

ADVANCED GIS AND REMOTE SENSING (IRGN 490)

Ran Goldblatt

**Lecture 9:
Google Earth Engine API-1
Images and Image Collections**

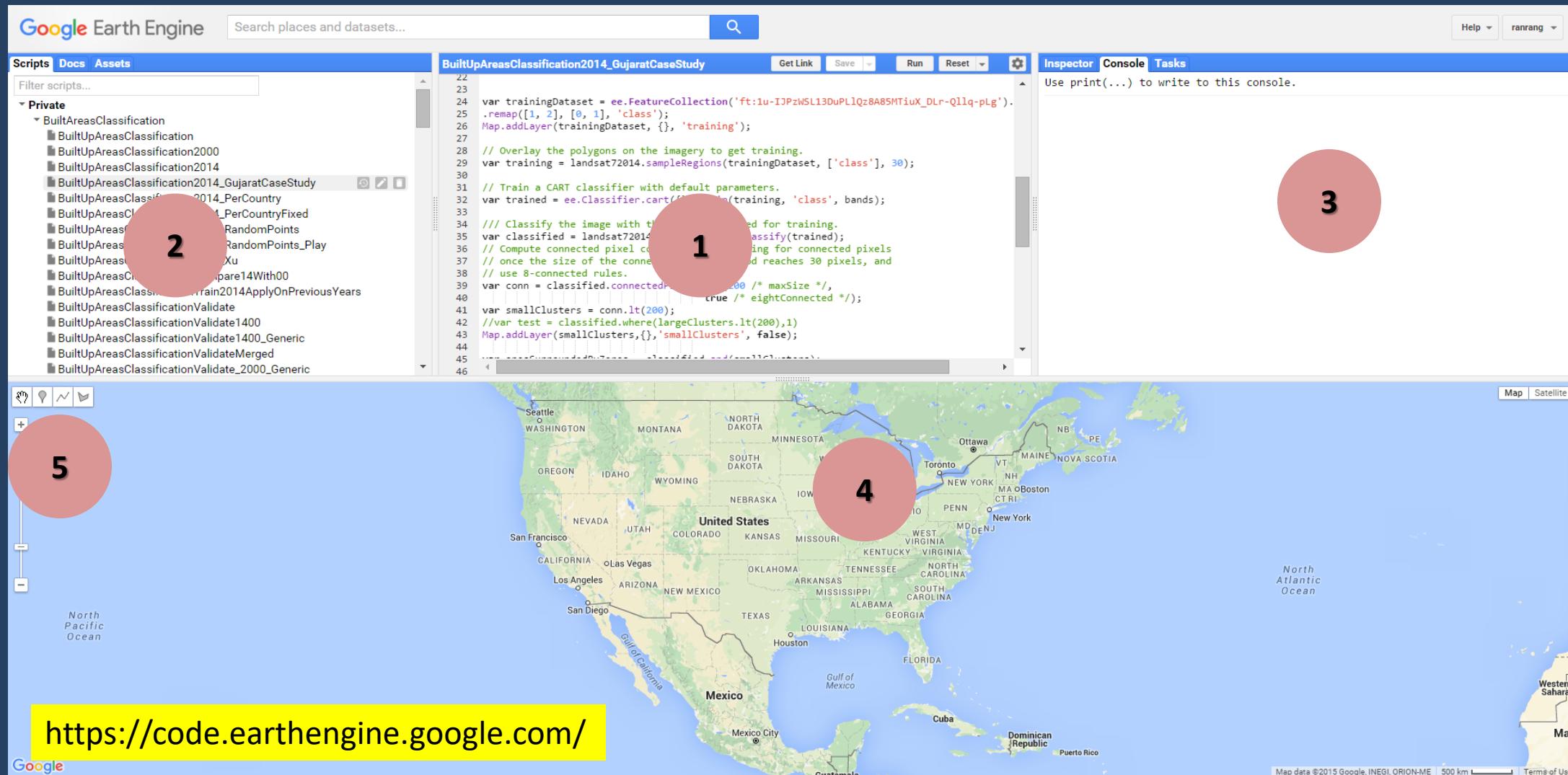
<https://code.earthengine.google.com/>

Google Earth Engine Coder (API)

- Used to run algorithms on georeferenced imagery and vectors.
- Includes a library of functions that can be applied to imagery for display and analysis.
- Vector datasets (feature collections) are used via Google Fusion Tables.
- MUCH more control on the analysis process

The Application Programming Interface (API) of Google Earth Engine (EE)

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



- 1- Code editor | 2- Archive panel, documentation, examples | 3- Report panel, debugging and output
4- Display screen | 5- Draw feature geometries**

Scripts Docs Assets

Filter scripts...

Private

Shared (14)

- CoralBleaching
- EconomicActivityAndNTL
- IRGN490_Goldblatt
 - API1
 - API2
 - + New folder
 - + New file
- NTL_Research
- SandBox
- SemiAridEcosystems
- UrbanByNTLInWork
- VietnamProject
- trial
- EE201

New Script

Get Link Save Run Reset Inspector Console Tasks

Use print(...) to write to this console.



Scripts Docs Assets

ee.DateRange
ee.Dictionary
ee.ErrorMargin
ee.Feature
ee.FeatureCollection
ee.Filter
ee.Geometry
ee.Image
ee.ImageCollection

ee.ImageCollection(args)
ee.ImageCollection.fromImages(images)
ee.ImageCollection.load(id, version)
aggregate_array(property)
aggregate_count(property)
aggregate count distinct(property)

New Script Get Link Save Run Reset Inspector Console Tasks

Use print(...) to write to this console.

Scripts Docs Assets

NEW C

users/rgoldblatt
users/slums
This folder is empty.
+ New folder
+ New image collection

New Script Get Link Save Run Reset Inspector Console Tasks

Use print(...) to write to this console.

Advanced GIS and Remote Sensing | Winter 2017 |
Ran Goldblatt

Scripts Docs Assets

India_builtUp_areas_L8_VisualClassification *

Get Link Save Run Reset Inspector Console Tasks

```

Imports (1 entry) ↗
  var geometry: Polygon, 4 vertices ⚙️ ⓘ

1 // Use these bands for prediction.
2 var bands = ['B1', 'B2', 'B3', 'B4', 'B5', 'B6', 'B7', 'B8', 'B9', 'B10', 'B11'];
3
4 var landsat8 = ee.ImageCollection('LANDSAT/LC8_L1T').filterDate('2014-01-01', '2014-12-31');
5 var landsat8simpleComposite = ee.Algorithms.Landsat.simpleComposite(landsat8);
6 Map.addLayer(landsat8simpleComposite,{},'landsat8simpleComposite')

// /// call the feature collections.
// var agroClimaticZoneForTest = ee.Filter.eq('AgroClimaticZone',13);

// var fullDataset =
// ee.FeatureCollection('ft:1SJY_YOrflLegIoitshZBahK44BSscfAPJh_S30wd').filter(agroClimaticZoneForTest);

// /// define the classifiers to be used.
// var randomForest = ee.Classifier.randomForest({'numberOfTrees': 10});
// var cart = ee.Classifier.cart({});


```

Point (72.9547, 23.4292) at 153m/px

Pixels

- landsat8simpleComposite: Image (11 bands) ⓘ

B1:	37
B2:	33
B3:	30
B4:	29
B5:	79
B6:	64
B7:	46
B8:	29
B9:	0
B10:	204
B11:	203

Objects

Scripts Docs Assets

India_builtUp_areas_L8_VisualClassification *

Get Link Save Run Reset Inspector Console **Tasks**

```

Imports (1 entry) ↗
  var geometry: Polygon, 4 vertices ⚙️ ⓘ

1 // Use these bands for prediction.
2 var bands = ['B1', 'B2', 'B3', 'B4', 'B5', 'B6', 'B7', 'B8', 'B9', 'B10', 'B11'];
3
4 var landsat8 = ee.ImageCollection('LANDSAT/LC8_L1T').filterDate('2014-01-01', '2014-12-31');
5 var landsat8simpleComposite = ee.Algorithms.Landsat.simpleComposite(landsat8);
6 Map.addLayer(landsat8simpleComposite,{},'landsat8simpleComposite')

// /// call the feature collections.
// var agroClimaticZoneForTest = ee.Filter.eq('AgroClimaticZone',13);

// var fullDataset =
// ee.FeatureCollection('ft:1SJY_YOrflLegIoitshZBahK44BSscfAPJh_S30wd').filter(agroClimaticZoneForTest);

// /// define the classifiers to be used.
// var randomForest = ee.Classifier.randomForest({'numberOfTrees': 10});
// var cart = ee.Classifier.cart({});


```

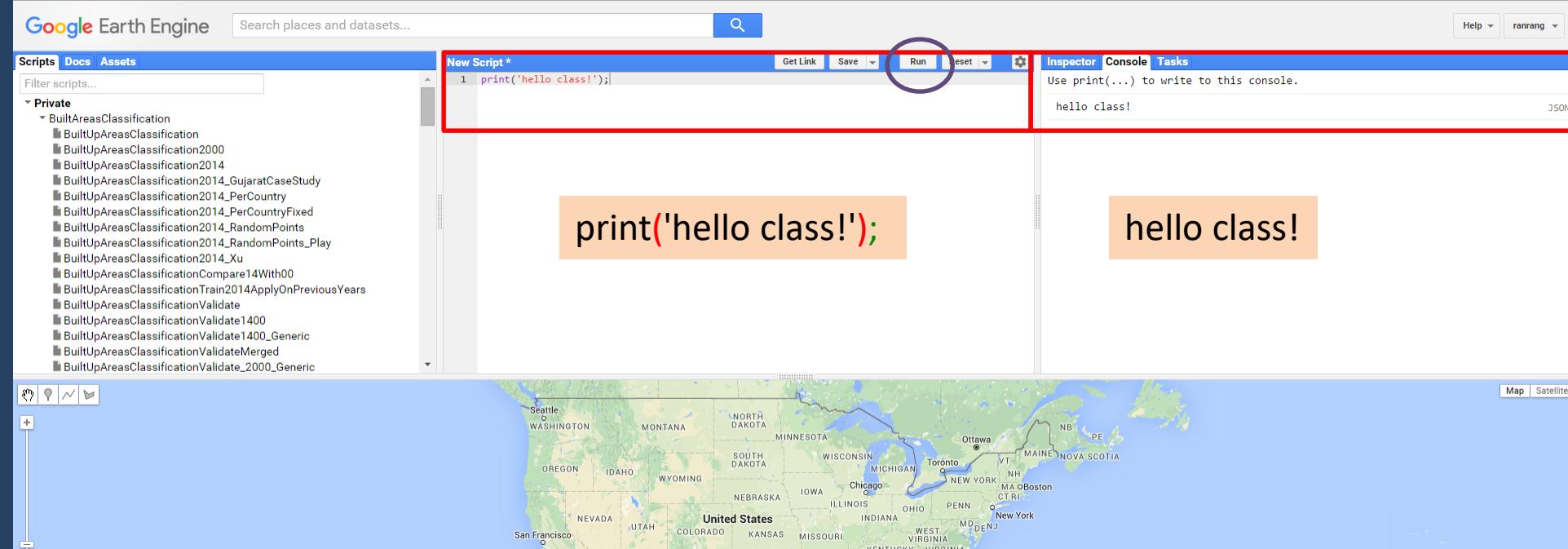
Older tasks

Completed

- HCM LU Functionallity_MedianValues ✓
- accuracyAssesementWith_GUE_BU_Mexico ✓
- accuracyAssesementWith_GUE_BU_USA ✓
- accuracyAssesementWith_GUE_BU_India ✓
- accuracyAssesementWith_GUE_BU_Mexico ✓
- accuracyAssesementWith_GUE_BU_USA ✓
- accuracyAssesementWith_GUE_BU_India ✓
- accuracyAssesementWith_GUE_BU_Mexico ✓
- accuracyAssesementWith_GUE_BU_USA ✓
- accuracyAssesementWith_GUE_BU_India ✓
- IndiaTestGrid20_011917B ✓
- IndiaTestGrid20_011917 ✓

Advanced GIS and Remote Sensing | Winter 2017 | Ran Goldblatt

Let's begin...



JavaScript **statements**

- JavaScript statements have a pair of **parentheses ()**
- They end with a **semicolon ;**
- Anything between single quotes is called a '**string**'.
- **Strings:**
 - Used to print out readable information.
 - Define sets of characters that name datasets or parts of datasets

`print('hello class');`

Image and Image collection = Rasters

Images

- Pixels with associated values.
- An image can have multiple bands
- Each band in an image contains:
 - A name
 - Pixel value
 - Resolution
 - Projection

The screenshot shows a GIS application interface. On the left is a satellite image of a landscape with a yellow circle highlighting a specific pixel. A callout arrow points from this pixel to a detailed view on the right. The right side displays the following information:

- Point (-115.1038, 37.6273) at 611m/px**
- Pixels**
 - Layer 1: Image (7 bands)**
B1: 47
B2: 51
B3: 60
B4: 72
B5: 76
B6_VCID_2: 208
B7: 65
- Objects**

Below this is a zoomed-in view of the satellite image with a purple magnifying glass highlighting the same yellow circle, indicating the location of the analyzed pixel.

Inspector **Console** **Tasks**
Use print(...) to write to this console.

```
Image LANDSAT/LE7_TOA_1YEAR/2014 (7 bands)
  type: Image
  id: LANDSAT/LE7_TOA_1YEAR/2014
  version: 1423795095111000
  bands: List (7 elements)
    0: "B1", unsigned int8, EPSG:4326, 1335834x638232 px
    1: "B2", unsigned int8, EPSG:4326, 1335834x638232 px
    2: "B3", unsigned int8, EPSG:4326, 1335834x638232 px
    3: "B4", unsigned int8, EPSG:4326, 1335834x638232 px
    4: "B5", unsigned int8, EPSG:4326, 1335834x638232 px
    5: "B6_VCID_2", unsigned int8, EPSG:4326, 1335834x638232 px
    6: "B7", unsigned int8, EPSG:4326, 1335834x638232 px
  properties: Object (1 property)
```

Advanced GIS and Remote Sensing | Winter 2017 | Ran Goldblatt

Image collection

- Collections/
Mosaics of EE images
- *Example:*
 - A collection of all L8 images in a given time period
 - Each image collection has an ID.

Inspector Console Tasks

Use print(...) to write to this console.

```
-> ImageCollection LANDSAT/LC8_L1T_8DAY_TOA (124 elements, 12 bands)
  type: ImageCollection
  id: LANDSAT/LC8_L1T_8DAY_TOA
  version: 1449778992138000
  > bands: List (12 elements)
  > features: List (124 elements)
    > 0: Image LANDSAT/LC8_L1T_8DAY_TOA/20130407 (12 bands)
    > 1: Image LANDSAT/LC8_L1T_8DAY_TOA/20130415 (12 bands)
    > 2: Image LANDSAT/LC8_L1T_8DAY_TOA/20130423 (12 bands)
    > 3: Image LANDSAT/LC8_L1T_8DAY_TOA/20130501 (12 bands)
    > 4: Image LANDSAT/LC8_L1T_8DAY_TOA/20130509 (12 bands)
    > 5: Image LANDSAT/LC8_L1T_8DAY_TOA/20130517 (12 bands)
    > 6: Image LANDSAT/LC8_L1T_8DAY_TOA/20130525 (12 bands)
    > 7: Image LANDSAT/LC8_L1T_8DAY_TOA/20130602 (12 bands)
    > 8: Image LANDSAT/LC8_L1T_8DAY_TOA/20130610 (12 bands) (highlighted)
    > 9: Image LANDSAT/LC8_L1T_8DAY_TOA/20130618 (12 bands)
    > 10: Image LANDSAT/LC8_L1T_8DAY_TOA/20130626 (12 bands)
    > 11: Image LANDSAT/LC8_L1T_8DAY_TOA/20130704 (12 bands)
    > 12: Image LANDSAT/LC8_L1T_8DAY_TOA/20130712 (12 bands)
    > 13: Image LANDSAT/LC8_L1T_8DAY_TOA/20130720 (12 bands)
    > 14: Image LANDSAT/LC8_L1T_8DAY_TOA/20130728 (12 bands)
    > 15: Image LANDSAT/LC8_L1T_8DAY_TOA/20130805 (12 bands)
    > 16: Image LANDSAT/LC8_L1T_8DAY_TOA/20130813 (12 bands)
    > 17: Image LANDSAT/LC8_L1T_8DAY_TOA/20130821 (12 bands)
```

“Calling” Image Collections

Option 1

The screenshot shows the Google Earth Engine interface. In the top left, the title "Google Earth Engine" is visible. Below it, a search bar contains the text "landsat 7 annual", which is highlighted with a red oval. To the right of the search bar are buttons for "Get Link", "Save", "Run", "Reset", and a settings gear icon. The main area is divided into sections: "SCRIPTS", "DOCS", and "ASSETS". Under "ASSETS", there is a "FILTER scripts..." dropdown and a "RASTERS" section. The "RASTERS" section lists several datasets, with "Landsat 7 annual TOA percentile composites" being the first item. Below the Rasters section is a "TABLES" section containing numerous tables. At the bottom of the interface is a map of the United States and surrounding regions, including parts of Canada, Mexico, and the North Atlantic. The map shows state/province boundaries and major cities. A legend at the bottom left identifies symbols for location, scale, and other map controls.

Searching and importing images

One method to access images and image collections:
“search” and import the dataset. No need to know the ID.

The screenshot illustrates the process of importing a Landsat 7 annual TOA percentile composite into the Google Earth Engine workspace.

Left Panel (Search Results):

- Search bar: landsat 7 annual
- Places: None
- Rasters:
 - Landsat 7 Annual TOA percentile composites
 - Landsat 7 Annual TOA Reflectance Composite (highlighted with a yellow box)
 - Landsat 7 Annual NDVI Composite
 - Landsat 7 Annual Raw Composite
 - Landsat 7 Annual EVI Composite
 - Landsat 7 Annual NDWI Composite
 - Landsat 7 Annual BAI Composite
 - Landsat 7 Annual Greenest-Pixel TOA Reflectance Composite
 - more »
- Tables: None

Right Panel (Import Confirmation):

Landsat 7 annual TOA percentile composites

These 1-year composites were created from all Landsat 7 images in the specified composite period, excluding images marked with negative sun elevation in their metadata. The composites were created using the `ee.Algorithms.Landsat.simpleComposite()` method with its default settings. Reflectance ([0,1]) in bands B1, B2, B3, B4, B5, and B7 is scaled to 8 bits ([0,255]) and temperature in band B6_VCID_2 is converted to units of Kelvin-100.

Data availability (time)
Jan 1, 1999 - Dec 31, 2014

Provider
Google

Tags
landsat, usgs, annual, global, l7, le7, etm, toa, percentile

ImageCollection ID
LANDSAT/LE7_TOA_1YEAR

Import

Central Callout: "Import" will add the image to the workspace

Bottom Callout: This image collection has a name:
LANDSAT/LE7_TOA_1YEAR

Map View: A world map showing the United States and surrounding regions, with state/province boundaries and city labels like Seattle, San Francisco, Los Angeles, and New York City.

Page Footer: Advanced GIS and Remote Sensing | Winter 2017 | Ran Goldblatt

The screenshot shows the Google Earth Engine interface. In the top left, there's a search bar with 'landsat 8 32'. The main area is a 'New Script' editor with the following code:

```
Imports (1 entry)
var imageCollection: ImageCollection "Landsat 8 32-Day TOA Reflectance Composit..."
```

The word 'var' is highlighted with a red circle. The interface includes tabs for Scripts, Docs, Assets, and a sidebar with a 'Private' folder containing numerous built-in scripts related to built-up areas classification.

Variables

- Variables store **values** that make the code **more readable**.
- Must begin with a letter (or a \$ or _)
- They are CASE-sensitive
- The first time you use a given variable, start with the word **var**.

The screenshot shows the 'New Script' editor again. The code is identical to the previous one:

```
Imports (1 entry)
var landsat8TOA: ImageCollection "Landsat 8 8-Day TOA Reflectance Composite" (12 bands)
```

The word 'var' is circled in red. A green callout box with the text 'You can choose the name of the variable!' points to the circled 'var' keyword.

The `print()` statement can be used to get information about the added image collection:

```
print(landsat8TOA);
```

The screenshot shows a Jupyter Notebook environment. On the left, a 'New Script *' cell contains the following Python code:

```
Imports (1 entry) ▾
var landsat8TOA: ImageCollection "Landsat 8 8-Day TOA Reflectance Composite" (12 bands)
1 print(landsat8TOA);
2
3
```

On the right, the 'Console' tab is selected, displaying the output of the `print(landsat8TOA)` command. The output is a JSON object representing the image collection:

```
Use print(...) to write to this console.

- ImageCollection LANDSAT/LC8_L1T_8DAY_TOA (124 elements, 12 bands)
  type: ImageCollection
  id: LANDSAT/LC8_L1T_8DAY_TOA
  version: 1449778992138000
  bands: List (12 elements)
  features: List (124 elements)
    0: Image LANDSAT/LC8_L1T_8DAY_TOA/20130407 (12 bands)
    1: Image LANDSAT/LC8_L1T_8DAY_TOA/20130415 (12 bands)
    2: Image LANDSAT/LC8_L1T_8DAY_TOA/20130423 (12 bands)
    3: Image LANDSAT/LC8_L1T_8DAY_TOA/20130501 (12 bands)
    4: Image LANDSAT/LC8_L1T_8DAY_TOA/20130509 (12 bands)
    5: Image LANDSAT/LC8_L1T_8DAY_TOA/20130517 (12 bands)
    6: Image LANDSAT/LC8_L1T_8DAY_TOA/20130525 (12 bands)
    7: Image LANDSAT/LC8_L1T_8DAY_TOA/20130602 (12 bands)
    8: Image LANDSAT/LC8_L1T_8DAY_TOA/20130610 (12 bands)
    9: Image LANDSAT/LC8_L1T_8DAY_TOA/20130618 (12 bands)
    10: Image LANDSAT/LC8_L1T_8DAY_TOA/20130626 (12 bands)
    11: Image LANDSAT/LC8_L1T_8DAY_TOA/20130704 (12 bands)
    12: Image LANDSAT/LC8_L1T_8DAY_TOA/20130712 (12 bands)
    13: Image LANDSAT/LC8_L1T_8DAY_TOA/20130720 (12 bands)
    14: Image LANDSAT/LC8_L1T_8DAY_TOA/20130728 (12 bands)
    15: Image LANDSAT/LC8_L1T_8DAY_TOA/20130805 (12 bands)
    16: Image LANDSAT/LC8_L1T_8DAY_TOA/20130813 (12 bands)
    17: Image LANDSAT/LC8_L1T_8DAY_TOA/20130821 (12 bands)
    18: Image LANDSAT/LC8_L1T_8DAY_TOA/20130829 (12 bands)
```

A red box highlights the 'features' list, which contains 124 individual image objects, each with a unique timestamp.

This image collection has a name: **LANDSAT/LC8_L1T_8DAY_TOA**

Another method to add images and image collections:

ee.ImageCollection()

This function “tells” EE that a given ID is a name of an image collection:

LANDSAT/LC8_L1T_8DAY_TOA is an Image collection- Access it as a parameter.

ee.ImageCollection('LANDSAT/LC8_L1T_8DAY_TOA')

The screenshot shows the Earth Engine code editor interface. On the left, a script pane contains the following code:

```
1 var landsat8TOA = ee.ImageCollection('LANDSAT/LC8_L1T_8DAY_TOA');
2 print(landsat8TOA);
```

In the center, a blue box highlights the line `var something = ee.ImageCollection()`. On the right, the Inspector pane displays the details of the `landsat8TOA` object, which is identified as an `ImageCollection` named `LANDSAT/LC8_L1T_8DAY_TOA` with 124 elements and 12 bands. The pane also shows the individual images within the collection, each with a timestamp from 20130407 to 20130821.

```
1 var landsat8TOA = ee.ImageCollection('LANDSAT/LC8_L1T_8DAY_TOA');
2 print(landsat8TOA);

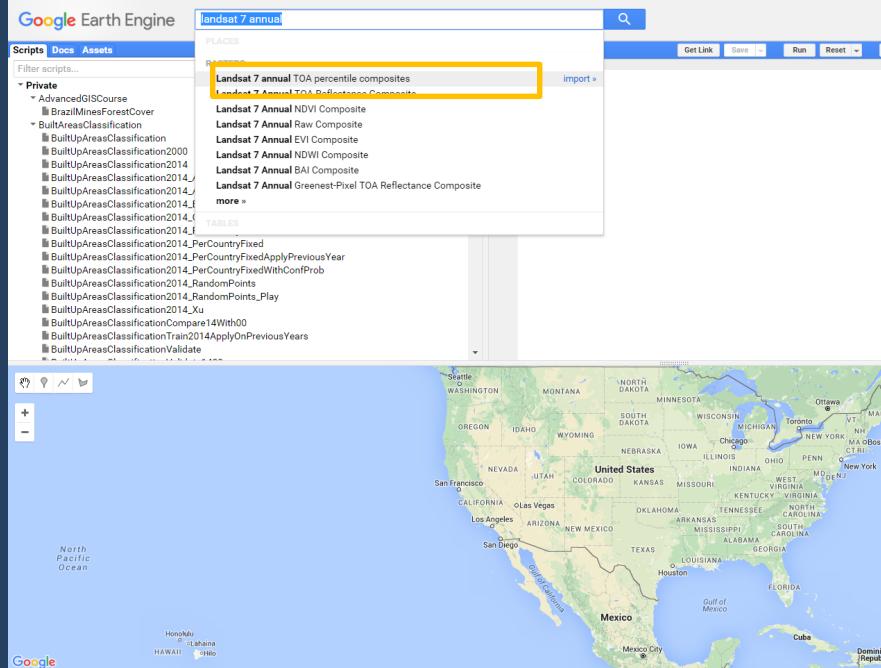
var something = ee.ImageCollection()

New Script * Get Link Save Run Reset Inspector Console Tasks
Use print(...) to write to this console.

- ImageCollection LANDSAT/LC8_L1T_8DAY_TOA (124 elements, 12 bands)
  type: ImageCollection
  id: LANDSAT/LC8_L1T_8DAY_TOA
  version: 1449778992138000
  bands: List (12 elements)
  features: List (124 elements)
    ▾ 0: Image LANDSAT/LC8_L1T_8DAY_TOA/20130407 (12 bands)
    ▾ 1: Image LANDSAT/LC8_L1T_8DAY_TOA/20130415 (12 bands)
    ▾ 2: Image LANDSAT/LC8_L1T_8DAY_TOA/20130423 (12 bands)
    ▾ 3: Image LANDSAT/LC8_L1T_8DAY_TOA/20130501 (12 bands)
    ▾ 4: Image LANDSAT/LC8_L1T_8DAY_TOA/20130509 (12 bands)
    ▾ 5: Image LANDSAT/LC8_L1T_8DAY_TOA/20130517 (12 bands)
    ▾ 6: Image LANDSAT/LC8_L1T_8DAY_TOA/20130525 (12 bands)
    ▾ 7: Image LANDSAT/LC8_L1T_8DAY_TOA/20130602 (12 bands)
    ▾ 8: Image LANDSAT/LC8_L1T_8DAY_TOA/20130610 (12 bands)
    ▾ 9: Image LANDSAT/LC8_L1T_8DAY_TOA/20130618 (12 bands)
    ▾ 10: Image LANDSAT/LC8_L1T_8DAY_TOA/20130626 (12 bands)
    ▾ 11: Image LANDSAT/LC8_L1T_8DAY_TOA/20130704 (12 bands)
    ▾ 12: Image LANDSAT/LC8_L1T_8DAY_TOA/20130712 (12 bands)
    ▾ 13: Image LANDSAT/LC8_L1T_8DAY_TOA/20130720 (12 bands)
    ▾ 14: Image LANDSAT/LC8_L1T_8DAY_TOA/20130728 (12 bands)
    ▾ 15: Image LANDSAT/LC8_L1T_8DAY_TOA/20130805 (12 bands)
    ▾ 16: Image LANDSAT/LC8_L1T_8DAY_TOA/20130813 (12 bands)
    ▾ 17: Image LANDSAT/LC8_L1T_8DAY_TOA/20130821 (12 bands)
```

var landsat8TOA = ee.ImageCollection('LANDSAT/LC8_L1T_8DAY_TOA');
print(landsat8TOA);

Let's search for another image collection:



Landsat 7 annual TOA percentile composites

These 1-year composites were created from all Landsat 7 images in the specified composite period, excluding images marked with a negative sun elevation in their metadata. The composites were created using the `ee.Algorithms.Landsat.simpleComposite()` method with its default settings. Reflectance ([0,1]) in bands B1, B2, B3, B4, B5, and B7 is scaled to 8 bits ([0,255]) and temperature in band B6_VCID_2 is converted to units of Kelvin-100.

Data availability (time)
Jan 1, 1999 - Dec 31, 2014

Provider
Google

Tags
landsat, usgs, annual, global, l7, le7, etm, toa, percentile

ImageCollection ID
`'LANDSAT/LE7_TOA_1YEAR'`

Import

A large blue arrow points downwards from the 'Import' button towards the code block below.

```
var landsat7TOA = ee.ImageCollection('LANDSAT/LE7_TOA_1YEAR');
print(landsat7TOA);
```

New Script *

Get Link Save Run Reset

```
1 var landsat7TOA = ee.ImageCollection('LANDSAT/LE7_TOA_1YEAR');
2 print(landsat7TOA);
```

```
var landsat7TOA =
ee.ImageCollection('LANDSAT/LE7_TOA_1YEAR');
print(landsat7TOA);
```

Inspector Console Tasks

Use `print(...)` to write to this console.

```
▼ ImageCollection LANDSAT/LE7_TOA_1YEAR (16 elements, 7 bands)
  type: ImageCollection
  id: LANDSAT/LE7_TOA_1YEAR
  version: 1449779677092000
  ▶ bands: List (7 elements)
  ▶ features: List (16 elements)
    ▶ 0: Image LANDSAT/LE7_TOA_1YEAR/1999 (7 bands)
    ▶ 1: Image LANDSAT/LE7_TOA_1YEAR/2000 (7 bands)
    ▶ 2: Image LANDSAT/LE7_TOA_1YEAR/2001 (7 bands)
    ▶ 3: Image LANDSAT/LE7_TOA_1YEAR/2002 (7 bands)
    ▶ 4: Image LANDSAT/LE7_TOA_1YEAR/2003 (7 bands)
    ▶ 5: Image LANDSAT/LE7_TOA_1YEAR/2004 (7 bands)
    ▶ 6: Image LANDSAT/LE7_TOA_1YEAR/2005 (7 bands)
    ▶ 7: Image LANDSAT/LE7_TOA_1YEAR/2006 (7 bands)
    ▶ 8: Image LANDSAT/LE7_TOA_1YEAR/2007 (7 bands)
    ▶ 9: Image LANDSAT/LE7_TOA_1YEAR/2008 (7 bands)
    ▶ 10: Image LANDSAT/LE7_TOA_1YEAR/2009 (7 bands)
    ▶ 11: Image LANDSAT/LE7_TOA_1YEAR/2010 (7 bands)
    ▶ 12: Image LANDSAT/LE7_TOA_1YEAR/2011 (7 bands)
    ▶ 13: Image LANDSAT/LE7_TOA_1YEAR/2012 (7 bands)
    ▶ 14: Image LANDSAT/LE7_TOA_1YEAR/2013 (7 bands)
    ▶ 15: Image LANDSAT/LE7_TOA_1YEAR/2014 (7 bands)
      type: Image
      id: LANDSAT/LE7_TOA_1YEAR/2014
      version: 1423795095111000
      ▶ bands: List (7 elements)
      ▶ properties: Object (1 property)
        ▶ properties: Object (21 properties)
```

This is an image

The Image collection includes features → Images

Every image in Earth Engine has a name. Example:

LANDSAT/LE7_TOA_1YEAR/2014

The image ID that we want to call: **LANDSAT/LE7_TOA_1YEAR/2014**

We need to explicitly tell Earth Engine that **LANDSAT/LE7_TOA_1YEAR/2014** is an image.

We will use the function: **ee.Image()**

“Take the name of the image and access it as a parameter”.

```
var landsat7TOA2014 = ee.Image('LANDSAT/LE7_TOA_1YEAR/2014');
print(landsat7TOA2014);
```

The screenshot shows the Earth Engine code editor interface. On the left, under 'New Script *', there is a code editor window containing the following JavaScript code:

```
1 var landsat7TOA2014 = ee.Image('LANDSAT/LE7_TOA_1YEAR/2014');
2 print(landsat7TOA2014);
```

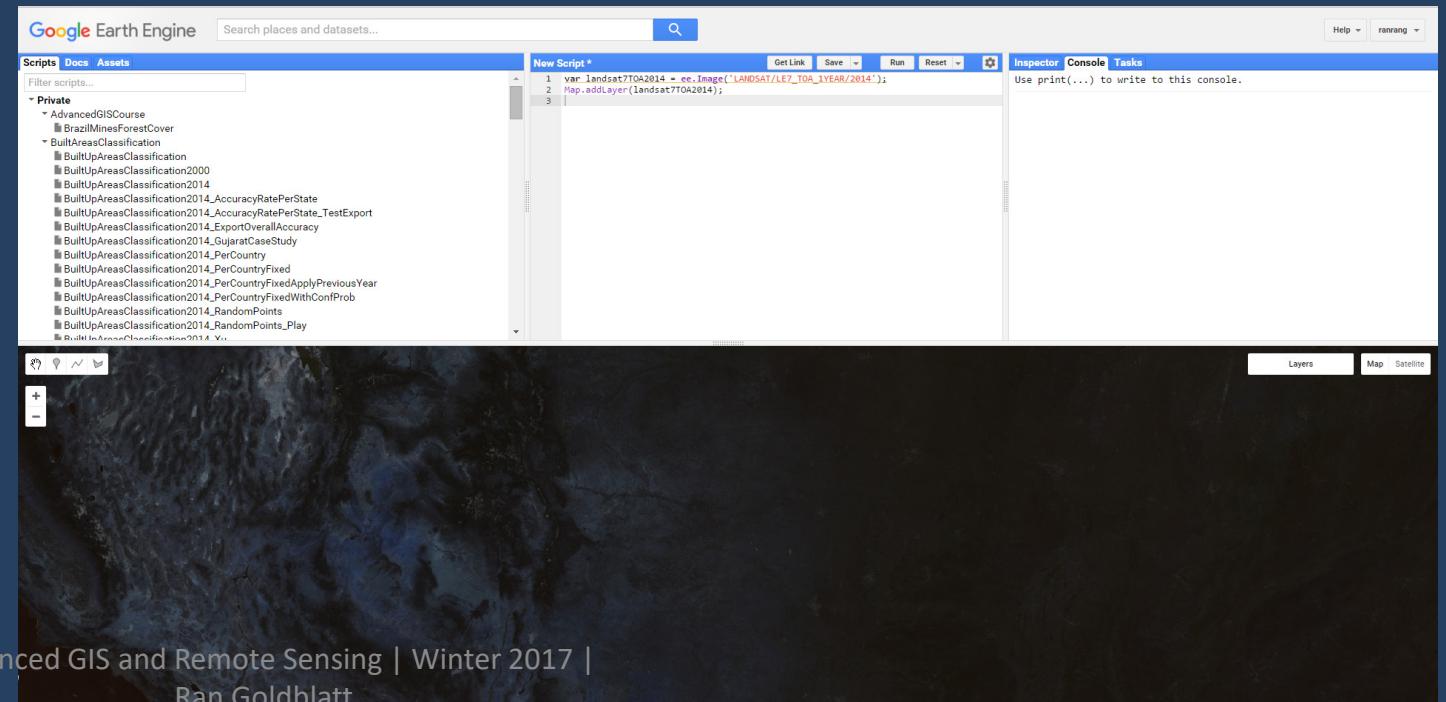
At the top of the interface are buttons for 'Get Link', 'Save', 'Run', 'Reset', and a settings gear icon. To the right of the code editor is the 'Inspector' tab, which is active. Below the tabs, a message says 'Use print(...) to write to this console.' The 'Inspector' panel displays the properties of the 'landsat7TOA2014' object, which is identified as an 'Image' with the id 'LANDSAT/LE7_TOA_1YEAR/2014'. It has 7 bands and one property. The properties object contains a single key-value pair where the key is 'version' and the value is '1423795095111000'.

Adding an image to the map:

Map.addLayer(something);

This statement says that we want the **Map.addLayer()** function to add something to the map.

```
var landsat7TOA2014 = ee.Image('LANDSAT/LE7_TOA_1YEAR/2014');
Map.addLayer(landsat7TOA2014);
```



What is wrong with the visualization?

Many images in Earth Engine have one or more bands.

For example, images from the **Landsat 7** satellite include:

Band 1 - blue

Band 2 - green

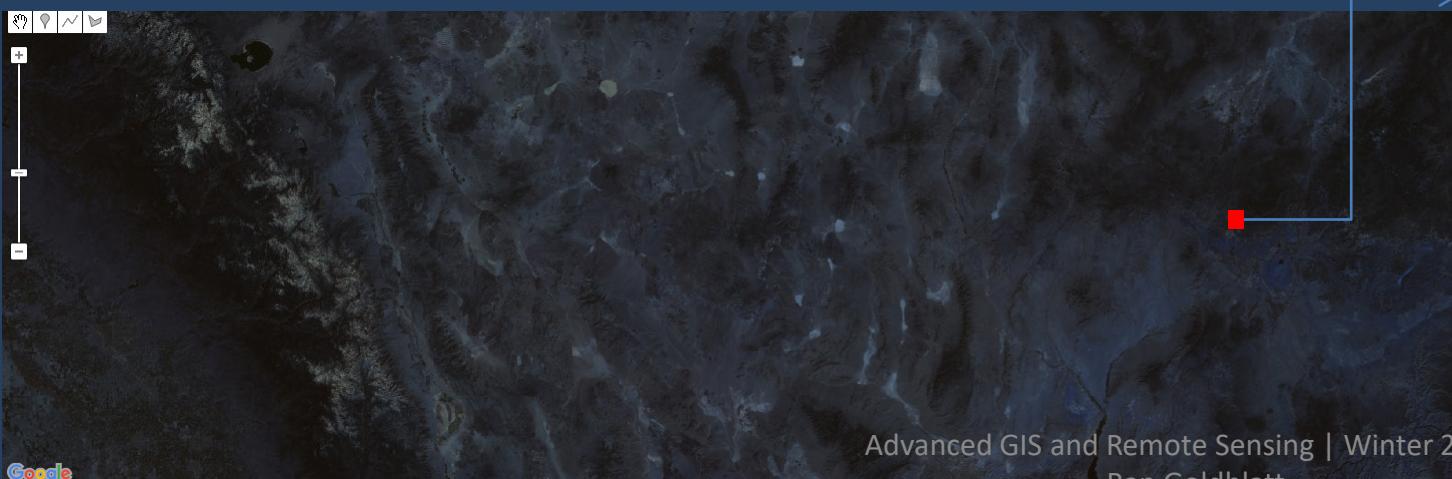
Band 3 - red

Band 4 - Near Infrared

Band 5 - Short-wave Infrared

Band 6 - Thermal Infrared

Band 7 - Short-wave Infrared



Inspector Console Tasks

Use print(...) to write to this console.

▼ Image LANDSAT/LE7_TOA_1YEAR/2014 (7 bands)

 type: Image
 id: LANDSAT/LE7_TOA_1YEAR/2014
 version: 1423795095111000

 bands: List (7 elements)

 ► 0: "B1", unsigned int8, EPSG:4326, 1335834x638232 px
 ► 1: "B2", unsigned int8, EPSG:4326, 1335834x638232 px
 ► 2: "B3", unsigned int8, EPSG:4326, 1335834x638232 px
 ► 3: "B4", unsigned int8, EPSG:4326, 1335834x638232 px
 ► 4: "B5", unsigned int8, EPSG:4326, 1335834x638232 px
 ► 5: "B6_VCID_2", unsigned int8, EPSG:4326, 1335834x638232 px
 ► 6: "B7", unsigned int8, EPSG:4326, 1335834x638232 px

 properties: Object (1 property)

► Point (-115.1038, 37.6273) at 611m/px

▼ Pixels

 ▼ Layer 1: Image (7 bands)

B1:	47
B2:	51
B3:	60
B4:	72
B5:	76
B6_VCID_2:	208
B7:	65

► Objects

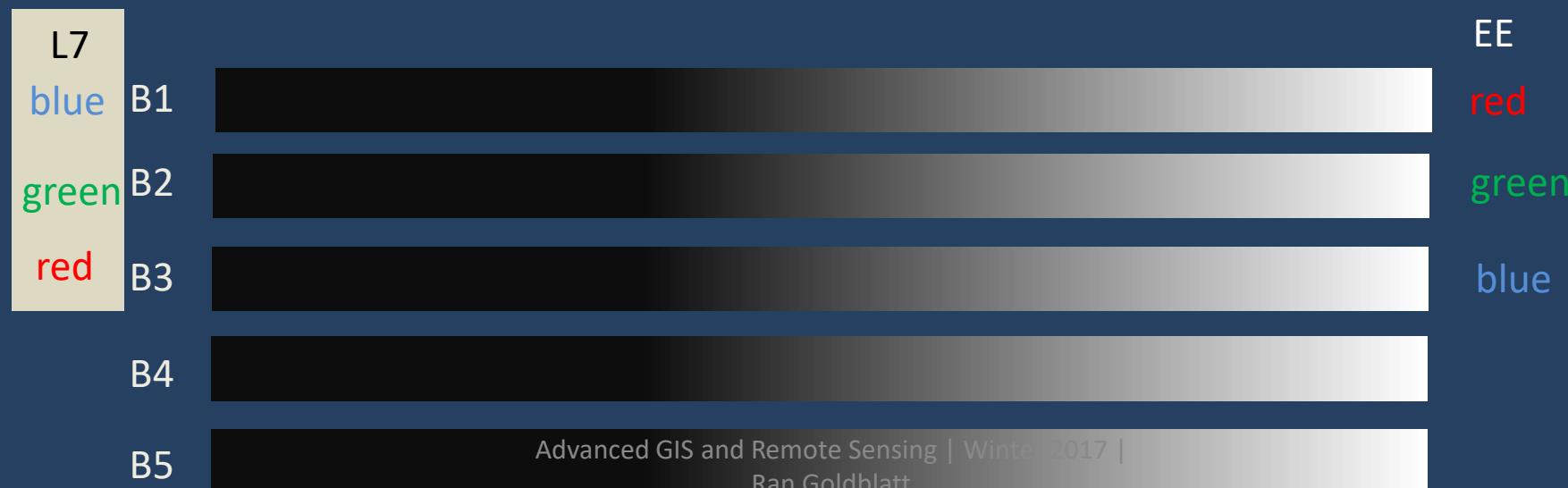
How does EE map the values in the image bands to colors on the map?

One band: value 0 is rendered as black and value 255 as white, with a linear gray gradient in between.



More than one band:

EE takes the first three bands and maps the first to red, the second to green, and the third to blue. The value of the pixel determines the intensity of the color.



The `Map.addLayer()` function can take a second parameter {} which describes how the image **should be visualized**.

```
Map.addLayer(ee.Image('LANDSAT/LE7_TOA_1YEAR/2014'),  
{}  
);
```

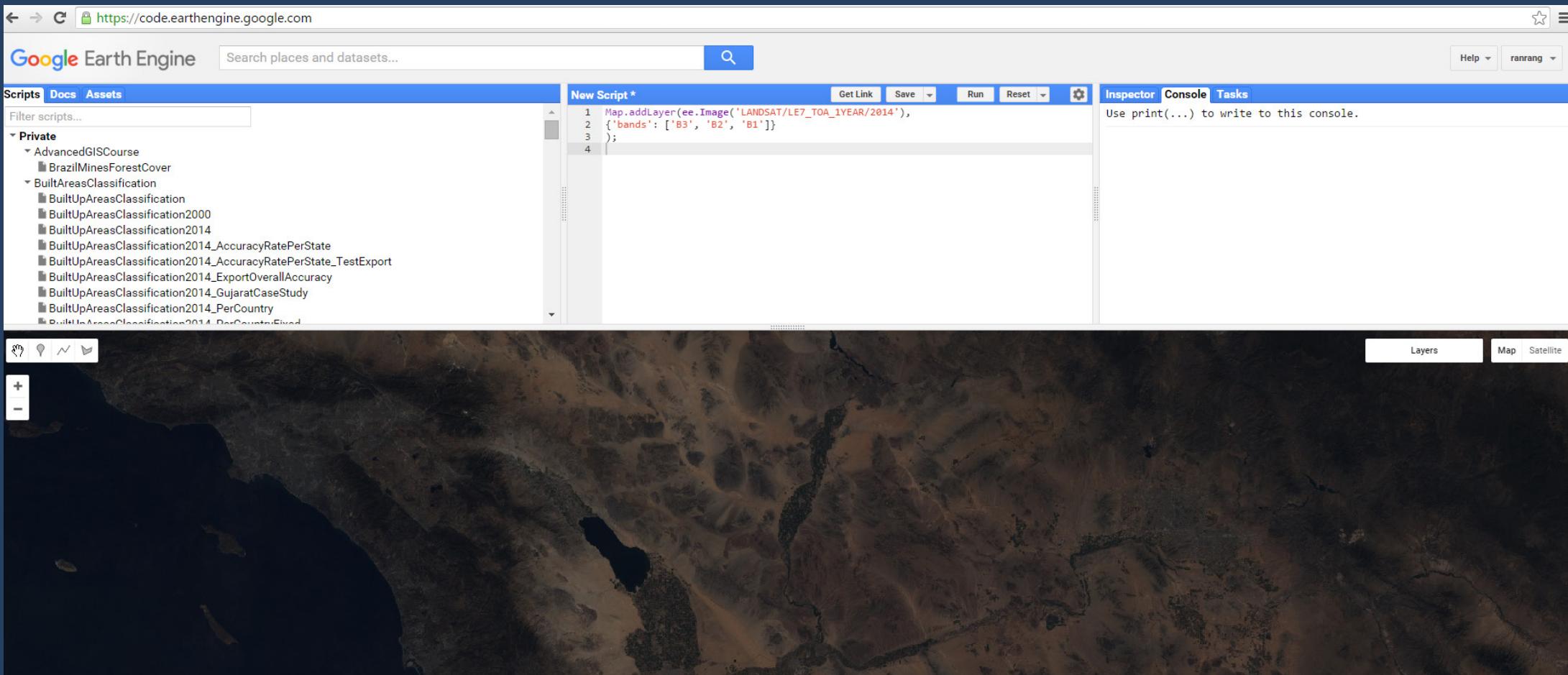
The format of this second parameter (visualization factor) is:
`{'string': value, 'string': value, ... , 'string': value}`

each string describes one **visualization factor**:
For example: 'min', 'max', 'bands'.

We want to add a *list* of bands to visualize:

```
Map.addLayer(ee.Image('LANDSAT/LE7_TOA_1YEAR/2014'),  
{'bands': ['B3', 'B2', 'B1']}  
);
```

A visualization of a single Landsat image in Earth Engine.



```
Map.addLayer(ee.Image('LANDSAT/LE7_TOA_1YEAR/2014'),  
{'bands': ['B3', 'B2', 'B1']}  
);
```

	R	G	B
L7	B3	B2	B1
L8	B4	B3	B2

Selecting specific band(s) in an Image

```
.select('band name')
```

Inspector Console Tasks
Use print(...) to write to this console.
Image LANDSAT/LE7_TOA_1YEAR/2014 (7 bands)
type: Image
id: LANDSAT/LE7_TOA_1YEAR/2014
version: 142379509511000
bands: List (7 elements)
0: "B1", unsigned int8, EPSG:4326, 1335834x638232 px
1: "B2", unsigned int8, EPSG:4326, 1335834x638232 px
2: "B3", **unsigned int8, EPSG:4326, 1335834x638232 px**
3: "B4", unsigned int8, EPSG:4326, 1335834x638232 px
4: "B5", unsigned int8, EPSG:4326, 1335834x638232 px
5: "B6_VCID_2", unsigned int8, EPSG:4326, 1335834x638232 px
6: "B7", unsigned int8, EPSG:4326, 1335834x638232 px
properties: Object (1 property)

```
Map.addLayer(ee.Image('LANDSAT/LE7_TOA_1YEAR/2014')
.select('B3'));
```

Which is the same as:

```
var L7 = ee.Image('LANDSAT/LE7_TOA_1YEAR/2014');
var B3_L7 = L7.select('B3');
Map.addLayer(B3_L7);
```

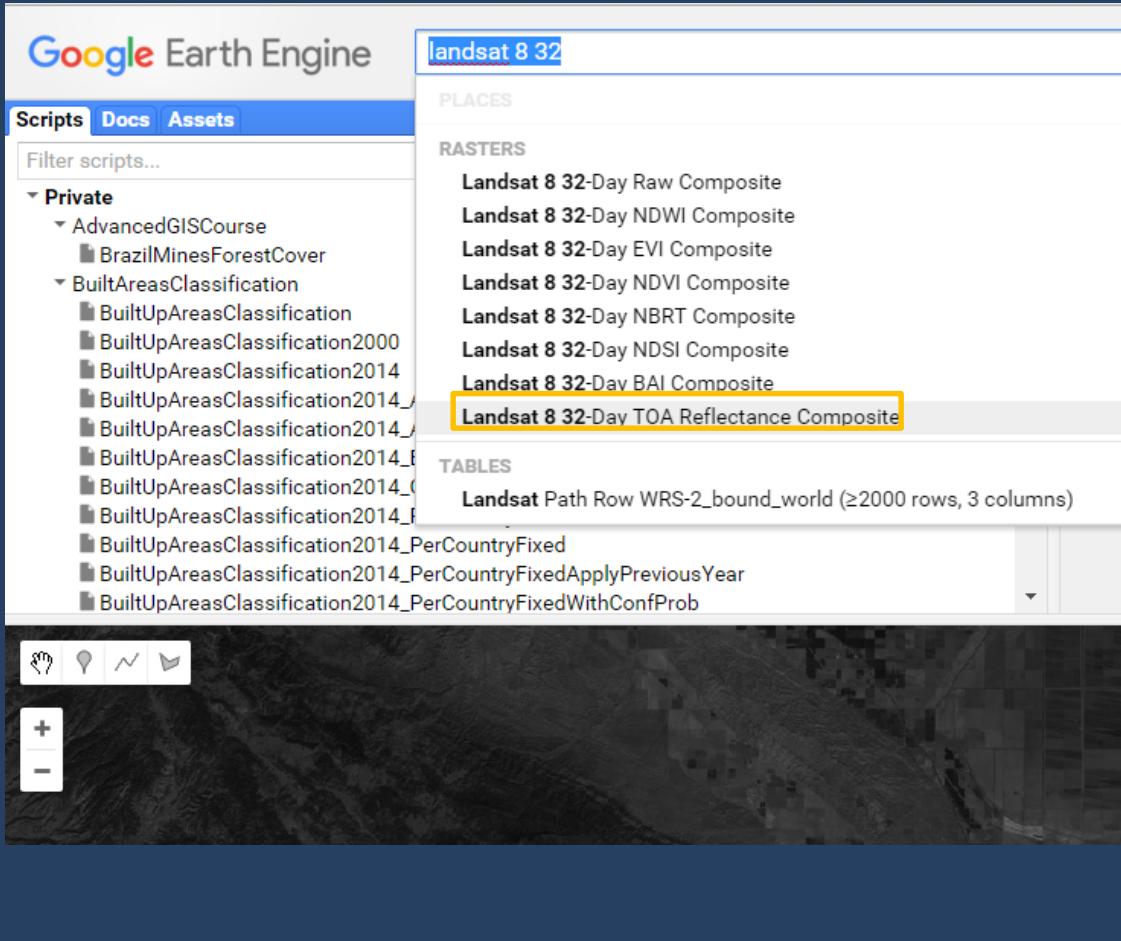
```
var L7 = ee.Image('LANDSAT/LE7_TOA_1YEAR/2014');
var B3_L7 = L7.select('B3');
Map.addLayer(B3_L7);
```

The screenshot shows the Google Earth Engine (GEE) web interface. The top navigation bar includes the GEE logo, a search bar, and user account information ('ranrang'). Below the header are tabs for 'Scripts', 'Docs', and 'Assets'. A sidebar on the left lists 'Private' scripts under 'AdvancedGISCourse' and other related datasets like 'BrazilMinesForestCover' and various 'BuiltUpAreasClassification' scripts from 2000 to 2014. The main workspace is titled 'New Script *' and contains the following code:

```
1 var L7 = ee.Image('LANDSAT/LE7_TOA_1YEAR/2014');
2 var B3_L7 = L7.select('B3');
3 Map.addLayer(B3_L7);
4 |
```

The 'Inspector' tab is active, showing a note: 'Use print(...) to write to this console.' The 'Console' tab is also visible. Below the script editor is a large satellite map of a rural landscape with agricultural fields and some dark, irregular areas. The map has a grid overlay. On the right side of the map are 'Layers', 'Map', and 'Satellite' buttons. At the bottom of the screen, there is a footer with the text 'Advanced GIS and Remote Sensing | Winter 2017 | Ran Goldblatt'.

Let's call another image collection:



The screenshot shows the Google Earth Engine interface. The search bar at the top contains the text "landsat 8 32". Below the search bar, the "RASTERS" section lists several Landsat 8 composites, with "Landsat 8 32-Day TOA Reflectance Composite" highlighted by a yellow box. The code editor on the right shows a snippet of JavaScript code:

```
('LANDSAT/LE7_TOA_1YEAR/2014');  
lect('B3');  
7);
```

A modal window for the "Landsat 8 32-Day TOA Reflectance Composite" is displayed. It includes a title, a detailed description of the composite creation process, and sections for data availability, provider, tags, and an import button.

Landsat 8 32-Day TOA Reflectance Composite

These Landsat 8 composites are made from Level L1T orthorectified scenes, using the computed top-of-atmosphere (TOA) reflectance. See [Chander et al. \(2009\)](#) for details on the TOA computation.

These composites are created from all the scenes in each 32-day period beginning from the first day of the year and continuing to the 352nd day of the year. The last composite of the year, beginning on day 353, will overlap the first composite of the following year by 20 days. All the images from each 32-day period are included in the composite, with the most recent pixel as the composite value.

Data availability (time)
Apr 7, 2013 - Dec 19, 2015

Provider
Google

Tags
landsat, usgs, l8, lc8, oli, tirs, 32day, toa

ImageCollection ID
LANDSAT/LC8_L1T_32DAY_TOA

Import

Task:

Filter an image collection only for scenes in a specific date
(second half of 2014)

Scripts

Docs

Assets

Filter scripts...

Private

- BuiltAreasClassification
 - BuiltUpAreasClassification
 - BuiltUpAreasClassification2000
 - BuiltUpAreasClassification2014
 - BuiltUpAreasClassification2014_GujaratCaseStudy
 - BuiltUpAreasClassification2014_PerCountry
 - BuiltUpAreasClassification2014_PerCountryFixed
 - BuiltUpAreasClassification2014_RandomPoints
 - BuiltUpAreasClassification2014_RandomPoints_Play
 - BuiltUpAreasClassification2014_Xu
 - BuiltUpAreasClassificationCompare14With00
 - BuiltUpAreasClassificationTrain2014ApplyOnPreviousYears
 - BuiltUpAreasClassificationValidate
 - BuiltUpAreasClassificationValidate1400
 - BuiltUpAreasClassificationValidate1400_Generic
 - BuiltUpAreasClassificationValidateMerged
 - BuiltUpAreasClassificationValidate_2000_Generic



North Pacific Ocean

Honolulu
Lahaina
Hawaii
Oahu

```
22
23
24 var trainingDataset = ee.FeatureCollection('ft:lu-IJPzWSL13DuPLlQz8A85MT1uX_DLr-Qllq-plG').
25 .remap([1, 2], [0, 1], 'class');
26 Map.addLayer(trainingDataset, {}, 'training');
27
28 // Overlay the polygons on the imagery to get training.
29 var training = landsat2014.sampleRegions(trainingDataset, ['class'], 30);
30
31 // Train a CART classifier with default parameters.
32 var trained = ee.Classifier.cart({}).train(training, 'class', bands);
33
34 // Classify the image with the same bands used for training.
```

Inspector Console Tasks

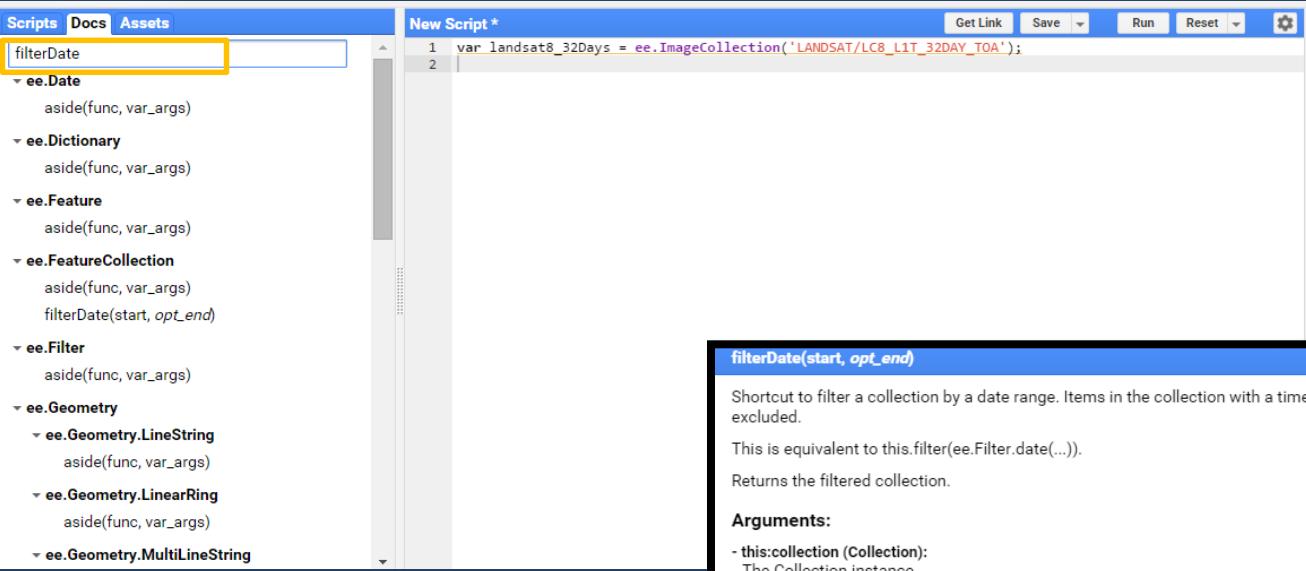
Use print(...) to write to this console.

Scripts Docs Assets

- Filter methods...
- ▶ ee.Algorithms
 - ▶ ee.Array
 - ▶ ee.Classifier
 - ▶ ee.ConfusionMatrix
 - ▶ ee.Date
 - ▶ ee.DateRange
 - ▶ ee.Dictionary
 - ▶ ee.ErrorMargin
 - ▶ ee.Feature
 - ▶ ee.FeatureCollection
 - ▶ ee.Filter
 - ▶ ee.Geometry



.filterDate()



The screenshot shows the Earth Engine code editor interface. On the left is a sidebar with tabs for 'Scripts', 'Docs', and 'Assets'. The 'Assets' tab is selected, showing a search bar with 'filterDate' typed in, which is highlighted with a yellow box. Below the search bar is a tree view of available functions: ee.Date, ee.Dictionary, ee.Feature, ee.FeatureCollection, ee.Filter, ee.Geometry, ee.Geometry.LineString, ee.Geometry.LinearRing, and ee.Geometry.MultiLineString. The main workspace is titled 'New Script *' and contains the following code:

```
1 var landsat8_32Days = ee.ImageCollection('LANDSAT/LC8_L1T_32DAY_TOA');
2
```

To the right of the workspace is a tooltip for the 'filterDate' function:

filterDate(start, opt_end)

Shortcut to filter a collection by a date range. Items in the collection with a time_start property that doesn't fall between the start and end dates will be excluded.

This is equivalent to this.filter(ee.Filter.date(...)).

Returns the filtered collection.

Arguments:

- **this:collection (Collection):** The Collection instance.
- **start (Date|Number|String):** The start date as a Date object, a string representation of a date, or milliseconds since epoch.
- **opt_end (Date/Number/String, optional):** The end date as a Date object, a string representation of a date, or milliseconds since epoch.

Returns: Collection

filterDate() method on an image collection.

Each action on an image collection is a *method*.

A method is applied on an image collection by

.methodName(arguments) after the image collection

ee.imageCollection.filterDate(date1, date2)

'2014-06-01', '2014-12-31'

'YYYY-MM-DD', 'YYYY-MM-DD'

```
ee.ImageCollection('LANDSAT/LC8_L1T_32DAY_TOA')  
.filterDate('2014-06-01', '2014-12-31');
```

Which is the same as:

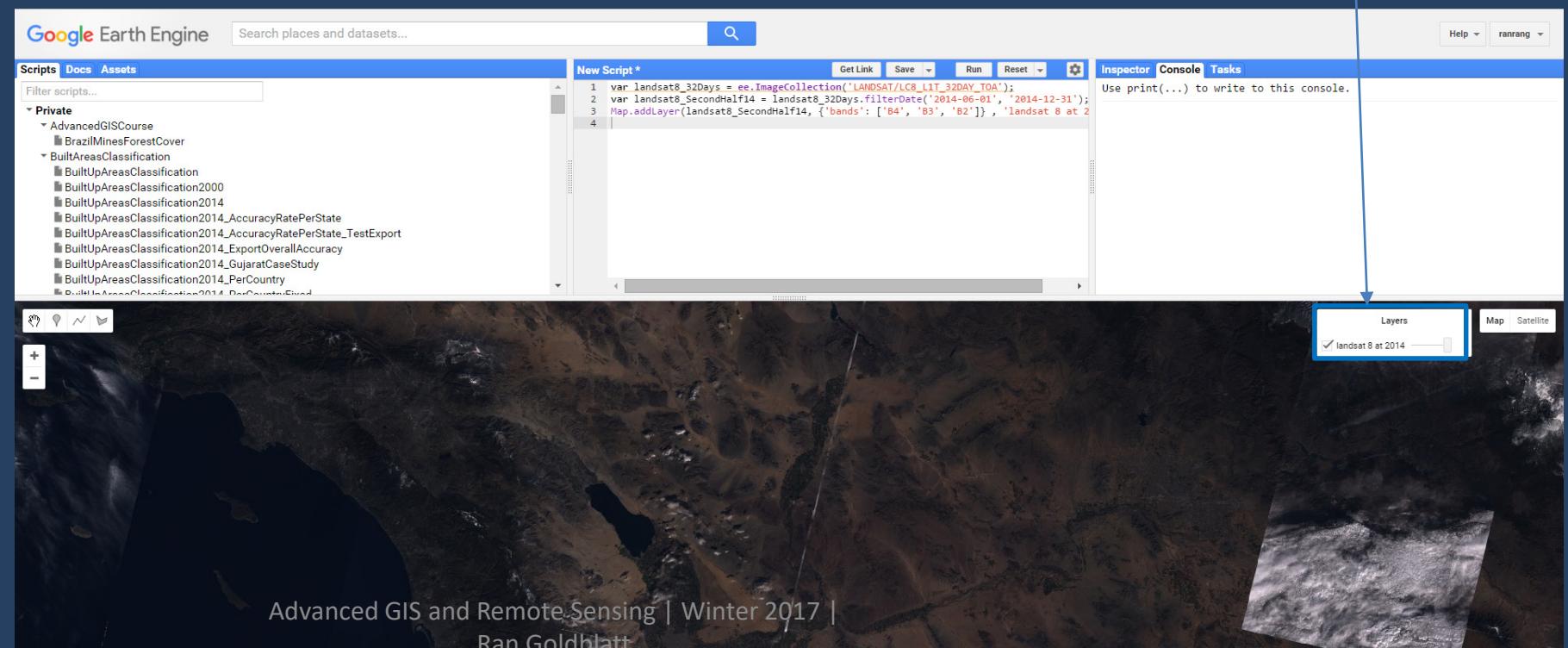
```
var L8 = ee.ImageCollection('LANDSAT/LC8_L1T_32DAY_TOA');  
  
var L8_SecondHalf = L8.filterDate('2014-06-01', '2014-12-31');
```

All together:

```
var L8 = ee.ImageCollection('LANDSAT/LC8_L1T_32DAY_TOA');
```

```
var L8_SecondHalf14 = L8.filterDate('2014-06-01', '2014-12-31');
```

```
Map.addLayer(L8_SecondHalf14, {'bands': ['B4', 'B3', 'B2']} , 'landsat 8 at 2014' );
```



Reducers

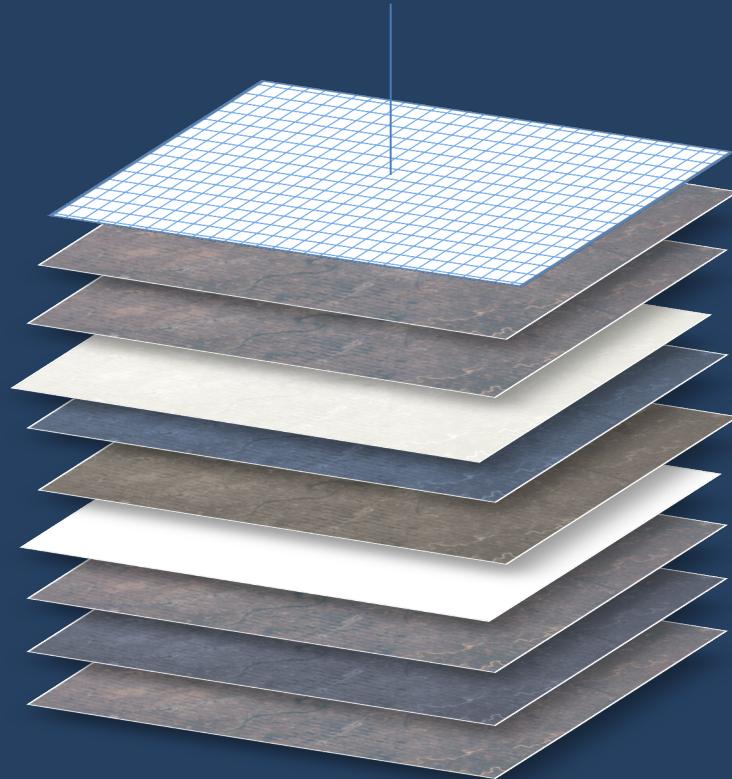
- Landsat 8 visits the same spot on the Earth **every 16 days**.
- Over a **6 month period**, there will be approximately **12 images**.
- Each pixel on the map is derived from a **stack of pixels** (a pixel from each image in the collection).
- The **default behavior** is to select the most recent pixel- the one from the recent scene in the stack.
- We can alter this behavior by using a **reducer**

For example: pick the median value in the stack.

Reducers

```
var imageCollection = ee.ImageCollection()
```

```
var minVal = imageCollection.min()  
var maxVal = imageCollection.max()  
var meanVal = imageCollection.mean()  
var medianVal = imageCollection.median()  
var firstVal = imageCollection.first()  
var sumVal = imageCollection.sum()
```



```
var L8_32Days = ee.ImageCollection('LANDSAT/LC8_L1T_32DAY_TOA');
var L8_SecondHalf14 = L8_32Days
.filterDate('2014-06-01', '2014-12-31').median();
Map.addLayer(L8_SecondHalf14, {"bands": ['B4', 'B3', 'B2']} , 'landsat 8 at 2014');
```



Creating a TOA calibrated composite of “raw” Landsat scenes

USGS Landsat 8 Raw Scenes (Orthorectified)

Landsat 8 DN values, representing scaled, calibrated at-sensor radiance, orthorectified scenes only.

This dataset contains the following bands:

- B1: Coastal aerosol (0.43 - 0.45 µm)
- B2: Blue (0.45 - 0.51 µm)
- B3: Green (0.53 - 0.59 µm)
- B4: Red (0.64 - 0.67 µm)
- B5: Near Infrared (0.85 - 0.88 µm)
- B6: Short-wave Infrared 1 (1.57 - 1.65 µm)
- B7: Short-wave infrared 2 (2.11 - 2.29 µm)
- B8: Panchromatic (0.50 - 0.68 µm)
- B9: Cirrus (1.36 - 1.38 µm)
- B10: Thermal Infrared 1 (10.60 - 11.19 µm)
- B11: Thermal Infrared 2 (11.50 - 12.51 µm)
- BQA: Data quality assessment band

The panchromatic band B8 has a spatial resolution of 15 meters per pixel, and all other bands have a spatial resolution of 30 meters per pixel, although the thermal bands B10 and B11 are generated from sensor data whose native resolution is 100 meters per pixel.

For more information, see [Landsat 8 Handbook](#)

Data availability (time)

Apr 11, 2013 - Nov 21, 2016

Provider

[USGS](#)

Tags

[landsat](#), [usgs](#), [l8](#), [lc8](#), [oli](#), [tirs](#), [radiance](#)

ImageCollection ID
LANDSAT/LC8_L1T

[Import](#)

Creating a composite of “raw” Landsat scenes:

1. Take the collection of the raw scenes
2. Apply TOA calibration
3. Assign a cloud score to each pixel
4. Select the lowest possible range of cloud scores at each point
5. Compute per-band percentile values from the accepted pixels.

Reminder:

Top of Atmosphere Reflectance (%)

Consider the angle and distance between
the sun, surface and satellite.

ee.Algorithms.Landsat.simpleComposite(raw image collection)

Google Earth Engine

Scripts Docs Assets

Filter methods...

ee.Algorithms

- ee.Algorithms.FMask
- ee.Algorithms.GeometryConstructors
- ee.Algorithms.Landsat
 - ee.Algorithms.Landsat.TOA(input)
 - ee.Algorithms.Landsat.calibratedRadiance(image)
 - ee.Algorithms.Landsat.pathRowLimit(collection, maxScenesPerPat...)
 - ee.Algorithms.Landsat.simpleCloudScore(image)**
 - ee.Algorithms.Landsat.simpleComposite(collection, percentile, clou...
 - ee.Algorithms.Landsat.surfaceReflectance(scene, precomputed, sur...
- ee.Algorithms.SAD
- ee.Algorithms.Test

New Script

Inspector Console Tasks

Use print(...) to write to this console.

ee.Algorithms.Landsat.simpleComposite(collection, percentile, cloudScoreRange, maxDepth, asFloat)

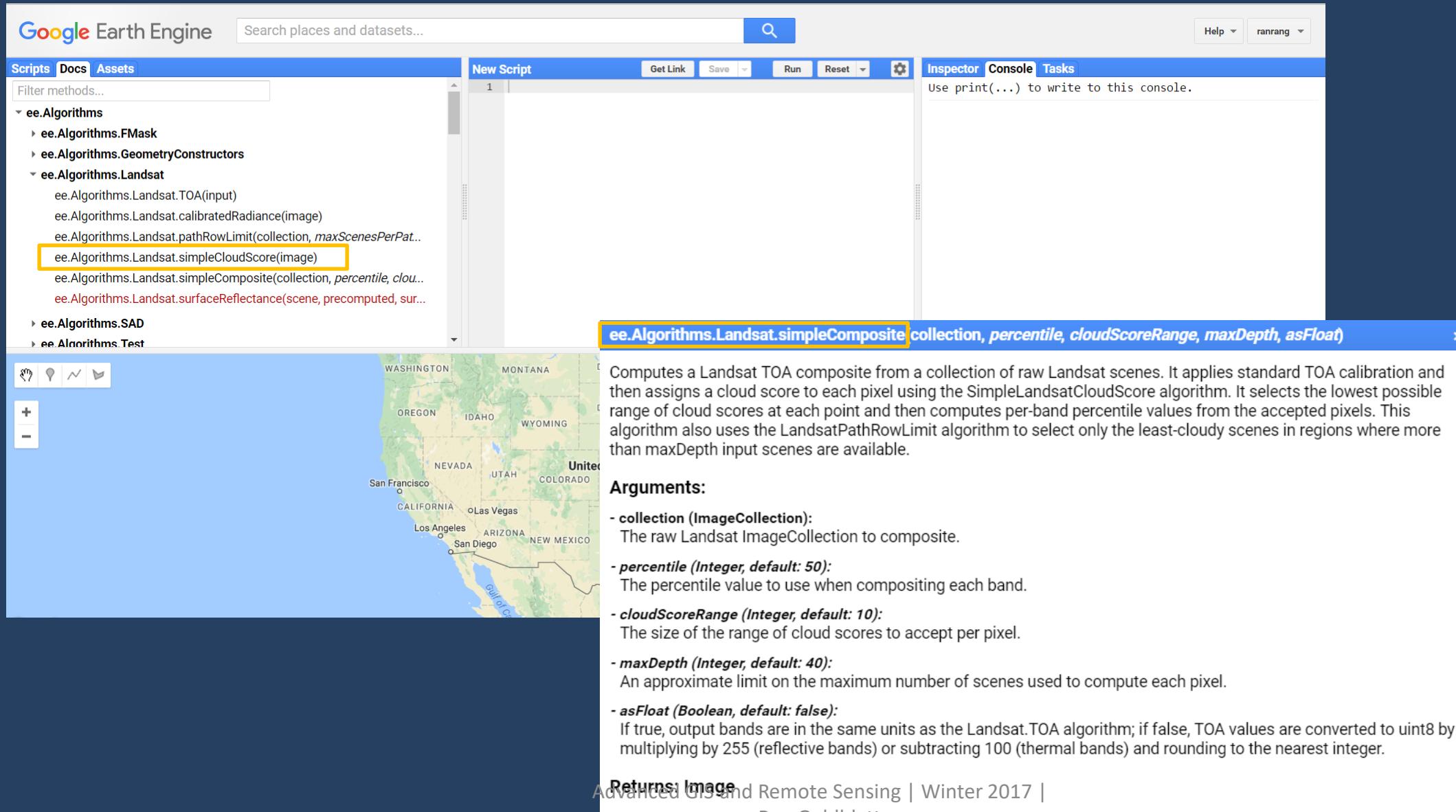
Computes a Landsat TOA composite from a collection of raw Landsat scenes. It applies standard TOA calibration and then assigns a cloud score to each pixel using the SimpleLandsatCloudScore algorithm. It selects the lowest possible range of cloud scores at each point and then computes per-band percentile values from the accepted pixels. This algorithm also uses the LandsatPathRowLimit algorithm to select only the least-cloudy scenes in regions where more than maxDepth input scenes are available.

Arguments:

- **collection (ImageCollection):**
The raw Landsat ImageCollection to composite.
- **percentile (Integer, default: 50):**
The percentile value to use when compositing each band.
- **cloudScoreRange (Integer, default: 10):**
The size of the range of cloud scores to accept per pixel.
- **maxDepth (Integer, default: 40):**
An approximate limit on the maximum number of scenes used to compute each pixel.
- **asFloat (Boolean, default: false):**
If true, output bands are in the same units as the Landsat.TOA algorithm; if false, TOA values are converted to uint8 by multiplying by 255 (reflective bands) or subtracting 100 (thermal bands) and rounding to the nearest integer.

Returns: Image

Advanced Image and Remote Sensing | Winter 2017 | Ran Goldblatt



Simple composite arguments:

```
var image = ee.Algorithms.Landsat.simpleComposite(  
{collection: //The raw Landsat ImageCollection to composite.  
percentile: //The percentile value to use when compositing each band. default: 50  
cloudScoreRange: // Range of cloud scores to accept per pixel. default: 10  
maxDepth: //Maximum number of scenes used to compute each pixel. default: 40  
asFloat: // Units of output bands (true: same as the Landsat.TOA algorithm; false:  
convert to uint8: multiply the reflective bands by 255, subtract 100 from the thermal  
bands, round to the nearest integer. default: false  
}  
)
```

Example:

```
var L8_Collection = ee.ImageCollection('LANDSAT/LC8_L1T');

var L8Filtered = L8_Collection.filterDate('2014-10-01', '2014-12-31');

var L8Median = L8Filtered.median();

Map.addLayer(L8Median, {'bands': ['B4', 'B3', 'B2'], min: 6000, max:17000}, 'landsat 8
median val');

var l8_Composite= ee.Algorithms.Landsat.simpleComposite({
  collection: L8Filtered,
  asFloat: true
});

Map.addLayer(l8_Composite , {'bands': ['B4', 'B3', 'B2'], min: 0, max:0.3}, 'landsat 8
simple composite');
```

Before....



And after (simple composite)....



NDVI

Google Earth Engine Search

Scripts Docs Assets Filter methods...

PLACES

RASTERS

- Landsat 7 Annual NDVI Composite
- Landsat 7 32-Day NDVI Composite (selected)
- Landsat 7 8-Day NDVI Composite
- Landsat 7 Annual Greenest-Pixel TOA Reflectance Composite

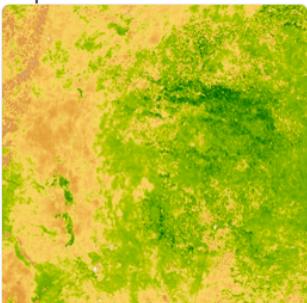
TABLES

Get Link Save Run Reset import » Inspector Console Tasks

Landsat 7 32-Day NDVI Composite

These Landsat 7 composites are made from Level L1T orthorectified scenes, using the computed top-of-atmosphere reflectance. The Normalized Difference Vegetation Index is generated from the Near-IR and Red bands of each scene, and ranges in value from -1.0 to 1.0.

These are most recent value composites created from all the scenes in each 32-day period beginning from the first day of the year and continuing to the 352nd day of the year. The last composite of the year, beginning on day 353, will overlap the first composite of the following year by 20 days. All the images from each 32-day period are included in the composite, with the most recent pixel as the composite value.

Sample 

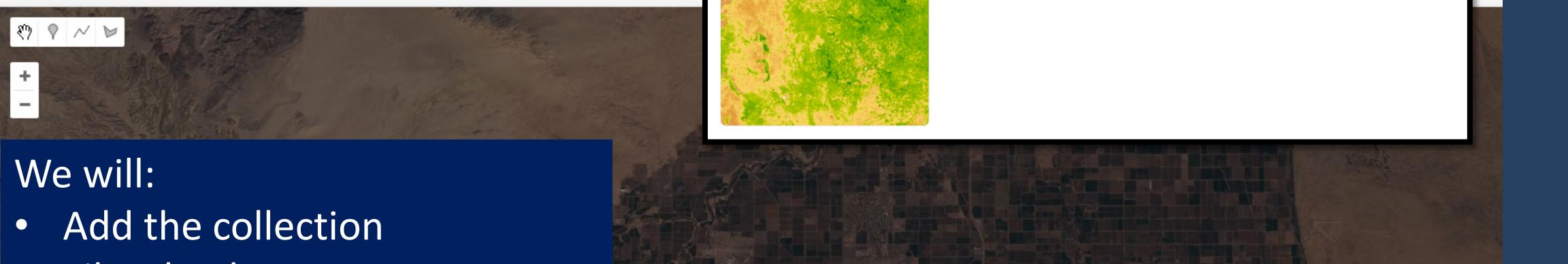
Data availability (time)
Jan 1, 1999 - Nov 17, 2015

Provider
Google

Tags
landsat, usgs, l7, le7, etm, 32day, ndvi

ImageCollection ID
LANDSAT/LE7_L1T_32DAY_NDVI

Open in workspace

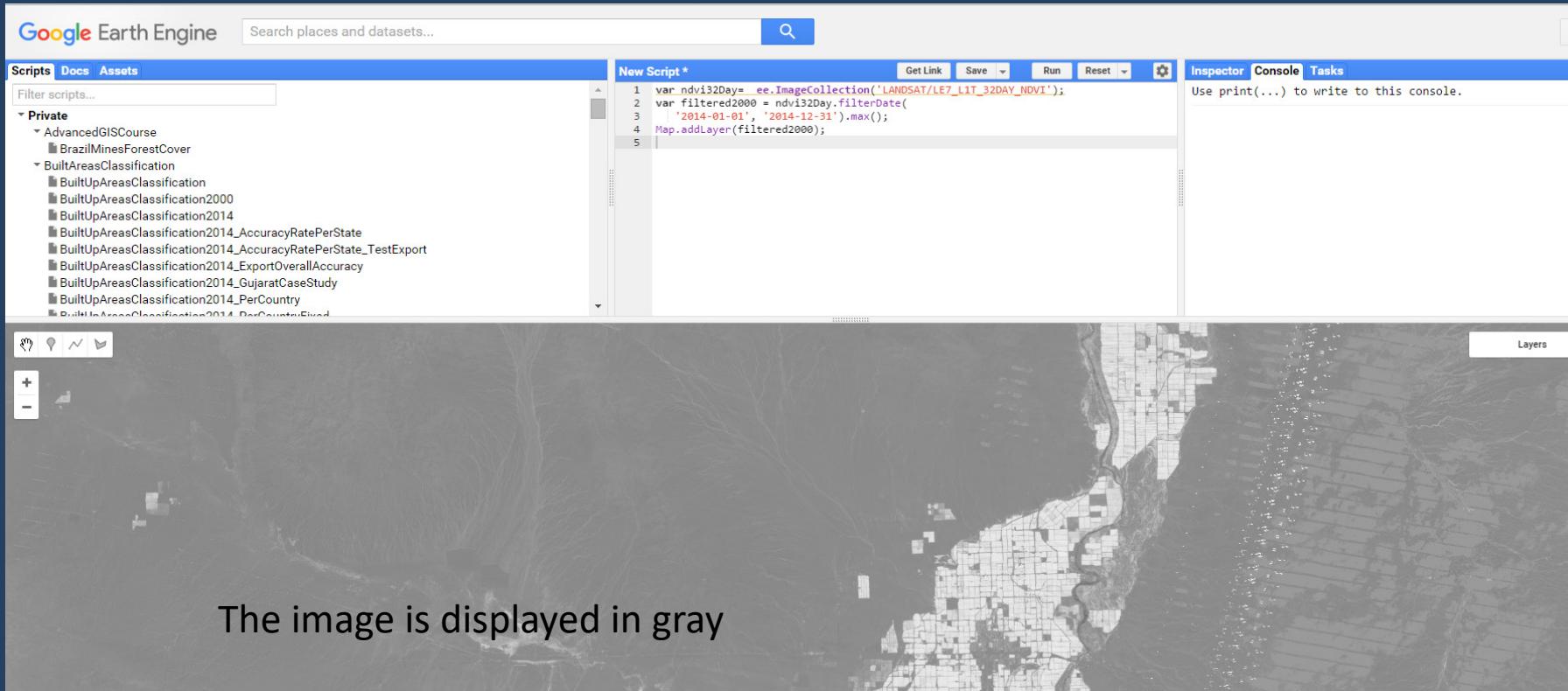


We will:

- Add the collection
- Filter by date
- Find the maximum ndvi value
- Visualize it.

LANDSAT/LE7_L1T_32DAY_NDVI

```
var ndvi32Day= ee.ImageCollection('LANDSAT/LE7_L1T_32DAY_NDVI');
var filtered2000 = ndvi32Day
.filterDate('2014-01-01', '2014-12-31').max();
Map.addLayer(filtered2000);
```



The image is displayed in gray

Palettes

- Palettes define the color scheme an image.
- A palette is a comma delimited string of colors, stretched over a band's maximum and minimum pixel values.

The string '000000' is black

'FFFFFF' is white

'FF0000' is red

'00FF00' is green

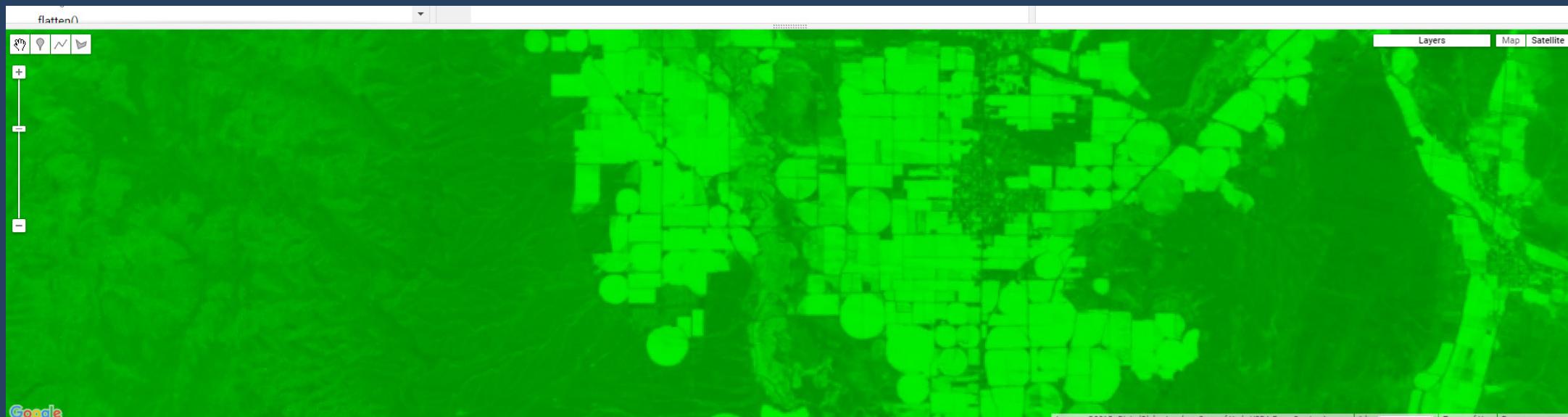
'0000FF' is blue.

http://www.w3schools.com/colors/colors_names.asp

We need to add a *visualization parameter* (just as we did with 'bands') that will specify a palette

```
var collection = ee.ImageCollection('LANDSAT/LE7_L1T_32DAY_NDVI');
var filtered2000 =
collection.filterDate('2014-01-01', '2014-12-31').max();
Map.addLayer(filtered2000, {'palette':'000000, 00FF00'});
```

000000 00FF00



Why is the entire image green?

-1

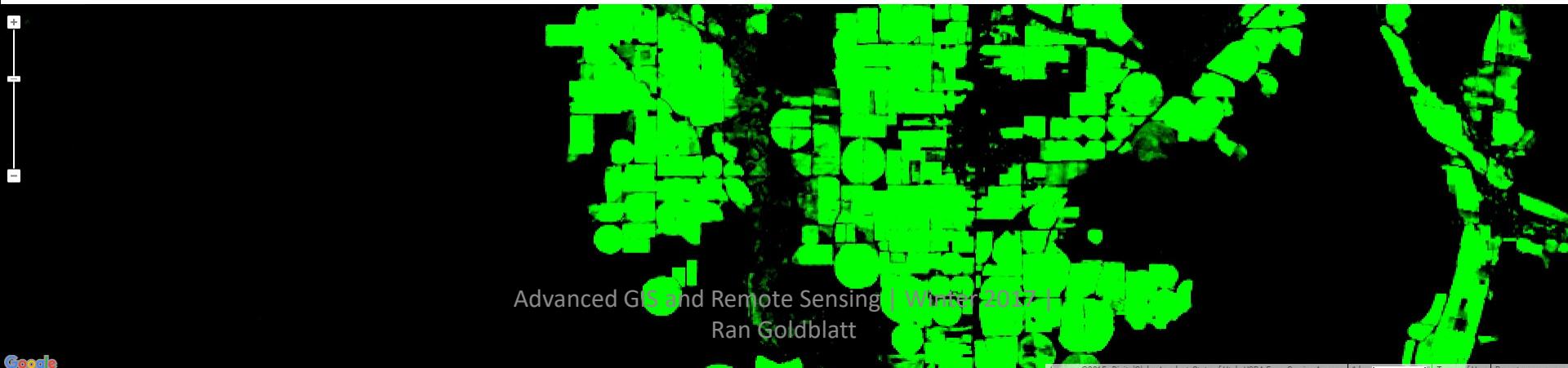
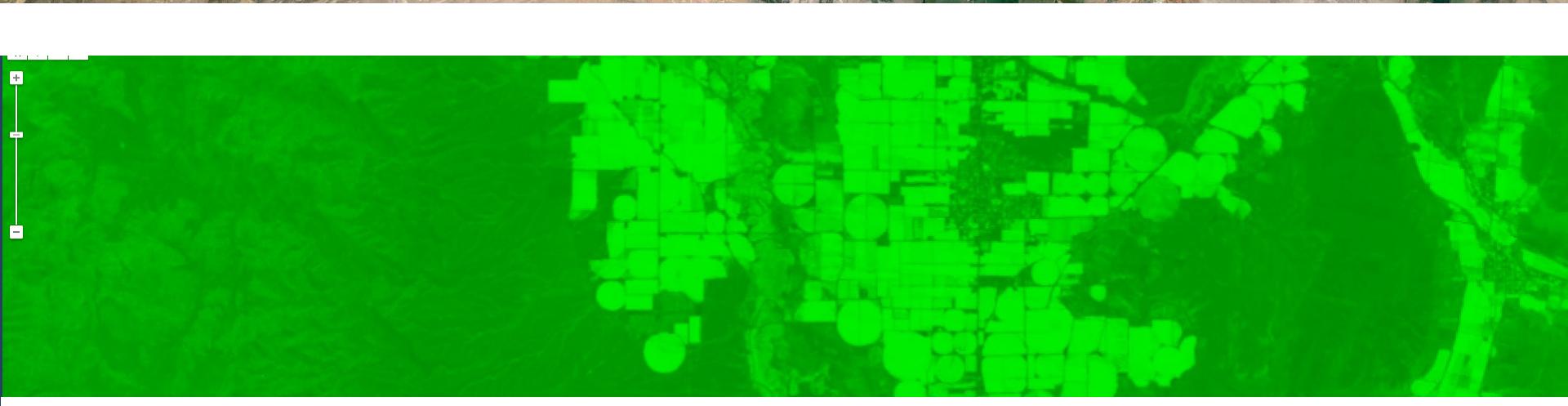
+1

```
var collection = ee.ImageCollection('LANDSAT/LE7_L1T_32DAY_NDVI');
var filtered2000 = collection.filterDate('2014-01-01', '2014-12-31').max();
Map.addLayer(filtered2000,
{'palette':'000000, 00FF00', 'min': 0.5, 'max': 0.7}
);
```

The screenshot shows the Google Earth Engine (GEE) web application interface. The top navigation bar includes 'Google Earth Engine', a search bar, and a user dropdown. The left sidebar has tabs for 'Scripts', 'Docs', and 'Assets'. Under 'Assets', there is a 'Private' folder containing various scripts and datasets, such as 'AdvancedGISCourse', 'BrazilMinesForestCover', and several 'BuiltUpAreasClassification' entries. The main workspace is titled 'New Script *' and contains the following code:

```
1 var collection = ee.ImageCollection('LANDSAT/LE7_L1T_32DAY_NDVI');
2 var filtered2000 = collection.filterDate('2014-01-01', '2014-12-31').max();
3 Map.addLayer(filtered2000,
4 {'palette':'000000, 00FF00', 'min': 0.5, 'max': 0.7}
5 );
```

The right side of the interface features three tabs: 'Inspector', 'Console', and 'Tasks'. The 'Console' tab contains the instruction 'Use print(...) to write to this console.' Below the workspace is a map of India with a green overlay representing the NDVI data. The bottom of the screen displays the footer text: 'Advanced GIS and Remote Sensing | Winter 2017 | Ran Goldblatt'.



Advanced GIS and Remote Sensing | Winter 2017
Ran Goldblatt

DEMO

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

```
print('hello class!');
```

1. Navigate to: El centro, California
2. Search for: landsat 7 annual TOA Percentile composites
3. Copy the ID of the image collection: LANDSAT/LE7_TOA_1YEAR
4. Call this image collection and print it to the console:

```
var landsat7TOA = ee.ImageCollection('LANDSAT/LE7_TOA_1YEAR');
print(landsat7TOA);
```

5. Copy the image ID of the 2014 image.
Call it, print it, and add it to the map:

```
var landsat7TOA2014 = ee.Image('LANDSAT/LE7_TOA_1YEAR/2014');
print(landsat7TOA2014);
Map.addLayer(landsat7TOA2014);
```

Define the bands for the visualization:

```
var landsat7TOA2014 = ee.Image('LANDSAT/LE7_TOA_1YEAR/2014');  
Map.addLayer(landsat7TOA2014,  
{"bands": ['B3', 'B2', 'B1']}  
);
```

1. Search for: Landsat 8 32-Day TOA Reflectance Composite.
2. Copy the ID of the collection (LANDSAT/LC8_L1T_32DAY_TOA)
3. Filter by date to the second half of 2014, and add this image to the map

```
var landsat8_32Days =  
ee.ImageCollection('LANDSAT/LC8_L1T_32DAY_TOA');
```

```
var landsat8_SecondHalf14 = landsat8_32Days  
.filterDate('2014-06-01', '2014-12-31');
```

```
Map.addLayer(landsat8_SecondHalf14,  
{'bands': ['B4', 'B3', 'B2']}, 'landsat 8 at 2014');
```

1. Define a variable for Band 2 (Blue)
2. Add Band 2 to the map

```
var landsat8_32Days =  
ee.ImageCollection('LANDSAT/LC8_L1T_32DAY_TOA');
```

```
var landsat8_SecondHalf14 = landsat8_32Days  
.filterDate('2014-06-01', '2014-12-31');
```

```
Map.addLayer(landsat8_SecondHalf14,  
{'bands': ['B4', 'B3', 'B2']}, 'landsat 8 at 2014');
```

```
var L8_Band2 = landsat8_SecondHalf14.select('B2');  
Map.addLayer(L8_Band2,{}, 'band 2');
```

Calculate the median value of this collection and add it to the map

```
var L8_32Days =  
ee.ImageCollection('LANDSAT/LC8_L1T_32DAY_TOA');  
  
var L8_SecondHalf14 = L8_32Days  
.filterDate('2014-06-01', '2014-12-31')  
  
var L8_SecondHalf14Median = L8_SecondHalf14.median();  
  
Map.addLayer(L8_SecondHalf14, {'bands': ['B4', 'B3', 'B2']} ,  
'landsat 8 ');  
  
Map.addLayer(L8_SecondHalf14Median, {'bands': ['B4', 'B3', 'B2']} ,  
'landsat 8 median' );
```

Simple composite of Landsat scenes

```
var L8_Collection = ee.ImageCollection('LANDSAT/LC8_L1T');

var L8Filtered = L8_Collection.filterDate('2014-10-01', '2014-12-31');

var L8Median = L8Filtered.median();

Map.addLayer(L8Median, {'bands': ['B4', 'B3', 'B2'], min: 6000, max:17000});

var image = ee.Algorithms.Landsat.simpleComposite({
  collection: L8Filtered,
  asFloat: true
});

Map.addLayer(image , {'bands': ['B4', 'B3', 'B2'], min: 0, max:0.3}, 'landsat 8
Median Witner 2014' );
```

1. Search for the ID of Landsat 7 32-Day NDVI Composite
LANDSAT/LE7_L1T_32DAY_NDVI
2. Filter by date (2014 and 2000) and extract the max NDVI value

```
var ndvi32Day= ee.ImageCollection('LANDSAT/LE7_L1T_32DAY_NDVI');

var filtered2014 = ndvi32Day.filterDate('2014-01-01', '2014-12-31').max();
var filtered2000 = ndvi32Day.filterDate('2000-01-01', '2000-12-31').max();

Map.addLayer(filtered2014, {}, 'max ndvi 2014');
Map.addLayer(filtered2000, {}, 'max ndvi 2000');
```

Add a visualization parameter:

```
var ndvi32Day= ee.ImageCollection('LANDSAT/LE7_L1T_32DAY_NDVI');

var filtered2014 = ndvi32Day.filterDate('2014-01-01', '2014-12-31').max();
var filtered2000 = ndvi32Day.filterDate('2000-01-01', '2000-12-31').max();

Map.addLayer(filtered2014, {'palette':'000000, 00FF00'}, 'max ndvi 2014');
Map.addLayer(filtered2000, {'palette':'000000, 00FF00'}, 'max ndvi 2000');
```

Define the minimum and maximum value for visualization

```
var ndvi32Day= ee.ImageCollection('LANDSAT/LE7_L1T_32DAY_NDVI');
```

```
var filtered2014 = ndvi32Day.filterDate('2014-01-01', '2014-12-31').max();
```

```
var filtered2000 = ndvi32Day.filterDate('2000-01-01', '2000-12-31').max();
```

```
Map.addLayer(filtered2014,
```

```
{'palette':'000000, 00FF00', 'min': 0.5, 'max': 0.7}, 'max ndvi 2014');
```

```
Map.addLayer(filtered2000,
```

```
{'palette':'000000, 00FF00', 'min': 0.5, 'max': 0.7}, 'max ndvi 2000');
```

Exercise:

Navigate to any region and present 2 Landsat images for the rainy season of 2013 and one for 2016, using the median value of the composite