36
34
32
30
28
26
°C
-7
-6
-5
-4
-3
-2
-1
1
2
3
4
5
6
7

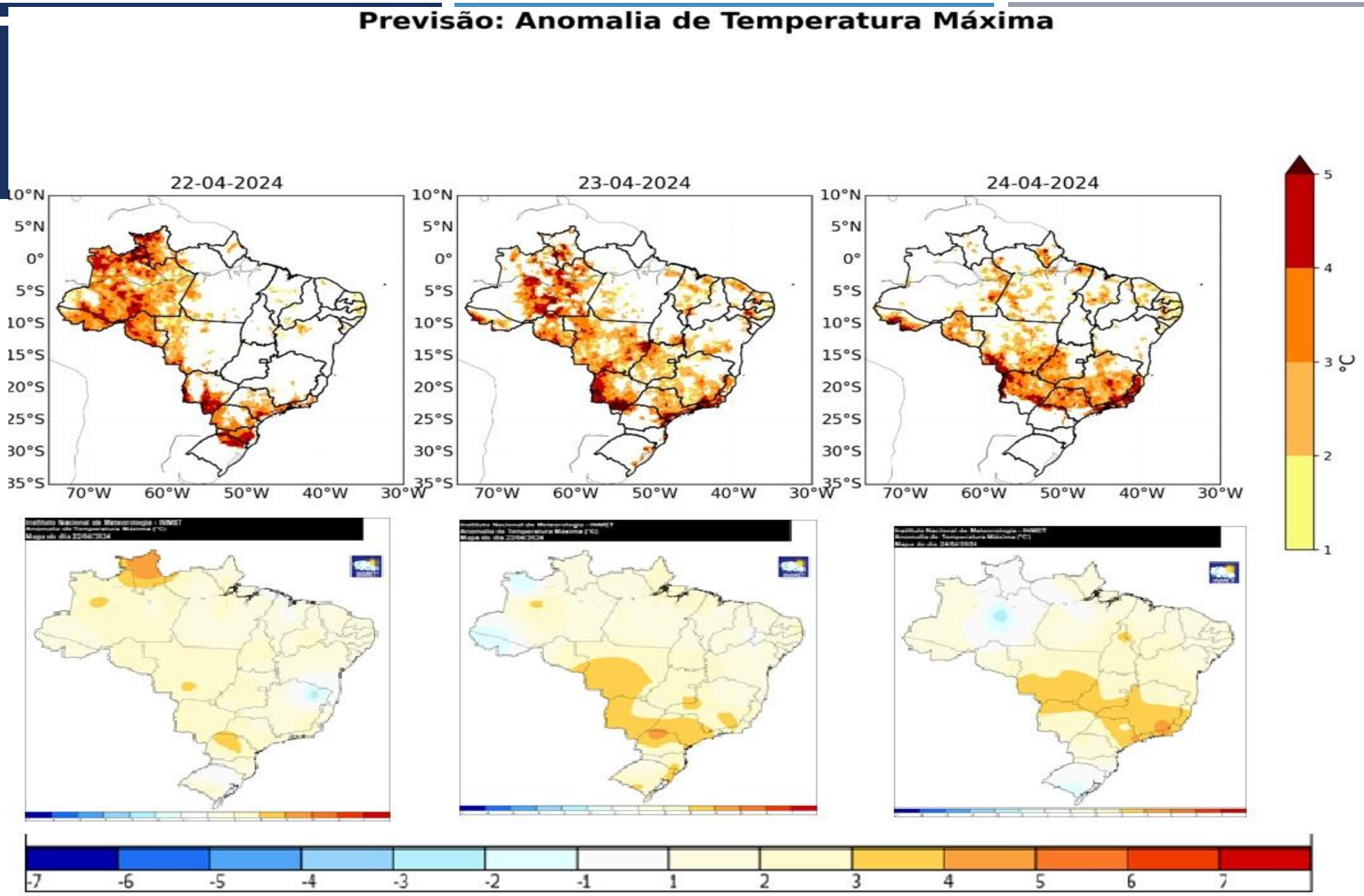
validation

Onda de Calor





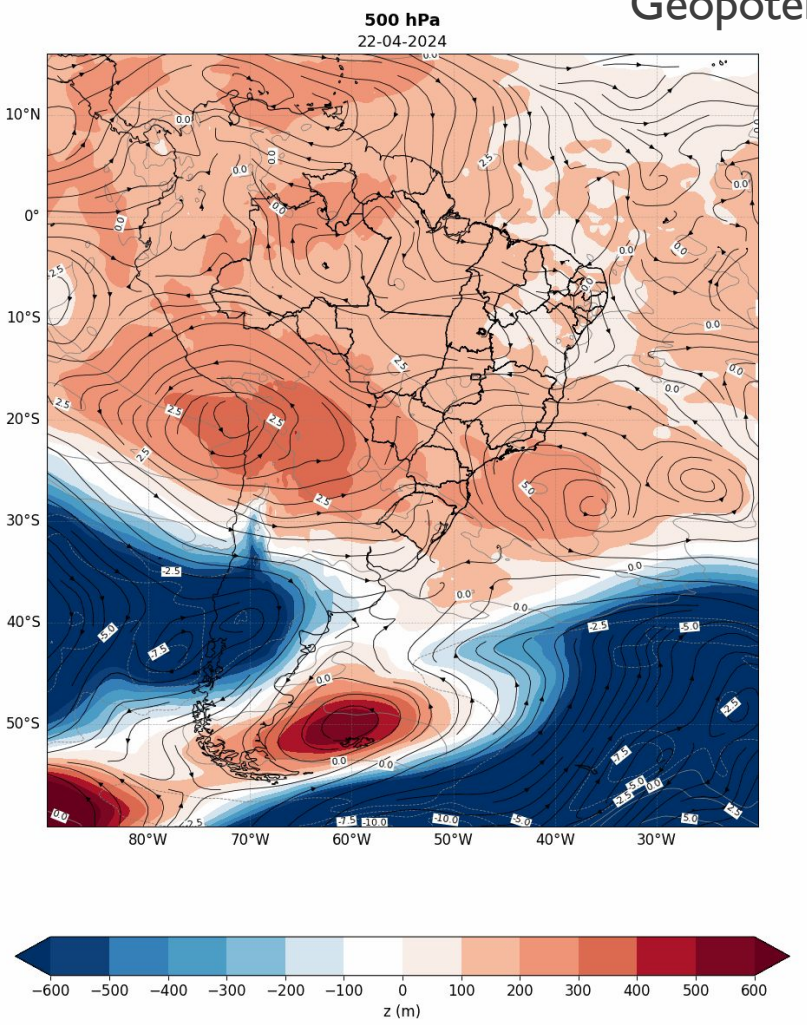
validation


Previsão: Anomalia de Temperatura Máxima
22-04-2024
23-04-2024
24-04-2024
10°N
5°N
0°
5°S
10°S
15°S
20°S
25°S
30°S
35°S
70°W
60°W
50°W
40°W
30°W
°C
5
4
3
2
1
-7
-6
-5
-4
-3
-2
-1
1
2
3
4
5
6
7

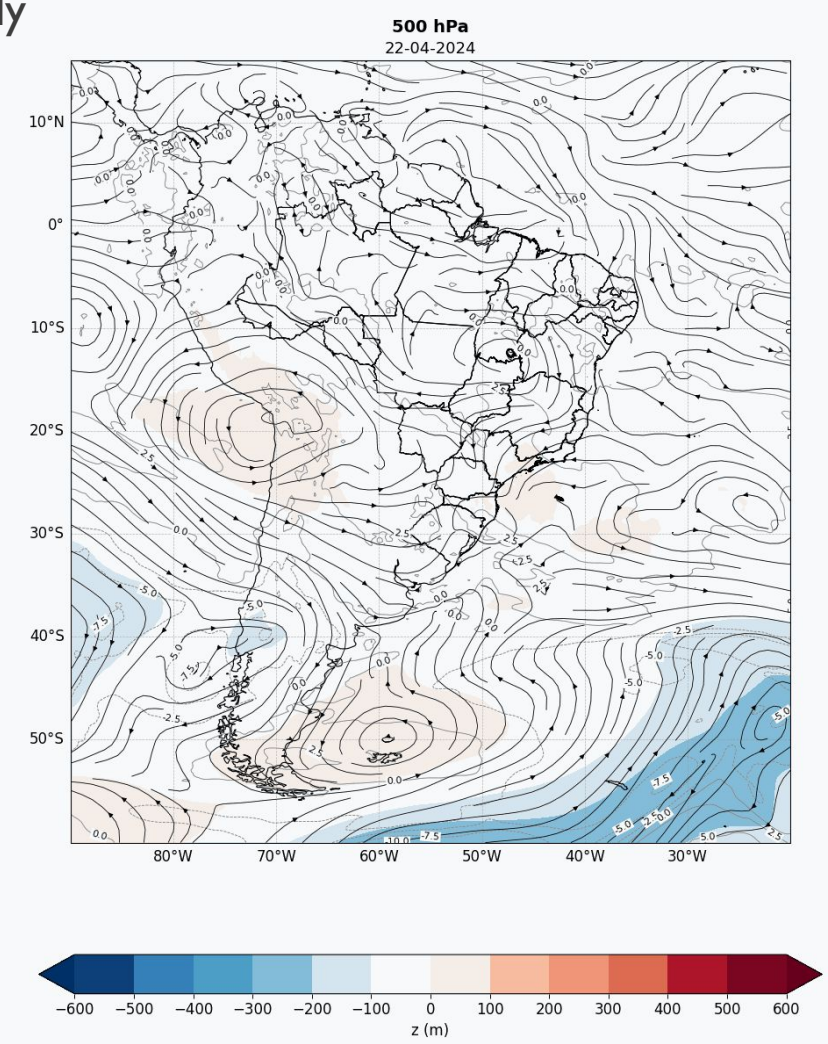
Heat wave case

Geopotential height anomaly

ERA5 Reanalysis



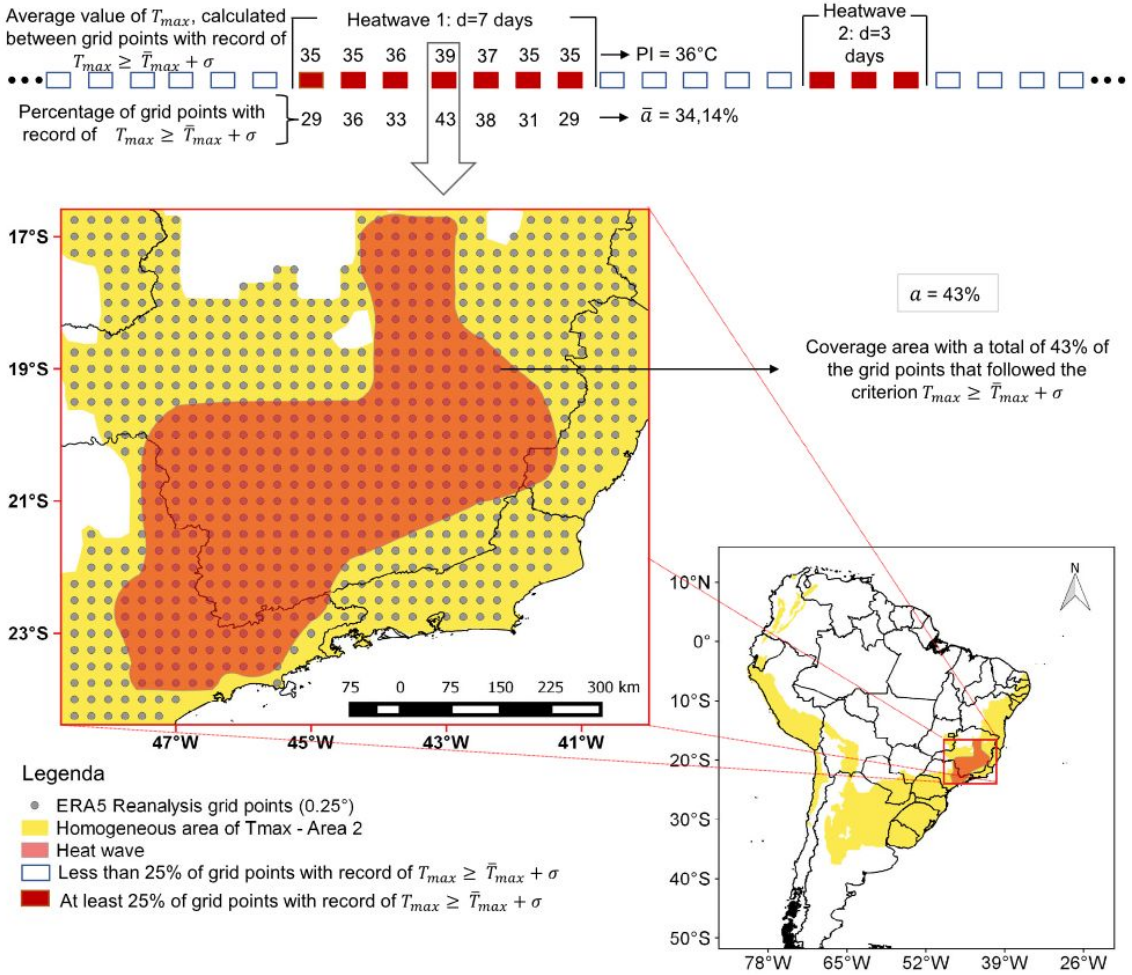
MONAN
18:00 UTC
18h forecast



But...how can we diagnose heat wave events?

Methodology for identifying heat waves - Adapted from Bitencourt et al. 2016

At grid point…



Some adaptations:

- Spatial coverage - from 50% to 25%
- Homogeneous area

Source: Araújo et al. 2022

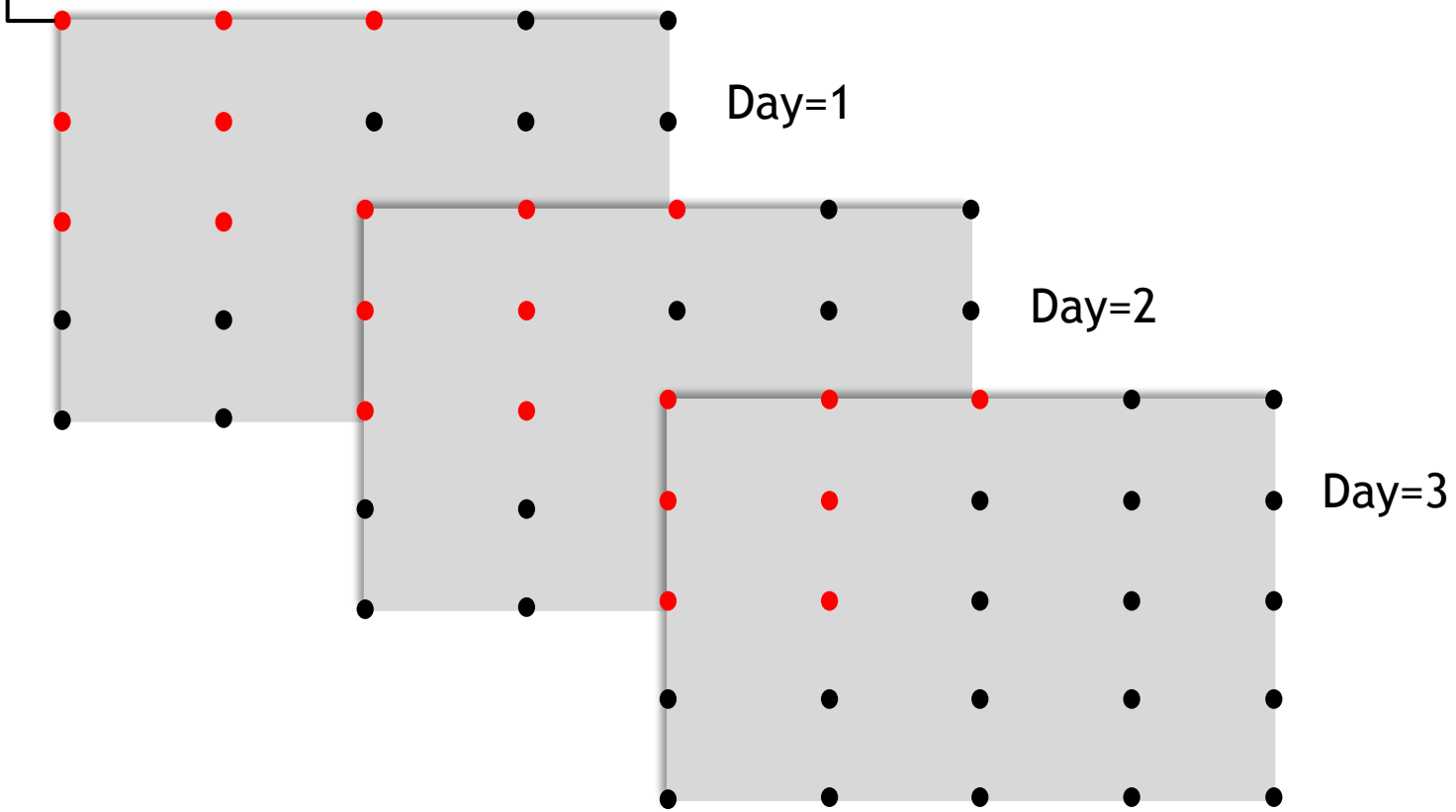
But...how can we diagnose heat wave events?

Algorithm for identifying heat wave episodes

$T_{max} \geq \overline{T}_{max} + \sigma$

In at least 25% of grid points **(spatial coverage)**

Minimum duration of 3 days



Intensity Parameter (IP):

IP > 75th percentile of the historical series

Heat wave

Applying the heatwave algorithm in Python

Identification tool

- Version Python 3.11

Main libraries used::

- xarray (NetCDF)
- pandas (Dataframe - Excel, csv)
- numpy (matrices)
- geopandas (Spatial and georeferenced data)

Python installation:

- Anaconda or Miniconda (virtual environments);

1. Create an environment with all the necessary libraries:
Example: conda create --name <environment_name> python=3.11 xarray[complete] pandas geopandas joblib pyproj matplotlib netCDF4

Note: Some libraries may need to be installed after the environment has been created.

2. List the created environments.
Example: conda env list

```
(HWI-tool) [glicia.garcia@headnode figs]$ conda env list
# conda environments:
#
base                  /opt/spack/opt/spack/linux-rhel8-zen2/gcc-11.2.0/anaconda3-2022.05-q74p53iarv7fk3uin3xzsgfmov7rqomj
HWI-tool              *  /pesq/share/monan/curso_OMM_INPE_2025/.conda/envs/HWI-tool
```

3. Activation of the virtual environment
Example: conda activate <environment_name>

Applying the heatwave algorithm in Python

4. Execution of the codes in the virtual environment.

Steps to obtain a heatwave forecast:

- MONAN model

1. Applying bias correction to forecast data

```
python $path_local/bias_correction.py --model='gfs' --date $date
```

Arguments

2° Algorithm for identifying heat waves in forecasting

```
python $path_local/id_heatwaves_fcst.py --model='gfs' --region $region --date $date
```

3° Visualization of the heat wave forecast

```
python $path_local/mapa_dias_OC_basemap.py --model='gfs' --region $region --date $date
```

Required information:

- Daily Climatology (366 days) (ERA5 reanalysis)
  - 1981-2020 (40 years)

Heat wave forecast:

- Bias correction (At least 15 days of forecast)

Project: https://github.com/gliciaGarcia/HWI-tool.git

Applying the heatwave algorithm in Python

1° Applying bias correction to forecast data

```
python $path_local/bias_correction.py --model='gfs' --date $date
```

Reading of the ERA5 reanalysis data (15 days)

```
def read_era5_reanalysis(dates, dir_out):
        """
        Read ERA5 reanalysis data.
        """
```

Delay 6 days

Function: Bias correction

```
def bias_correction(
        day_fcst,
        dates=list,
        file_obs=str,
        model_fcst=str,
        dir_fcst=str,
        dir_out=str,
):
    """Function: Bias correction.
    :param file_obs: Reference data (15 days of observation).
    :type dates: forecast time
    :param model_fcst: Forecast model.
    :type dir: str
    :param dir_fcst: Directory where the t2m forecast file is located.
    :type dir: str
    :param dir_out: Directory + name of the bias-corrected t2m forecast file.
    :type dir: str
    """
```

```
def arguments():
    parser = argparse.ArgumentParser(prog='bias_correction.py')
    parser.add_argument(
        '--date',
        type=str,
        default=date.today(),
        help='Date: %Y%m%d',
    )
    parser.add_argument(
        '--model',
        type=str,
        default=None,
        help='Forecast model',
    )
    return parser.parse_args()
```

```
    # Removing bias from the forecast
    prev_corr = prev_init_today - final_vies
    prev_corr.to_netcdf(f'{dir_out}/{model_fcst}.t00z.t2m.p18Z.nc')

    return prev_corr
```

Applying the heatwave algorithm in Python

2° Algorithm for identifying heat waves in forecasting

```
python $path_local/id_heatwaves_fcst.py --model='gfs' --region $region --date $date
```

```python
def arguments():
    parser = argparse.ArgumentParser(prog='id_heat_waves_by_P90.py')
    parser.add_argument(
        '--date',
        type=str,
        default=datetime.today(),
        help='Date: %Y%m%d',
    )

    parser.add_argument(
        '--model',
        type=str,
        default=None,
        help='Modelo de previsão: wrf ou gfs ou combinada',
    )

    parser.add_argument(
        '--region',
        type=str,
        default='BR',
        help='Região: CE ou NEB ou BR ou area1-summer',
    )

    return parser.parse_args()
```

Function: Heat wave identification - forecast

```python
def previsao_onda_de_calor(
        day,
        model=str,
        area=str,
        dir_forecast=str,
        dir_climatology=str,
        dir_out=str,
):
    """This script identifies heat wave events in forecast data.

    Args:
        day (str): forecast day
        model (str): model name.
        area (str): region of interest.
        dir_forecast (str): forecast data directory.
        dir_climatology (str): climatology data directory.
        dir_out (str): output data directory.
    """
```

Applying the heatwave algorithm in Python

2° Algorithm for identifying heat waves in forecasting

```
python $path_local/id_heatwaves_fcst.py --model='gfs' --region $region --date $date
```

Function: Heat wave identification - forecast

```
def previsao_onda_de_calor(
        day,
        model=str,
        area=str,
        dir_forecast=str,
        dir_climatology=str,
        dir_out=str,
):
    """This script identifies heat wave events in forecast data.

    Args:
        day (str): forecast day
        model (str): model name.
        area (str): region of interest.
        dir_forecast (str): forecast data directory.
        dir_climatology (str): climatology data directory.
        dir_out (str): output data directory.
    """
```

I. Mask data reference and forecast in the region of interest.

```
# Region Mask
try:
    read_mask = xr.open_dataset(f'{dir_local}/tools/mask_region_{area}.nc').rename({'lon': 'longitude', 'lat': 'latitude'})
except:
    read_mask = xr.open_dataset(f'{dir_local}/tools/mask_region_{area}.nc')
```

```
# Climatology mask
nc = nc.where(mask, np.nan)

# Forecast
nc_prev = xr.open_dataset(f'{dir_forecast}/{model}.t00z.t2m.p18Z.nc')

# Regrid the source dataset using target coordinates
nc_prev = nc_prev.interp(coords=target_coords, method='linear')
nc1 = nc_prev.where(mask, np.nan)
```

Applying the heatwave algorithm in Python

2° Algorithm for identifying heat waves in forecasting

```
python $path_local/id_heatwaves_fcst.py --model='gfs' --region $region --date $date
```

```
def previsao_onda_de_calor(
        day,
        model=str,
        area=str,
        dir_forecast=str,
        dir_climatology=str,
        dir_out=str,
):
```

Function: Heat wave identification - forecast

II. First criterion based on climatology ERA5 $T_{max} \geq \overline{T}_{max} + \sigma$

```
# ------------------------------------------------------------------------------------------------------------
# COUNTING THE NUMBER OF GRID POINTS IN THE REGION
# ------------------------------------------------------------------------------------------------------------
points_land = np.count_nonzero(~np.isnan(tmax_count))      # Total number of grid points over the continent.
print("total de pontos sobre o continente:", points_land, '\n')

# ------------------------------------------------------------------------------------------------------------
# First criterion: clim Tmax + std for each grid point - climatological reference from 1981 to 2020 of ERA5.
# ------------------------------------------------------------------------------------------------------------
P1 = (nc['t2m'] + nc['std']).data

# ------------------------------------------------------------------------------------------------------------
# APPLICATION OF THE CRITERION TMAX > clim Tmax + std WITH A MINIMUM OF 3 CONSECUTIVE DAYS.
# ------------------------------------------------------------------------------------------------------------
crit = np.where(Tmax > P1, Tmax, np.nan)
nc1['crit90'] = (('time', 'latitude', 'longitude'), crit)
```

Applying the heatwave algorithm in Python

2° Algorithm for identifying heat waves in forecasting

```
python $path_local/id_heatwaves_fcst.py --model='gfs' --region $region --date $date
```

```
def previsao_onda_de_calor(
        day,
        model=str,
        area=str,
        dir_forecast=str,
        dir_climatology=str,
        dir_out=str,
):
```

Function: Heat wave identification - forecast

III. Criteria for spatial coverage and minimum number of days (3 days)

```
# Applying the second condition (minimum of three days).
list_index = []
for idx, value_time in enumerate(nc1.time.data):
    value_time = pd.to_datetime(value_time)
    count_valid = np.count_nonzero(~np.isnan(nc1.sel(time=value_time).crit90.data))
    if (count_valid/points_land) > 0.25:  # Spatial extent (default: 0.25).
        list_index.append(idx)

# Eliminating list sequences of indices with a size smaller than 3
list_filter = split_list(list_index)  # Separating by sequence of values
```

```
Module tools.tools_idhw_v2
```

```
# Função para eliminar os índices do time com abrangência menor que 25% do total de pontos válidos
def split_list(mylist):
    # Calculando as diferenças
    d = np.diff(mylist)
    # Quando as diferenças não são 1, salve esta informação
    # Precisamos a +1 para compensar o elemento perdido
    breaks = list(np.arange(len(mylist) - 1)[d != 1] + 1)
    slices = zip([0] + breaks, breaks + [len(mylist)])
    # fatia as listas
    int_list = [mylist[a:b] for a, b in slices]
    filter = [int_list[idx] for idx in range(0, len(int_list))
              if len(int_list[idx]) > 2]  # tirar as listas menores que 3

    return list(filter)
```

Applying the heatwave algorithm in Python

2° Algorithm for identifying heat waves in forecasting

```
python $path_local/id_heatwaves_fcst.py --model='gfs' --region $region --date $date
```

```
def previsao_onda_de_calor(
        day,
        model=str,
        area=str,
        dir_forecast=str,
        dir_climatology=str,
        dir_out=str,
):
```

Function: Heat wave identification - forecast

IV. Calculation of the Intensity Parameter (last criterion applied)

```
    #-------------------------------------------------------------------------------------------
    # CALCULATING THE AVERAGE BETWEEN THE DAYS OF THE EVENT (INTENSITY PARAMETER - PI)
    #-------------------------------------------------------------------------------------------
    P75 = nc.isel(time=idx)['percentil75'].mean(dim=['time', 'latitude', 'longitude']).values

    if PI > P75:
        msg = "Evento de onda de calor identificado! \n"
        # Days that did not pass the heatwave criterion
        dates_without_oc = [list_dates[index] for index in range(len(list_dates)) if index not in idx]
        sel_days = nc1.sel(time=dates_without_oc)
        days_empty = sel_days.where(False, np.nan)

        dataset = xr.concat([evento, days_empty], dim='time')
        dataset = dataset.sortby('time')
    else:
        msg = "Não foi identificado evento de onda de calor! \n"
        dataset = nc1.where(False, np.nan)
    list_datasets.append(dataset)

dataset_final = xr.merge(list_datasets)
dataset_final.to_netcdf(file_out)
```

From the list_filter variable, where the list of days with possible events is stored, the last criterion for identifying the heat wave is checked (PI > P75).

Applying the heatwave algorithm in Python

4° Execution of the codes in the virtual environment.

Execution steps to identify heatwave events in the reference:

- ERA5 reanalysis (European Centre for Medium-Range Weather Forecasts - ECMWF)

1° Heat wave identification algorithm in the reference

```
python $path_local/id_heatwaves.py --date-init $date_i --date-end $date_f --region $region
```

2° Visualization of the heat wave event in the reference

```
python $path_local/plot_reference_heatwave.py --date-init $date_i --date-end $date_f --region $region
```

Applying the heatwave algorithm in Python

1° Heat wave identification algorithm in the reference

```
python $path_local/id_heatwaves.py --date-init $date_i --date-end $date_f --region $region
```

```python
def arguments():
    parser = argparse.ArgumentParser(prog='plot_reference_heatwave.py')
    parser.add_argument(
        '--date-init',
        type=str,
        default=(datetime.today() - timedelta(days=50)).strftime("%Y%m%d"),
        help='Date: %Y%m%d',
    )

    parser.add_argument(
        '--date-end',
        type=str,
        default=(datetime.today() - timedelta(days=6)).strftime("%Y%m%d"),
        help='Date: %Y%m%d',
    )
    parser.add_argument(
        '--region',
        type=str,
        default='BR',
        help='Região: BR ou NEB ou area1-summer',
    )

    return parser.parse_args()
```

Function: Heat wave identification - Reference

```python
def onda_de_calor(
        day_init,
        day_final,
        area=str,
        dir_reference=str,
        dir_climatology=str,
        dir_out=str,
):
    """This script identifies heat wave events in reference data.

    Args:
        day_init (str): start date to find the event.
        day_final (str): final date to find the event
        area (str): region of interest.
        dir_reference (str): reference data directory.
        dir_climatology (str): climatology data directory.
        dir_out (str): output data directory.
    """
```

Time to practice the tool…

We will use the following tutorial:

https://github.com/gliciaGarcia/HWI-tool/blob/main/tutorial_run_tool.md

Important documents and links

Anacondas Installation: https://docs.anaconda.com/anaconda/install/

Python installation tutorial: https://guilherme.readthedocs.io/en/latest/pages/tutoriais/python.html

Download Visual Studio Code: https://code.visualstudio.com/download

Python applications in Geosciences: https://drive.google.com/file/d/15_62F9lb21XDhCsYL_YoKIuuAATNWpNw/view

Python Libraries:
https://docs.xarray.dev/en/stable/
https://pandas.pydata.org/docs/
https://numpy.org/doc/2.1/
https://geopandas.org/en/stable/docs.html

Course documents:
https://dataserver.cptec.inpe.br/dataserver_dimnt/monan/curso_OMM_INPE_2025/Validation/HeatWave/

References

ARAÚJO, G. R. G., FRASSONI, A., SAPUCCI, L. F., BITENCOURT, D., & de BRITO NETO, F. A. Climatology of heatwaves in South America identified through ERA5 reanalysis data. *International Journal of Climatology*, 2022.

BITENCOURT, D. P.; FUENTES, M. V.; MAIA, P. A.; AMORIM, F. T. Frequência, Duração, Abrangência Espacial e Intensidadedas Ondas de Calor no Brasil. Revista Brasileira de Meteorologia, v. 31, n. 4, p. 506–517, 2016. ISSN 19824351.

BITENCOURT, D. P.; FUENTES, M. V.; FRANKE, A. E.; SILVEIRA, R. B.; ALVES, M. P. The climatology of cold and heat waves in Brazil from 1961 to 2016. International Journal of Climatology, v. 40, p. 2464–2478, 2019. ISSN 10970088.

GEIRINHAS, J. L.; TRIGO, R. M.; LIBONATI, R.; COELHO, C. A. S.; PALMEIRA, A. C. Climatic and synoptic characterization of heat waves in Brazil. International Journal of Climatology, v. 38, n. 4, p. 1760–1776, 2018. ISSN 08998418. Disponível em: <http://doi.wiley.com/10.1002/joc.5294>.

SANTOS, J. G. M.; SIMÕES, J. L. D. R.; RAMOS, D. N. S.; EICHHOLZ, C. W. Aplicações de Python em geociências. São José dos Campos: INPE, 2022. 174 p. IBI: <8JMKD3MGP3W34T/46RTMU5>. (sid.inpe.br/mtc-m21d/2022/05.10.16.53-PUD). Disponível em: ibi:8JMKD3MGP3W34T/46RTMU5.

SILVEIRA, I. H.; OLIVEIRA, B. F. A.; CORTES, T. R.; JUNGER, W. L. The effect of ambient temperature on cardiovascular mortality in 27 Brazilian cities. Science of The Total Environment, v. 691, p. 996–1004, nov 2019. ISSN 00489697. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S0048969719330669>.

WORLD METEROLOGICAL ORGANIZATION - WMO. Guidelines on analysis of extremes in a changing climate in support of informed decisions for adaptation. Data Climate, 2009.

Contacts

Institutional: glicia.garcia@inpe.br
Personal email: glicia.garcia123@gmail.com