**Remote sensing of burned areas using Google Earth Engine**

**Laura Turnbull-Lloyd (laura.turnbull@durham.ac.uk)**

**1. Introduction to Google Earth Engine**

Google Earth Engine (GEE) is a powerful web platform for cloud-based processing of remote sensing data on large scales. The advantage lies in its remarkable computation speed, as processing is outsourced to Google servers. The platform provides a variety of constantly updated datasets, so no download of raw imagery is required.

Earth Engine allows users to run algorithms on georeferenced imagery and vectors stored on Google's infrastructure. The Google Earth Engine API provides a library of functions which may be applied to data for display and analysis. Earth Engine's public data catalog contains a large amount of publicly available imagery and vector datasets. Private assets can also be created in users' personal folders.

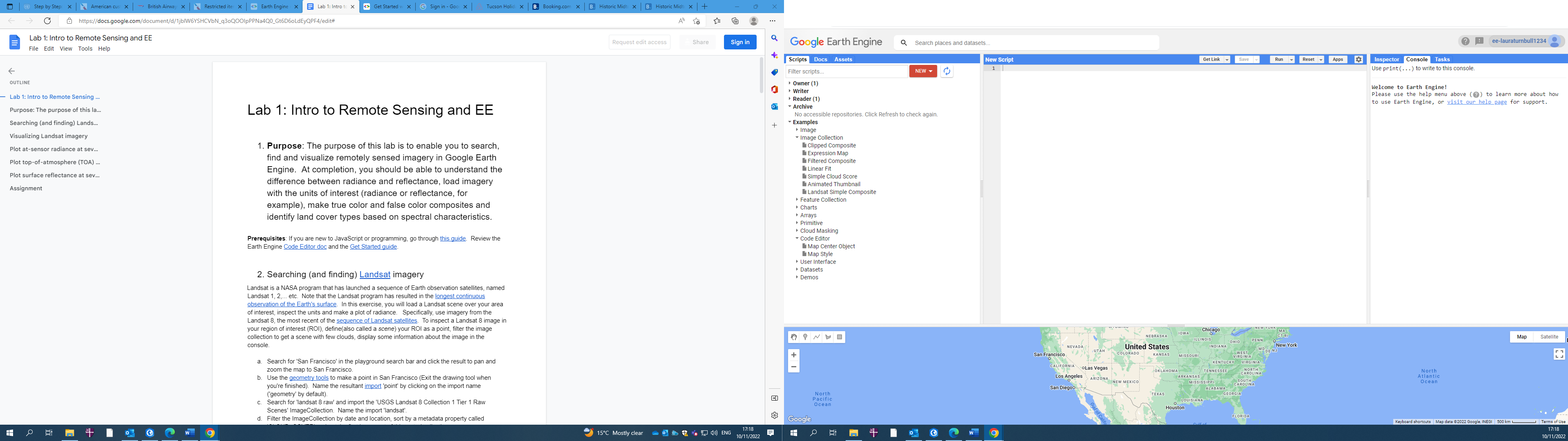
In this session, we’ll only really be able to scratch the surface of the types of hazard-relevant analyses you can undertake within GEE. Therefore, here are some links to other useful resources in case you want to build upon what you will learn in this session:

[Get Started with Earth Engine  |  Google Earth Engine  |  Google Developers](https://developers.google.com/earth-engine/guides/getstarted)

**2. Getting started with Google Earth Engine**

You need to register to get access to GEE, which you can do [here](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwjivri-z637AhVWiFwKHbYoDEUQFnoECAsQAQ&url=https%3A%2F%2Fearthengine.google.com%2Fnew_signup%2F&usg=AOvVaw1aC1AGnYd4g52QUiN6XpSG). To open Google Earth Engine: <https://code.earthengine.google.com/>. You will need to log on.

When you open the code editor, you will see a screen that looks like this:



Code editor

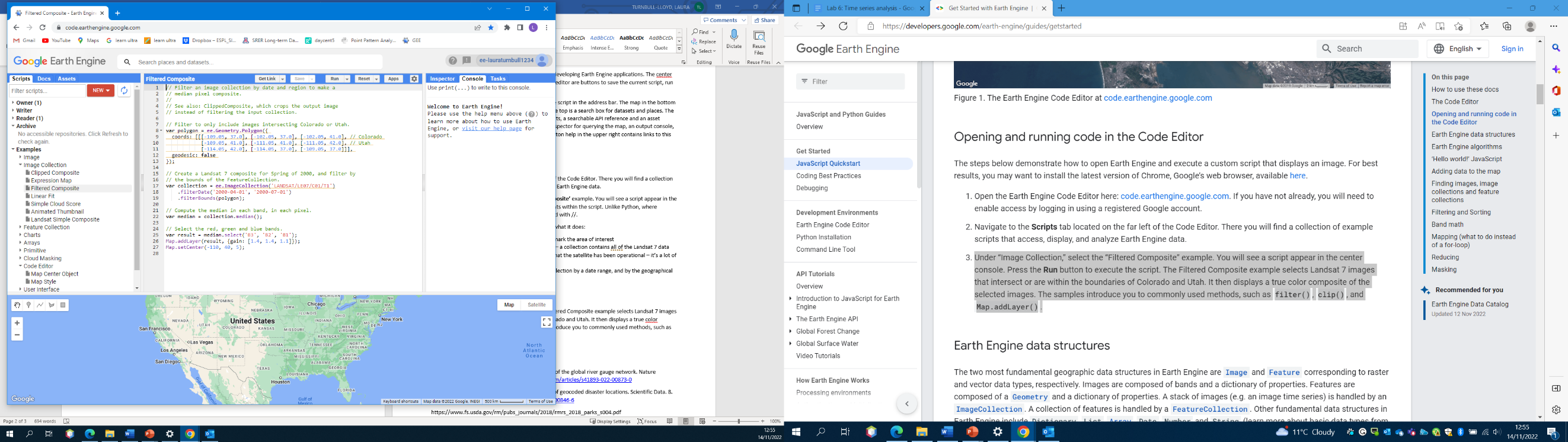
The empty box in the middle is the code editor. This is where you enter code to analyse data. GEE uses Java (a programming language).

**The Code Editor** is an interactive environment for developing Earth Engine applications. The centre panel provides a JavaScript code editor. Above the editor are buttons to save the current script, run it, and clear the map.

The Get Link button generates a unique URL for the script in the address bar. The map in the bottom panel contains the layers added by the script. At the top is a search box for datasets and places. The left panel contains code examples, your saved scripts, a searchable API reference and an asset manager for private data. The right panel has an inspector for querying the map, an output console, and a manager for long-running tasks. The help button help in the upper right contains links to this Guide and other resources for getting help.

**Let’s have a look at a script:** Navigate to the **Scripts** tab located on the far left of the Code Editor. There you will find a collection of example scripts that access, display, and analyse Earth Engine data (**Examples**).

Under ‘**Image Collection’**, select the ‘**Filtered Composite’** example. You will see a script appear in the centre console. You will see that there are comments within the script. Comments in the script are marked with //.

Have a look through the script. Let’s walk through what it does:

1. Creates a polygon using xy coordinates to mark the area of interest.

2. It opens Landsat 7 data (called a collection – a collection contains all of the Landsat 7 data for the entire globe, for the entire period that the satellite has been operational – it’s a lot of data!).

3. It then filters the data stored within the collection by a date range, and by the geographical area contained within the polygon.

4. It then computes the mean value in each band (B3, B2, B1) in each pixel.

5. Then it adds the resulting median data for the red, green and blue bands (i.e., a true colour composite) to the map (more about this below).

Press the **Run** button to run the script. You can see the true colour composites show in the map.

As with all data analysis, it’s good to explore the data you’re working with.

To determine how many images are in the image collection (called collection), we can run (by copying and pasting the below 3 lines into the end of the script):

// See how many images are in the collection

var count = collection.size()

print('Number of images in the collection', count);

Re-run the script. Now, in the console window, you can then see how many images are in the collection.

You can save the script (by clicking on save) in your google drive, if you wish to save your edit.

**A note on band combinations and true-colour composites:** Landsat (and other satellites) measure radiance in multiple spectral bands. A common way to visualize images is to set the red band to display in red, the green band to display in green and the blue band to display in blue. This means trying to match the spectral response of the instrument to the spectral response of the photoreceptors in the human eye. It's not a perfect match, but nevertheless, a visualization done in this manner is called a true-colour image, or a true colour composite.

**3. Data structures in GEE**

The two most fundamental geographic data structures in Earth Engine are **Image** and **Feature** corresponding to raster and vector data types, respectively. Images are composed of bands and a dictionary of properties. Features are composed of a **Geometry** and a dictionary of properties. A stack of images (e.g. an image time series) is handled by an **ImageCollection**. A collection of features is handled by a **FeatureCollection**.

**4. Mapping out the 2013 wildfire near Andratx, Mallorca**

Now we’re going to use a Landsat 8 image collection to map out the 2013 wildfire near Andratx, Mallorca, using the Normalized Burn Ratio. This ratio needs to be computed for imagery before and after a wildfire, and by calculating the difference between these images (the dNBR), we can determine the wildfire burn severity, which shows the spatial impact and severity of the wildfire.

Load the GEE script by clicking on this link:

<https://code.earthengine.google.com/39e1937897c211f87a8d134da38c4ca9>

The code is explained within the script – don’t worry about some of the detail here – the most important thing is to have a general awareness of what the code is doing.

Landsat images have a row and path identifier, which link images to a particular location on Earth’s surface (see <https://landsat.usgs.gov/landsat_acq#convertPathRow>). However, Mallorca is covered by several overlapping Landsat tiles, so it’s easier to define a polygon to pick the area we’re interested in.

**What’s in the script?**

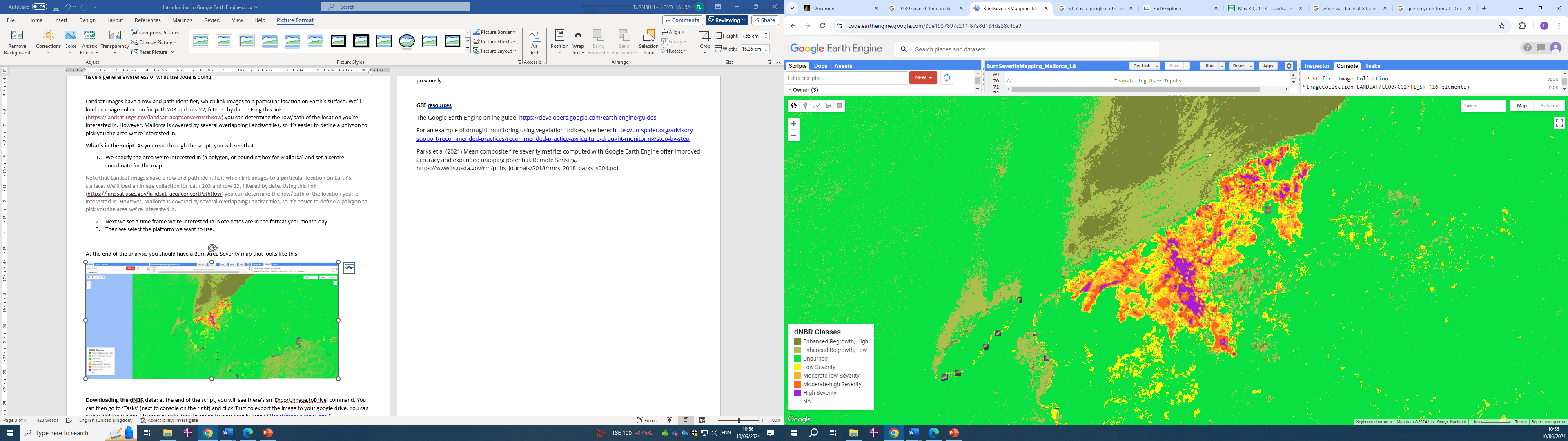
As you read through the script, you will see that:

1. First, we specify the area we’re interested in (a polygon, or bounding box for Mallorca) and set a centre coordinate for the map.

Note that Landsat images have a row and path identifier, which link images to a particular location on Earth’s surface. We’ll load an image collection for path 203 and row 22, filtered by date. Using this link (<https://landsat.usgs.gov/landsat_acq#convertPathRow>) you can determine the row/path of the location you’re interested in. However, Mallorca is covered by several overlapping Landsat tiles, so it’s easier to define a polygon to pick you the area we’re interested in.

1. Next we set a time frame we’re interested in. Note dates are in the format year-month-day.
2. Then we select the platform we want to use – L8.
3. The rest of the script processes the pre and post fire images to apply a cloud filter (we don’t want clouds to feature in the analysis….), output true colour composites, and calculate the Normalized Burn Ratio.
4. The results of running the script (press the run button at the top) are shown in the map, and in the Console window. You can toggle on/off map layers by clicking on the Layer tab in the map window.

At the end of the analysis, you should have a Burn Area Severity map that looks like this:



**Downloading the dNBR data:** at the end of the script, you will see there’s an ‘Export.image.toDrive’ command. You can then go to ‘Tasks’ (next to console on the right) and click ‘Run’ to export the image to your google drive. You can access data you export to your google drive by going to your google drive: <https://drive.google.com/>

**5. Exploring the FireCCI51 product for the same time period**

Here you can open a script to view the Fire CCI51 data product for the same time period.

<https://code.earthengine.google.com/f23fd6894c5a81f633f14a06c04f4fc6>

**6. Summary**

You should now have an awareness of the power and usefulness of analysing remote sensing imagery through GEE. The scripts you have looked at today, and all the other example scripts available in GEE and in the Google Earth Engine online guide: <https://developers.google.com/earth-engine/guides> can be customized, by adjusting the time window of analyses, or the spatial location. Outputs from GEE can subsequently be analysed in other spatial data analysis software such as R, QGIS, ArcPro etc.

**Other GEE resources**

The Google Earth Engine online guide: <https://developers.google.com/earth-engine/guides>

For an example of drought monitoring using vegetation indices, see here: <https://un-spider.org/advisory-support/recommended-practices/recommended-practice-agriculture-drought-monitoring/step-by-step>

Lizundia-Loiola, J., et al. (2020) A spatio-temporal active-fire clustering approach for global burned area mapping at 250 m from MODIS data. https://doi.org/10.1016/j.rse.2019.111493

Parks et al (2021) Mean composite fire severity metrics computed with Google Earth Engine offer improved accuracy and expanded mapping potential. Remote Sensing. https://www.fs.usda.gov/rm/pubs\_journals/2018/rmrs\_2018\_parks\_s004.pdf