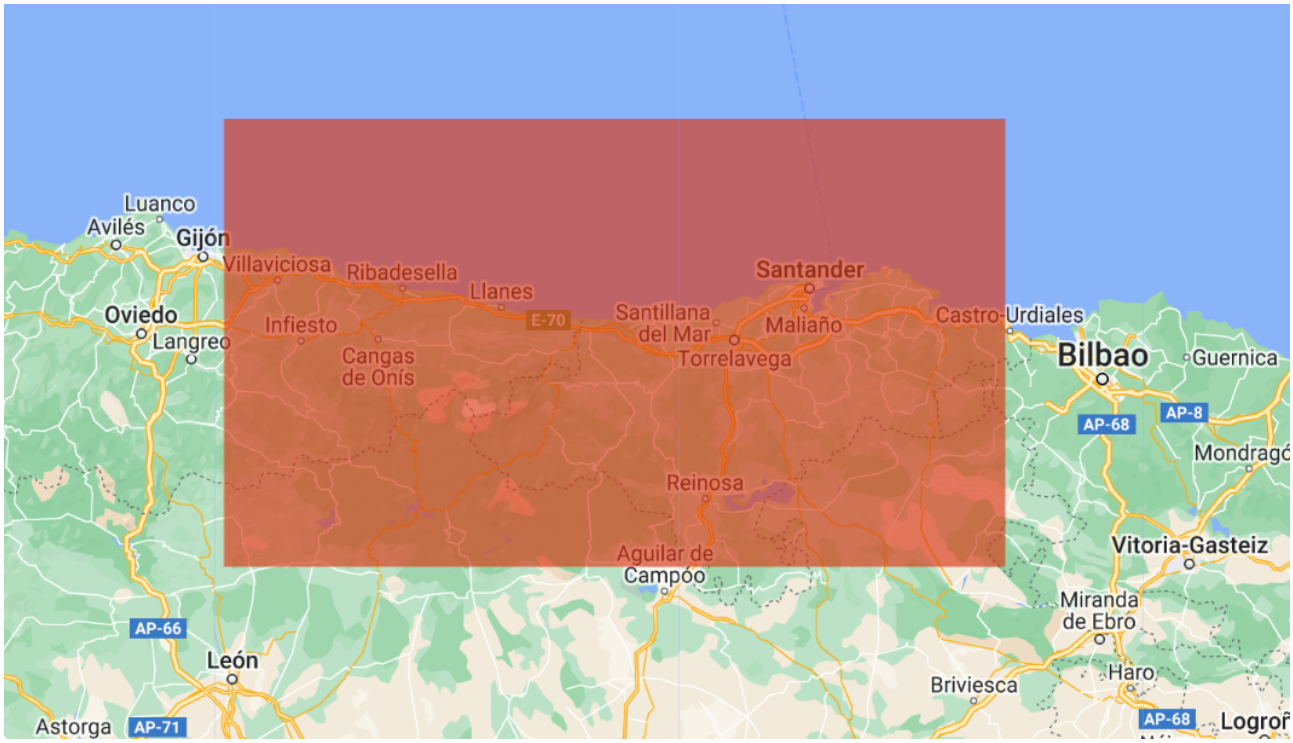
Simone Boesso 1800408 Exercise 7

Drought monitoring

This practical exercise shows how satellite based rainfall data can be used for drought monitoring. A deviation (anomaly) in rainfall from its Long Period Average (LPA) of 30 years or more can be used as an indicator of drought.
For any given region and time period, we can calculate the percentage deviation of rainfall (precipitation) from long-term average and determine which regions are likely experiencing drought.

The area I choose is the following:



In particular, with the following coordinates:

```
Imports (1 entry)
  var geometry: Polygon, 4 vertices
    type: Polygon
    coordinates: List (1 element)
      0: List (5 elements)
        0: [-5.596970496794897,42.84729990923986]
        1: [-3.2307900768730224,42.84729990923986]
        2: [-3.2307900768730224,43.83603678091719]
        3: [-5.596970496794897,43.83603678091719]
        4: [-5.596970496794897,42.84729990923986]

    geodesic: false
```

The steps I followed are the following:

- loading the CHIRP data
- the computation for the RFn:
- Computation of the current monthly precipitation for each month of a specific year (in mm).
- Compute the monthly rainfall deviation (RFdev) in percentage

So to achieve the goal I have to compute the long term monthly precipitation mean for each month.

```
// Long Period Average (LPA) should by 30 years or more
// List of 30 years
var lpaYears = ee.List.sequence(1986, 2016);
var months = ee.List.sequence(1, 12);
print('lpaYears', lpaYears);
print('month', months);

// Iterate over the years and the months with a nested map
// to create a (nested) monthly totals list
// which will be converted in a monthly totals collection
var monthlyImages_nested_list = lpaYears.map(function(year) {
  return months.map(function(month) {
    var filtered = chirps
      .filter(ee.Filter.calendarRange(year, year, 'year'))
      .filter(ee.Filter.calendarRange(month, month, 'month'));
    var monthly = filtered.sum();
    return monthly.set({'month': month, 'year': year});
  });
});

print ('monthlyImages_nested_list',monthlyImages_nested_list);

var monthlyImages_list = monthlyImages_nested_list.flatten();
print ('monthlyImages_list', monthlyImages_list);

// All this work only to compute this collection!!
// We now have 1 image per month for entire long-period duration
// and not 6 (pentadal) images per month
var monthlyCol = ee.ImageCollection.fromImages(monthlyImages_list);
print('monthlyCol', monthlyCol);

print('N images of a month', monthlyCol.filter(ee.Filter.eq('month', 8)));

// We can now compute the monthly average for each month over
// all the years by iterating over the months with map:
// i.e. average July precipitation -> mean precipitation over all
// the July months in the collection (in the N years under
// investigation)
var longTermMeans_list = months.map(function(month) {
    var filtered = monthlyCol.filter(ee.Filter.eq('month', month));
    var monthlyMean = filtered.mean();
    return monthlyMean.set('month', month);
});
print('longTermMeans_list',longTermMeans_list);
// monthly RF_n images
var monthlyNormalRainfallcoll = ee.ImageCollection.fromImages(longTermMeans_list);
print('monthlyNormalRainfallcoll', monthlyNormalRainfallcoll);
```

My target year for the analysis is 2018.

```
// Now we take 2018 data and compute monthly average
// for this specific year
var year_i = '2018';
var filtered_year_i = chirps
  .filter(ee.Filter.date(year_i+'-01-01', year_i+'-12-31'))
// Calculate monthly current average rainfall
// iterating over the months with map
var year_i_monthlyTotals_list = months
  .map(function(month) {
    return filtered_year_i
      .filter(ee.Filter.calendarRange(month, month, 'month'))
      .sum()
      .set('month', month);
});
```

Finally I can compute the deviation:

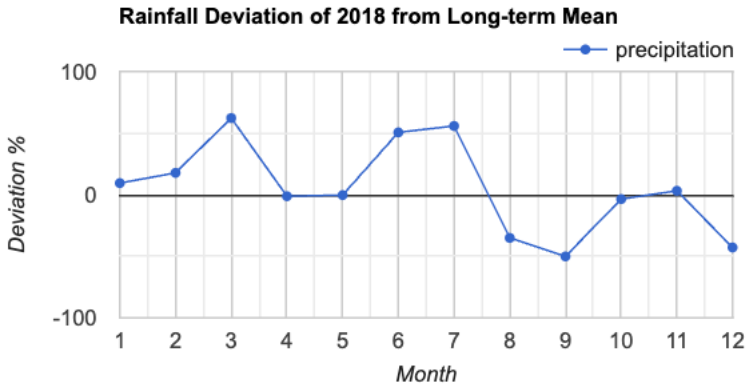
```
// Calculate deviation in %
// RF_dev = 100 * (RF_i - RF_n) / RF_n
var deviation = months.map(function(month) {
  var longTermMean = monthlyNormalRainfallcoll
    .filter(ee.Filter.eq('month', month)).first();
  var monthlyObserved = observedCurrentRainfall
    .filter(ee.Filter.eq('month', month)).first();
  var deviation = (monthlyObserved.subtract(longTermMean)
    .divide(longTermMean)).multiply(100)
    .set('month', month);
  return deviation;
});
```

These are the results we got.

Here the long-term total precipitation for a year in our region of interest:

Here the total precipitation in the chosen year (2018):

Finally let’s see the deviation for each month



The code is here