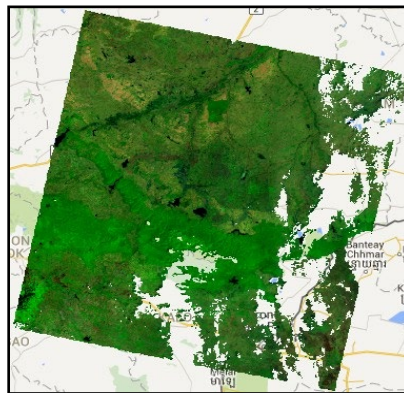


EXERCISE 1

Creating a Cloud-free Landsat Composite in Google Earth Engine



Introduction

The Earth Engine Playground comes fully loaded with a diverse collection of tools for processing and analyzing spatial data. These tools are accessible as JavaScript functions and methods. In addition to GEE functions specific for the analysis of spatial data, most standard JavaScript functions are also included. The availability of generic and specific functions allows users to create customized analysis routines and develop highly sophisticated customized scripts. In this exercise, we will learn about a routine developed for creating and downloading cloud-free Landsat mosaics.

The following exercise will walk you through the process of creating a cloud free composite that utilizes imagery from three Landsat sensors (5, 7, and 8) and uses a more sophisticated cloud masking algorithm. We won't explore this added complexity in depth during this exercises; however, this means that this script will be well-suited for application to subsequent workflows for digital soil mapping, or other applications.

Objectives

- Learn how to import shapefiles as assets in GEE
- Investigate cloud-free composite script
- Run script and adjust parameters

Required Data:

- **VT_boundary.shp** – shapefile representing example area of interest. Download from the [Course Downloads](#) folder.

Prerequisites

- **Google Chrome installed on your machine**
- **An approved Google Earth Engine account**
- **Follow the links below to gain read access to the GEE code repositories we will refer to in the script.**
 - [Click here to gain access to the GTAC module repository](#)
 - [Click here to gain access to the GTAC training repository](#)



Table of Contents

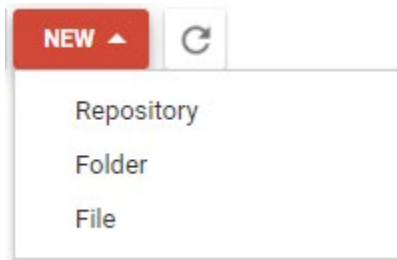
Part 1: Set up your GEE environment.....	4
Part 2: Import shapefile as asset	5
Part 3: Investigate the Cloud Free Composite script.....	10
Part 4: Prepare a cloud free composite for a study area.....	11
Part 5: Assess composite, adjust parameters.....	14
Part 6: Review user editable variables	15

Part 1: Set up your GEE environment

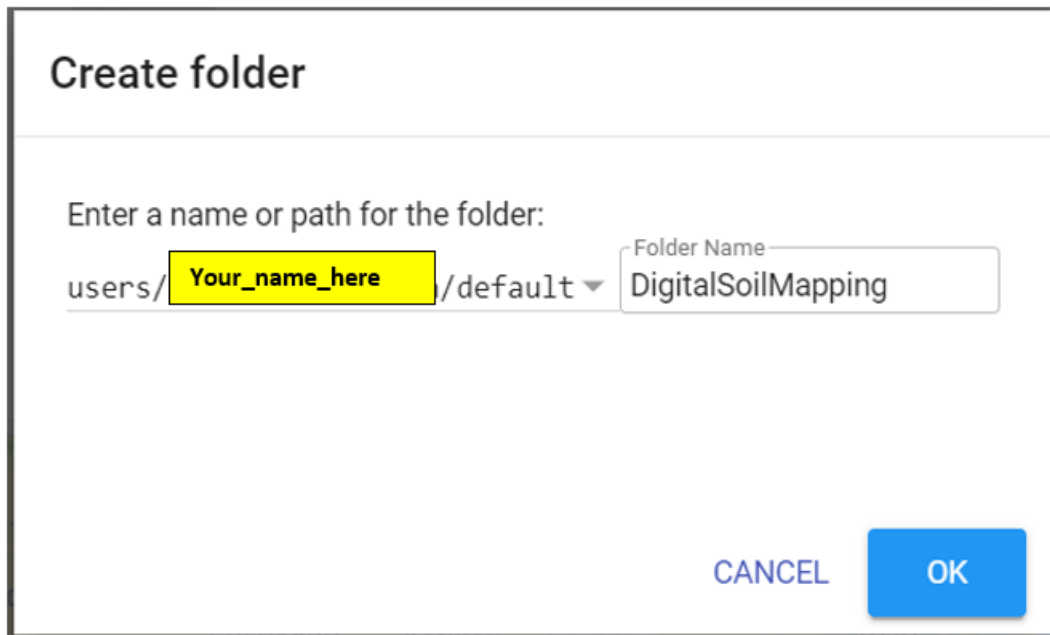
In this section, you will use basic Earth Engine methods to create a collection of Landsat images and use basic cloud masking and compositing methods. If this material is all new to you, you might want to refer to the self-paced tutorial for [Geospatial Scripting in JavaScript and Earth Engine](#).

A. Create a new folder in the code editor

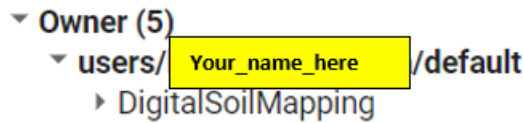
1. In your browser, navigate to <https://code.earthengine.google.com/> and login to your Google account if prompted
2. Create a new folder in your default repository
 - i. Under the Scripts tab in the upper left window, click the New button and select File to create a folder this course and script for this exercise.



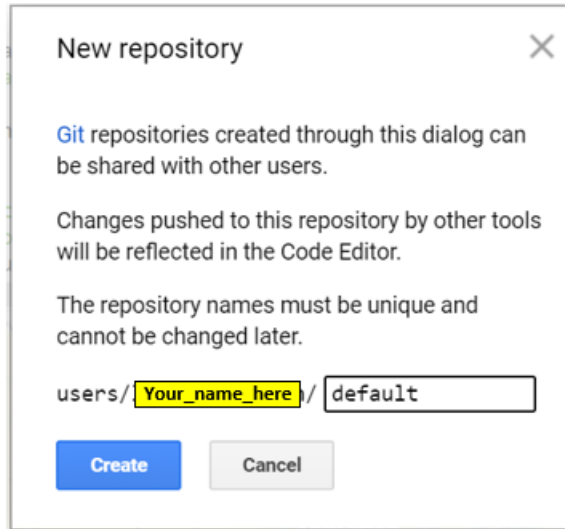
3. In the dialog box, enter the path 'DigitalSoilMapping' and click OK – This creates a folder called DigitalSoilMapping.



- i. After the folder has been created, it will show up in your Scripts pane, under the Owner section.



- ii. This folder will be where you can write and save scripts throughout the course of this exercise.
- iii. If you don't have a "default" repository available, you can create one before completing Step 3. Click New > Repository and enter "default" in the dialog box that appears.



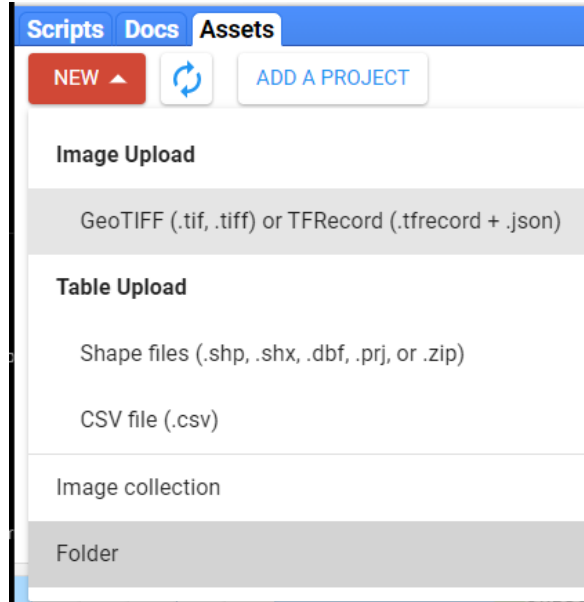
Note: The Earth Engine code editor stores scripts in [Git](#) repositories, or repos, hosted by Google. In your script manager, repositories are arranged by access level and your private scripts are stored in the Owner folder (**users/username/default**). You can create additional repositories, share them and use the extremely popular distributed version control system, [Git](#), to manage your repos or even sync them with external systems like [GitHub](#).

Part 2: Import shapefile as asset

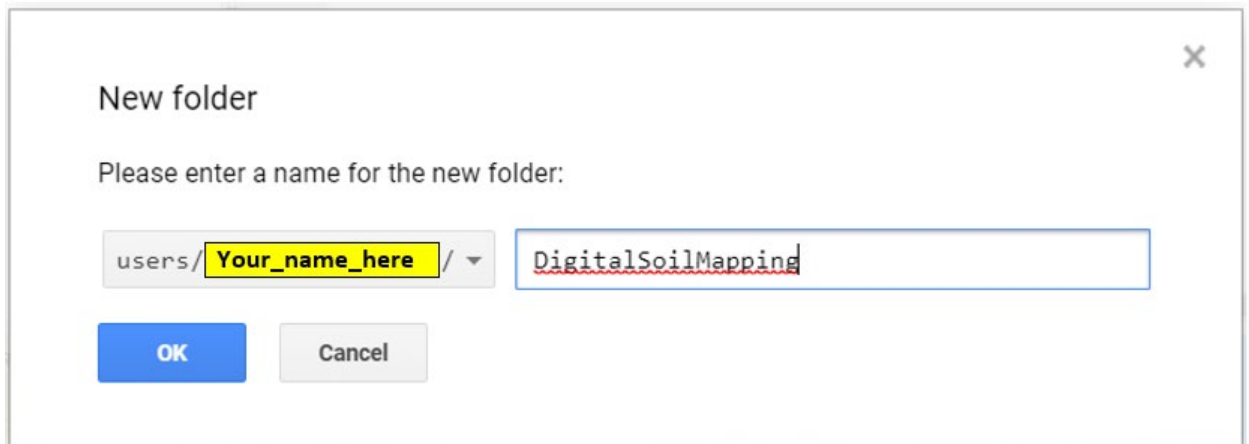
Next, we will learn how to import shapefiles into GEE. While we can draw polygons on the map to specify an area of interest for a script (more on that later), importing shapefiles into GEE allows us to work with a pre-defined area of interest.

A. Create a new assets folder

1. On the lefthand pane, click on the Assets tab. Then, click the New button, and click Folder.



2. In the window that appears, write “DigitalSoilMapping” in order to create a new Assets folder, and Click OK.
 - i. Distinct from the scripts folder, the assets folder is designed to hold shapefiles, tabular data, and other inputs.



3. The folder will appear in the Assets pane, and should be empty. Now, we can upload a shapefile representing our area of interest to this folder.

B. Download course data

1. Start by visiting the [Course Downloads](#) folder. Click on the 02/Data folder to open it. You will see a list of folders, and a .zip file titled “DSM_EarthEngine_CourseData.zip.” Download the .zip to your computer and unzip it.
 - i. The file structure of the folders in the Data folder online is the same as the file structure in the .zip folder. If you encounter any difficulties with the .zip file, you can navigate to the appropriate files in the online interface and download them one at a time.
2. The data folder contains three sub-folders representing three distinct areas of interest. To start, we’ll be working with the Essex_VT area.

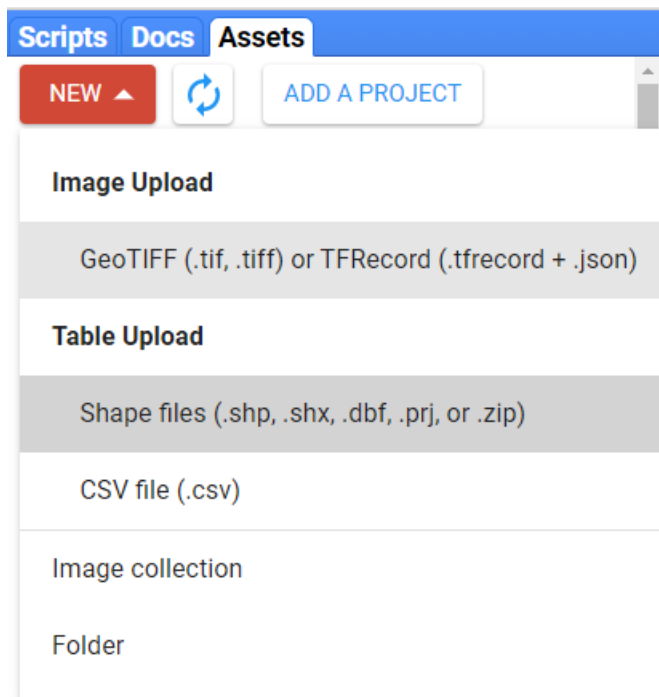
- Essex_VT
- GNP_MT
- RyanFlat_TX

3. Open the folder and see that there are two folders inside. The VT_boundary folder contains a shapefile representing a polygon of the boundaries for a region of interest (ROI) that we'll use for our soil mapping project. The VT_pedons folder contains a shapefile representing point data containing ~100 soil sample points with predictor data.

- i. We'll use the pedon data later when we begin our classification. For now, we're only interested in the boundary shapefile.

C. Load shapefile into GEE

1. Go back to your EE window and interface. On the Assets tab, click New, then click Shape files under Table Upload.



2. In the window that appears, click Select, and navigate to the folder that contains the VT_boundary shapefile. Use CTRL+click to select the files ending in .dbf, .prj, .shp, and .shx, and click Open.

Name	Date modified	Type	Size
VT_boundary.CPG	3/12/2021 12:23 PM	CPG File	1 KB
VT_boundary.dbf	3/12/2021 12:23 PM	DBF File	1 KB
VT_boundary.prj	3/12/2021 12:19 PM	PRJ File	1 KB
VT_boundary.sbn	3/12/2021 12:23 PM	SBN File	1 KB
VT_boundary.sbx	3/12/2021 12:23 PM	SBX File	1 KB
VT_boundary.shp	3/12/2021 12:23 PM	SHP File	139 KB
VT_boundary.shp	3/12/2021 12:41 PM	XML Document	7 KB
VT_boundary.shx	3/12/2021 12:23 PM	SHX File	1 KB

3. In the Upload a new shapefile asset window, the file name will have auto-populated from the input files. Add the name of the folder you just created, followed by a slash, before the file name, in the field under the Asset ID heading. Then click Upload.
 - i. The Asset Name field should read "DigitalSoilMapping/VT_boundary". This creates the object VT_boundary inside the DigitalSoilMapping folder.

Upload a new shapefile asset

Source files

SELECT

Please drag and drop or select files for this asset.
Allowed extensions: shp, zip, dbf, prj, shx, cpq, fix, qix, sbn or shp.xml.

VT_boundary.dbf	
VT_boundary.prj	
VT_boundary.shp	
VT_boundary.shx	

Asset ID

users/ Your_name_here /

Properties

Metadata properties about the asset which can be edited during asset upload and after ingestion. The "system:time_start" property is used as the primary date of the asset.

Add start time
Add end time
Add property

Advanced options

Character encoding
 🔍 ?

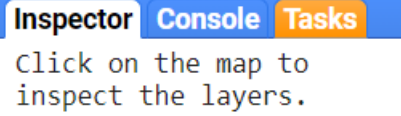
Maximum error
 ?

Split large geometries ?

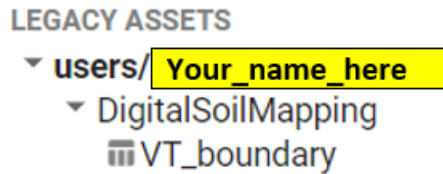
[Learn more](#) about how uploaded files are processed.

CANCEL
UPLOAD


4. You will see the Tasks tab in the righthand pane turn orange. Click on it to see that EE is starting to ingest, or upload, your asset. This can take anywhere from 30 seconds to a few minutes.

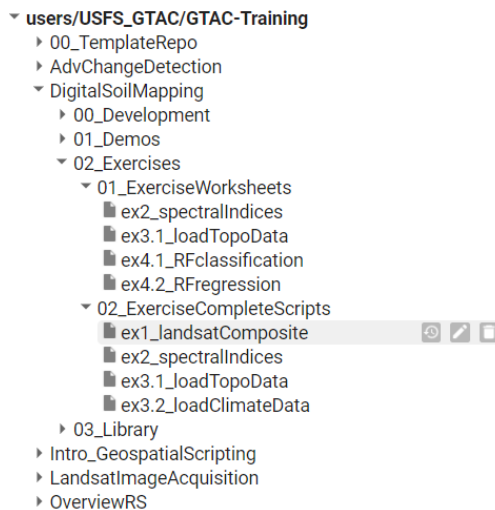


- i. After the bar turns blue, you will see the shapefile appear in the folder in the lefthand pane.



D. Access the GTAC training code repository

1. One of the benefits of Google Earth Engine is that we can easily create and share repositories and scripts with collaborators, or end users who can use the tools we develop. That’s what we’ve done with the scripts for our GEE training courses.
2. [Follow this link to add the GTAC Training GEE code repository to your Earth Engine account.](#)
 - i. The repository “users/USFS_GTAC/GTAC-Training” will appear under the “Reader” tab in the scripts pane. If it doesn’t appear immediately, click the refresh  button at the top of the scripts pane.
3. Open the GTAC-Training repository, then open the DigitalSoilMapping folder and the 02_Exercises folder.
 - i. Note that there are folders for several other GTAC courses on Google Earth Engine. You can refer to these scripts at any time during this workshop or in the future.
 - ii. Within the DigitalSoilMapping/02_Exercises folder, there are sub-folders for ExerciseWorksheets and ExerciseCompleteScripts. The “worksheets” contain outlines of code that you will fill in, as you follow the exercises. The “complete scripts” show what the script looks like after the exercise has been completed, and will run a complete script.



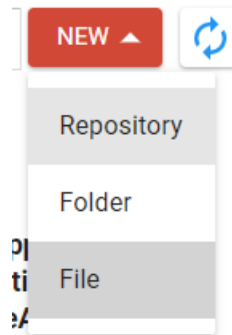
- i. You can’t edit these scripts in this repository—you have been granted “Read-only” access! But, you can save them to the folder you created in Part 1 to complete the exercises.

- ii. Note that there is no worksheet for this exercise! Instead, we'll run the completed script, and walk through the different parameters.

Part 3: Investigate the Cloud Free Composite script

A. Open the Script link

1. Open the appropriate script in the GTAC-Training repository.
 - i. Open the script located at [USFS GTAC/GTAC-Training/DigitalSoilMapping/02 Exercises/02 ExerciseCompleteScripts/ex1_landsatComposite](#)
2. Copy the code to a new script in the folder you created in Part 1.
 - i. Use CTRL+ A to select the entire script.
 - ii. Use CTRL + C to copy the entire script.
 - iii. Click the red NEW button at the top of the lefthand pane and scroll down to select "File" to create a new file.



- iv. In the window that appears, make sure your default repository is selected, and copy in "DigitalSoilMapping/ex1_landsatComposite." Click OK. This creates a new, blank script called "ex1_landsatComposite" in the DigitalSoilMapping folder.

Create file

Enter a name or path for the file:

users/**Your_name_here**/default

Enter description (optional):

CANCEL

- v. Click the new script you just created to open it in the Code Editor.
- vi. With your cursor in the Code Editor, use CTRL+V to paste the script you copied into the new script.
- vii. Use CTRL + S or click Save at the top of the window to save the script to your personal repository.
- viii. Now the script will be editable for you!

B. Expand the code and review the script

1. We will be running the script more or less as-is. This script is also provided for you to use to create Landsat image composites in future projects, as well. But, the goal is for you to understand how to run the script for an area of interest.
2. Take some time to explore the script.
 - i. Lines 1-16 provide documentation about the authors, audience, and purpose of the script. Through the script, any lines (or portions of lines!) colored green, and preceded by a double slash // are comments. That means we can add annotations to our code, without running them as lines of code.
 - ii. Lines 18-44 provide User Editable Variables. This script has been structured so that all of the inputs that a user can change are at the very beginning, to facilitate ease of use by varied audiences. This section is divided into four categories: (1) composite area; (2) composite time period; and (3) composite parameters.

C. Explore the following:

1. **Composite area** - this section allows the user to specify the area for which composites are made.
2. **Composite time period** - this section allows the user to specify the range of years for which a composite is made
3. **Composite parameters** - this section allows the user to modify various parameters to customize the composites

Part 4: Prepare a cloud free composite for a study area

In this section we will create a cloud-free composite using the shapefile we uploaded in Part 2. However, you could use these same instructions for any study area that you are interested in.

A. Import the shapefile into the script

1. Update the file path in Line 23 to refer to the shapefile you uploaded.
 - i. This line located the shapefile in your assets folder, and names it "VT_boundary" so that you can access it in the script.
 - ii. Edit the path so it refers to your username. If you named your folders and file the same way we did in the exercises, you should be able to replace the username "lleatherman_usda" with your GEE username.

```
// Import shapefile from assets folder
```

```
var VT_boundary =
ee.FeatureCollection('users/YOUR_USERNAME/DigitalSoilMapping/VT_boundary');
```

iii. You can also get the path to the file by locating the file in your assets folder, clicking the name of the file to open a summary pane, clicking the “Copy” icon next to “Table ID”, and pasting the file path into your script.

Table: VT_boundary

DESCRIPTION FEATURES PROPERTIES Edit

No description.

Table ID

users/**Your_name_here**/DigitalSoilMapping/VT_boundary

Date
Start date: NA
End date: NA

File Size 69.86KB

IMPORT DELETE SHARE CLOSE

2. Convert the shapefile to an import record. This stores the shapefile as an object in your script.

i. You will see a yellow line below Line 23. Hover over Line 23 in the script, and a window will pop up telling you that “VT_boundary” can be converted to an import record.’

```
22 // Import shapefile from assets folder
23 var VT_boundary = ee.FeatureCollection('users/1leatherman_usda/DigitalSoilMapping/VT_boundary');
24
25 //Composite area of
26 var compositeArea
```

"VT_boundary" can be converted to an import record. [Convert](#) [Ignore](#)


3. Click convert. You will see that Line 23 is now absent from your code, and VT_boundary has appeared in a new box called Imports at the top of the Code Editor.

DigitalSoilMapping/ex1_landsatComposite [Get Link](#)

Imports (1 entry)

- var VT_boundary: Table users/1leatherman_usda/DigitalSoilMapping/VT_boundary

Note: It’s not necessary to convert the shapefile to an import record to use it in the script. But, the imports pane provides a convenient place to view objects you’ve imported into the script, and can facilitate sharing files among scripts.

To import the shapefile into subsequent scripts, you can copy the code snippet for the shapefile using the code snippet button  in the Imports window into a new script. Or, you can copy Lines 22-23 directly, retaining the comment in future scripts.

For future exercises, you can refer to this documentation to ensure that the VT_boundary file (or other input files!) are available for your script.

B. Run the script with a user selected ROI

1. Set up the user editable variables according to the code below.

```

////////////////////////////////////
// User Editable Variables
////////////////////////////////////

// Import shapefile from assets folder
var VT_boundary =
ee.FeatureCollection('users/1leatherman_usda/DigitalSoilMapping/VT_boundary');

//Composite area of interest
var compositeArea = VT_boundary.geometry().bounds();
//var compositeArea = geometry; // use a polygon drawn on the map rather than an
uploaded shapefile
var roiName = 'Essex_VT'; // Give the study area a descriptive name.
var buffer_distance = 1; // Distance to buffer composite area in meters. Must be
> 0.

//Composite time period
var year = 2019; // Start year for composite
var startJulian = 152; // Starting Julian Date
var endJulian = 252; // Ending Julian date
var compositingPeriod = 0; // Number of years into the future to include

//Composite parameters
var cloudThresh = 20; // Specify cloud threshold (0-100)- lower number masks out
more clouds
var possibleSensors = ['L5','L7','L8']; // Specify which sensors to pull from-
supports L5,L7, and L8

```

```
var reducerPercentile = 50; // Reducer for compositing

//Export parameters
var exportToDrive = 'no'; // gives the option to export image composite to your
google drive; Default is 'no'
var crs = 'EPSG:32618'; // EPSG number for output projection. 32618 = WGS84/UTM
Zone 18N. For more info- http://spatialreference.org/ref/epsg/

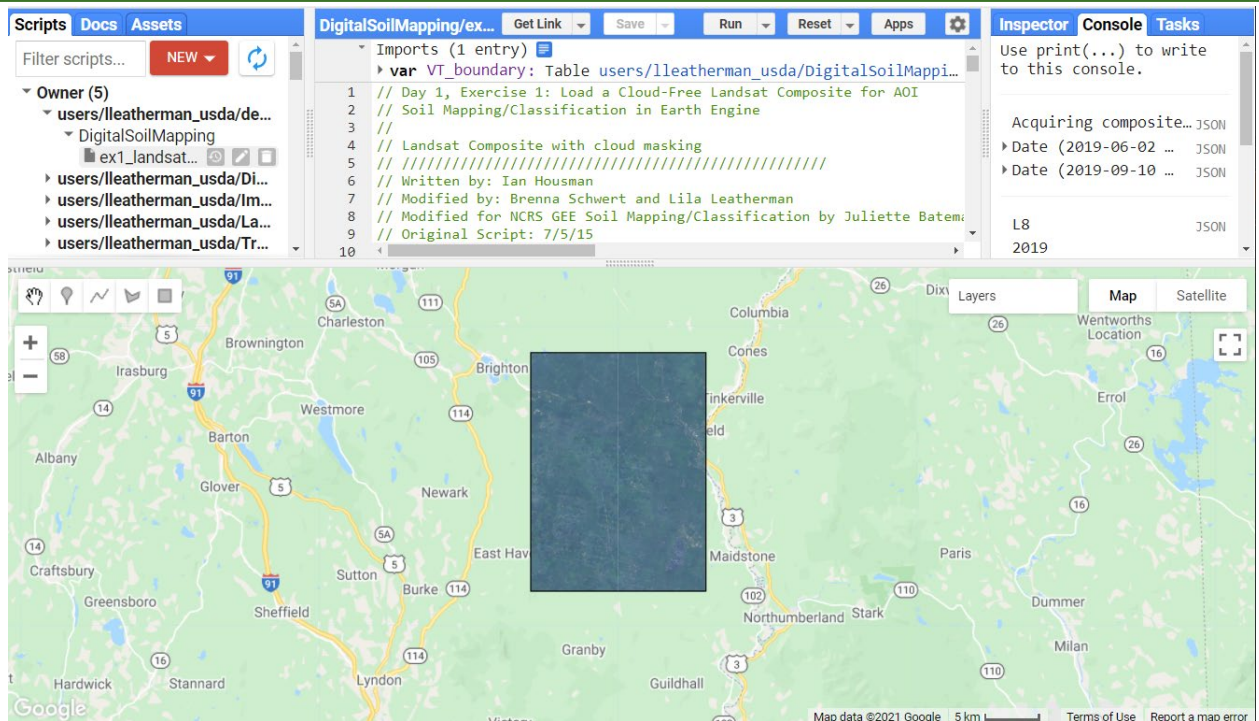
//!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
//!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
// End of user editable parameters. Do not edit anything below this line. Unless
you are brave. ;)
//!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
//!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

2. Run the script and review the results.
 - i. Click the **Run** button.

Part 5: Assess composite, adjust parameters

A. Examine imagery

1. Zoom to the composite area in the Viewer.



2. Use the **Viewer** tools to explore the composite created by the script.

i. Use the Layer buttons to toggle off and on the composite.

(a) Note the name of the composite in the **Layers** menu.

1. The composite nomenclature consists of the roiName, years included in the composite, and the Julian start and end dates.

ii. Is there complete coverage for the composite? Are there any clouds that could not be masked?

B. Adjust cloud and shadow thresholds and compare changes

1. Review the cloud threshold adjustment.

i. Adjust the **cloudThresh** variable to obtain an optimum level of cloud masking. Remember, the lower the threshold, the less clouds there will be.

ii. You will probably not be able to eliminate all clouds. Some areas have such persistent cloud cover that there may not be any clear imagery in your composite time period.

C. Examine pixel values

1. Review the Appendix to learn about using the **Inspector** to examine pixel values

2. Use the **Inspector** to examine pixel values in the composites

i. Throughout this course, we will be adding additional useful bands to our composite image. As we move forward, continue to examine the pixel values -- they can be used to evaluate data quality and assist in the classification process.

(a) Input screen shot of inspector here

Part 6: Review user editable variables

Now that you have had a chance to run the script, let's review the input parameters.

A. Composite Area – controls for the spatial extent

1. **compositeArea**: allows the user to specify the region of interest. This defaults to the pre-specified geometry, but it could be changed to either another geometry or an asset.
2. **roiName**: specifies the region of interest for labeling in the viewer
3. **buffer_distance**: specify the buffer (in meters) beyond your compositeArea. Must be greater than zero.

B. Composite time period – controls temporal extent

1. **year**: specify the start year of your composite
2. **startJulian**: Julian day for starting composite.
3. **endJulian**: Julian day of year to end composite. The startJulian and endJulian dates allow you to specify a subset of each year to include in the composite. For example if your startJulian day = 305 (November 1) and your endJulian day = 90 (March 31) your composite will only include imagery collected from November 1 to March 31. This allows you to constrain the composite to a particular season (e.g., winter) for a more uniform image.
4. **compositingPeriod**: the number of years into the future the composite will include. For example if your year variable is 2010 and compositingPeriod is set to 5, the script will produce a composite using imagery from 2010-2015. If the compositingPeriod were set to 0, only imagery from 2010 would be included.

C. Composite parameters

1. **cloudThresh**: controls the sensitivity of the cloud masking algorithm. Lower numbers increase masking, higher numbers decrease masking.
2. **possibleSensors**: controls which sensors are included in the composite. Only Landsat 5, 7, and 8 are supported by the script.
3. **reducerPercentile**: controls the percentile of the composite. For example if the reducerPercentile is set to 50, each pixel in the resulting composite will be "average" value over the composite period.

D. Export parameters

1. **exportToDrive**: controls whether or not the composite is exported as a .tif to your personal Drive folder.
2. **crs** : specifies a projection, based on an EPSG number, to use for the output product.

Congratulations! You have completed this exercise, and you now know how to create a cloud-free Landsat composite from Google Earth Engine.