Introduction to Google Earth Engine

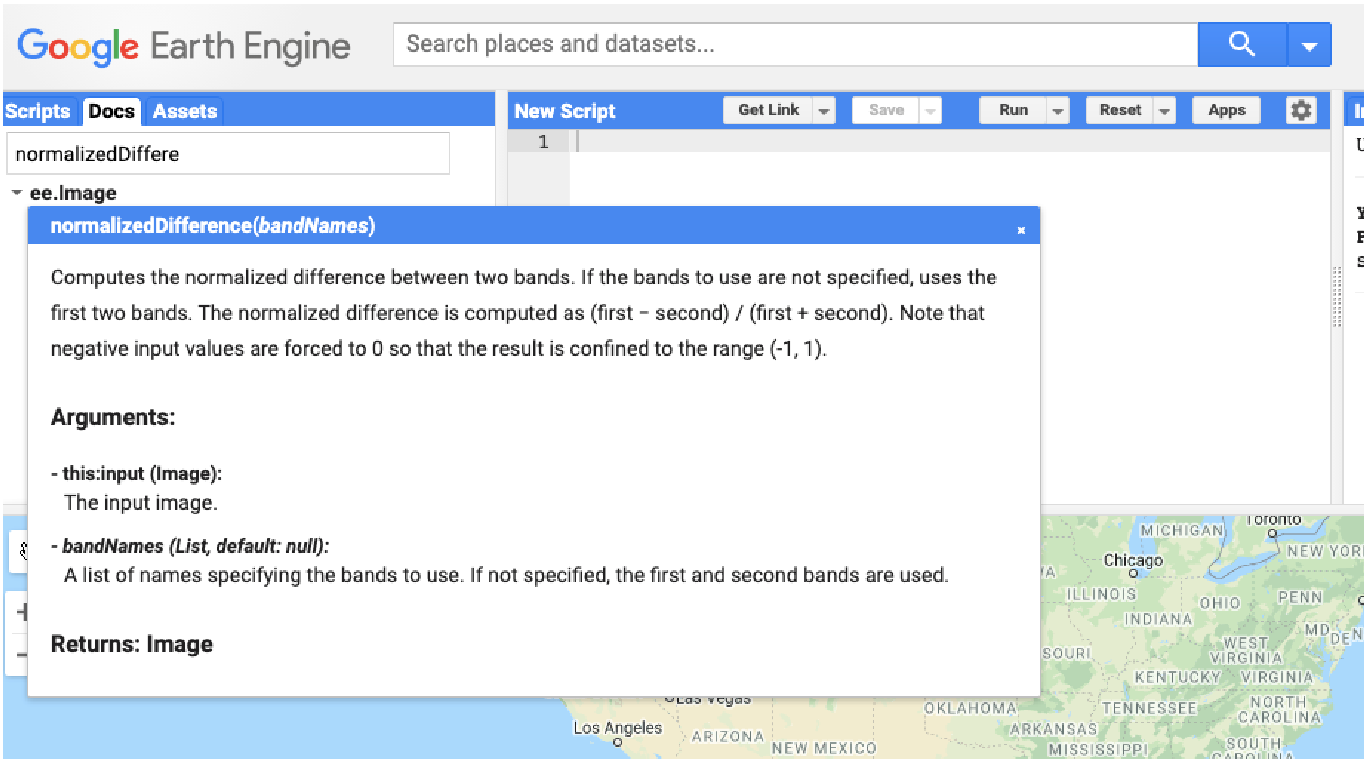
Session 2 - Image functions

> Objectives
>
> - Understand the concept of image functions and where to search for them.
> - Learn how to derive normalised indices in GEE using normalised burn ratio as an example.
> - Learn how to perform a difference between two images.
> - Learn how to apply to colour palette to an image.
> - Learn how we can export data from GEE to use in other GIS/remote sensing software.

One of the benefits of Earth Engine is the ability to process EO data quickly and efficiently. There are lots of functions (similar to tools in ArcMap), built into GEE, that can be used to process an image. We can see all the functions that are available using by searching the Docs.

> **Step 1:** Navigate to the docs tab and the ee.Image dropdown. Take a minute to look at the functions that can be a applied to an image.

We can use the docs tab to search for a function to apply to an image. By clicking on a given function, we can see a description about the function, the arguments it requires and what is returned as output. Check out the info for the normalizedDifferrence function below.



normalizedDifference(*bandNames*)

Computes the normalized difference between two bands. If the bands to use are not specified, uses the first two bands. The normalized difference is computed as (first − second) / (first + second). Note that negative input values are forced to 0 so that the result is confined to the range (-1, 1).

Arguments:

- this:input (Image):
The input image.

- bandNames (List, default: null):
A list of names specifying the bands to use. If not specified, the first and second bands are used.

Returns: Image

The normalizedDifference function requires two inputs. The image to calculate the the normalized difference on and the names of the bands to use. The normalized difference is the difference between reflected energy for two different image bands divided by their sum.

> normalizedDifference = band_1 - band_2 / band_1 + band_2

There are several useful normalized difference indices, the most common is NDVI (normalized difference vegetation index) which is useful for identifying healthy vegetation. In this example we will look at normalized burn ratio (NBR) which is used to detect burnt areas using near-infrared (NIR) and shortwave-infrared (SWIR) energy. The NIR and SWIR bands for Sentinel-2 are band 8 and 11 respectively. Check this out for more info on NBR

> NBR = NIR - SWIR/NIR + SWIR

> **Step 2:** Use the normalized difference function to calculate NBR for image_1. Type your code in below `//// START NEW CODE BELOW ////`

> Click here to access the script

```
// calculate NBR for image_1
var NBR_image_1 = image_1.normalizedDifference(['B8', 'B11'])

// print NBR_image_1 and add to map view
print(NBR_image_1)
Map.addLayer(NBR_image_1, {min: -1, max: 1}, 'NBR_image_1')
```

> **Top-tip:** Typing your code rather than using copy & paste will help with your understanding of what each line is doing.

The output from the function is a single band raster image containing the NBR values. The output defaults to a greyscale as no colour ramp/palette has been defined. A colour palette is defined in a similar way to defining the band combination for a multi-band raster image.

`palette:['red', 'white', 'green']`

> **Step 3:** create a visParams variable defining a colour ramp for NBR_image_1 under `// create visParams variable which includes a colour ramp` and add the layer to map.

```
// create visParams variable which includes a colour ramp
var visParams = {min: -1, max: 1, palette:['red', 'white', 'green']}

// add NBR_image_1 to map view
Map.addLayer(NBR_image_1, visParams, 'NRB_image_1_colourRamp')
```

The first colour specified in the ramp is applied to the minimum values and the last colour the maximum values. In this example, standard colours have been used where the colour can be specified by name. What if different shades were required, or more colours?

You can use colour hex codes in GEE and the palette argument will accept a list with more colours if required. Check out this for more info.

You've successfully calculated NBR for a Sentinel-2 image. Well done! But what if we had multiple images from different dates to monitor burnt areas over time? You could copy the code that you've already written but this might be suitable for lots of images.

> **Step 4:** Create a function that takes an image as an argument and returns NBR for the input image. Do this **above** line *n* `// calculate NBR for image_1`

> **Top-tip:** Remember to include an appropriate comment so you remember what the following line of code is doing.

```
// create function to calculate NBR
var calcNBR = function(image) {
    return image.normalizedDifference(['B8', 'B11'])
};
```

Now we have a function that can be used to calculate NBR for any image that is supplied as input. Lets put this to use!

> **Step 5:** Calculate NBR for image_1 and image_2 using `calcNBR()` and add them both to the map view using VisParams. Do this below `calcNBR` and `visParams`

```
// Create function to calculate NBR for any image
var calcNBR = function(image) {
  return image.normalizedDifference(['B8', 'B11']);
};

// create visParams variable which includes a colour ramp
var visParams = {min: -1, max: 1, palette:['red', 'white', 'green']};

// calculate NBR for image_1 & image_2
var NBR_image_1 = calcNBR(image_1);
var NBR_image_2 = calcNBR(image_2);

// add NBR images to the map view
Map.addLayer(NBR_image_1, visParams, 'NBR pre-fire');
Map.addLayer(NBR_image_2, visParams, 'NBR post-fire');
```

Now we can calculate NBR for any image efficiently using `calcNBR`. We'll learn about one more really useful image function in GEE that we can use to estimate burn severity. This is called dNBR and is the difference between NBR pre & NBR post a fire event. This example is from Kangaroo Island, Australia which experienced significant bush fires in decemeber 2019. Image_1 was acquired before the fires and a image_2 after. We can do this using the `subtract()` function.

> dNBR = preNBR - postNBR

> **Step 6:** Calculate dNBR using the subtract function and add to the map view with appropriate visParams.

```
// calculate dNBR using the subtract function
var dNBr = NBR_image_1.subtract(NBR_image_2)

// define visParams for dNBR
var dNBR_visParms = {min:-1, max:1, palette:
['#ffffb2','#fed976','#feb24c','#fd8d3c','#fc4e2a','#e31a1c','#b10026']}

// add dNBR to the map view using visParams
Map.addLayer(dNBr, dNBR_visParms, 'dNBR');
```

It is clear that this area has undergone a severe burn based on the output the dNBR output. Performing a difference in GEE is done with the `subtract()` function where the image we want to takeaway from is specified first, and the image we want to takeaway is in the brackets. Can you think of any other geospatial applications where differencing would be useful?

This exercise should hopefully really emphasize the benefits of GEE. You can quickly access, process and derive outputs. To calculate dNBR for Kangaroo Island using a local computing platform would require downloading and proprocessing which could take longer than the analysis!

Now these outputs have been derived the chances we want to make a spatially referenced map using traditional software like ArcMap or QGIS. Do to so we need to export the data from GEE.

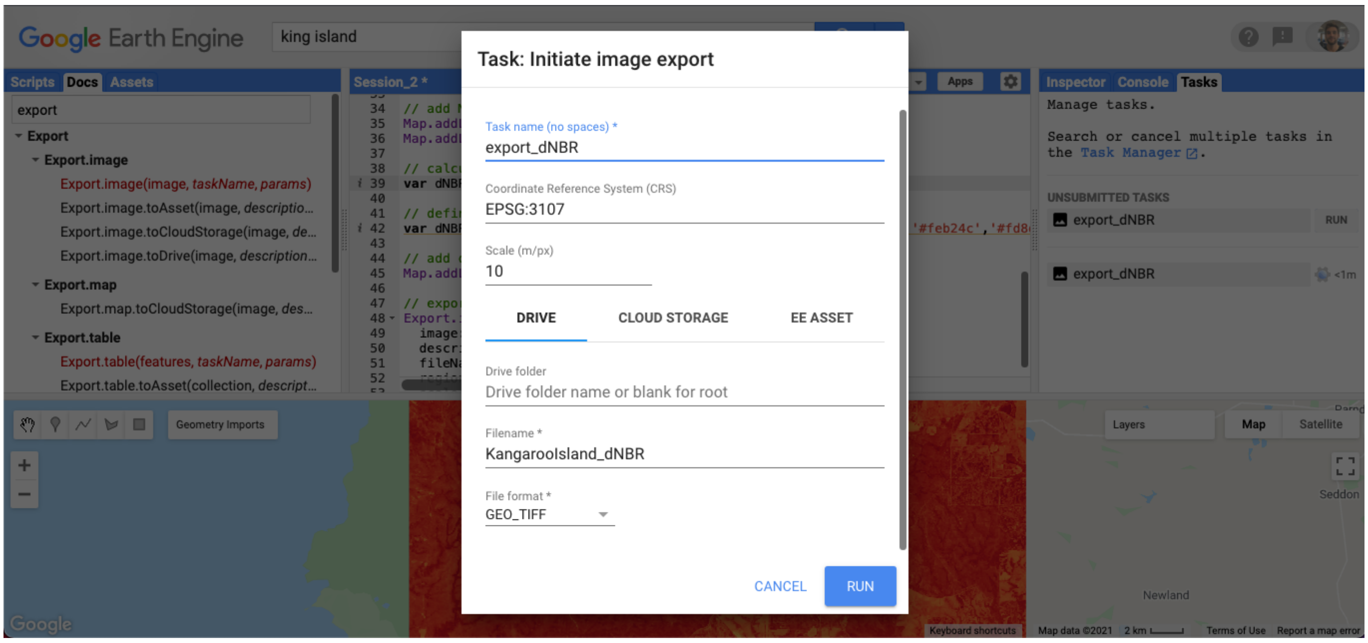
We do this using `Export.image`. We can export an image to three locations:

1. `toAsset` - storage on you GEE account (we'll learn more about assets in the next session)
2. to Google cloud storage
3. to Google drive

In this example you will export it to your Google drive.

> **Step 7:** Export the dNBR output to Google drive using `export.image.toDrive()`. Navigate to the task manager after running the script and click RUN.

```
// export dNBR layer to Google drive
Export.image.toDrive({
 image: dNBR,
 description: 'export_dNBR',
 fileNamePrefix: 'KangarooIsland_dNBR',
 region: geometry,
 scale: 10,
 crs: 'EPSG:3107'
});
```



Task: Initiate image export

Task name (no spaces) *
export_dNBR

Coordinate Reference System (CRS)
EPSG:3107

Scale (m/px)
10

DRIVE CLOUD STORAGE EE ASSET

Drive folder
Drive folder name or blank for root

Filename *
KangarooIsland_dNBR

File format *
GEO_TIFF

Manage tasks.

Search or cancel multiple tasks in the Task Manager.

UNSUBMITTED TASKS
export_dNBR RUN

export_dNBR <1m

You'll notice that the options we've specified in the script can be edited before we run the export task. The task name allows us to track the export in the task manager.

We need to assign a coordinate system for the image. If we leave this blank it will default to the CRS used in the map view. The scale is the spatial resolution/pixel size (in this case the same as Sentinel-2).

If we want to save it to a particular folder in Google drive we can specify the folder as well as the file name. The default format for raster data is a geoTiff. We can also change the destination before running the export to either cloud storage or an asset.

There are other optional parameters that can be specified. You can view these in the docs tab for the export image functions.

Exporting outputs from GEE is very useful. This can be outputs that have been derived from processing using Earth Engine or even preprocessed EO data to then be used in other software or workflows. Exporting data as an asset makes it easier to access if it is needed multiple times. We'll cover this is the next session.

Key points

- There are many functions/tools that be applied to data in GEE, we can search for these using the docs tab.
- We can wrap these functions into variables to apply these functions to multiple data/imagery efficiently.
- A colour palette is applied to a single band raster by specifying the colours in the palette via their hex code.
- Differencing is performed with the `subtract()` function. The expression tool can be used to apply more complex calculations
- `Export.image` can be used to export images from GEE to your Google drive or a Google cloud bucket via the task manager or to be stored in your GEE account as an asset.