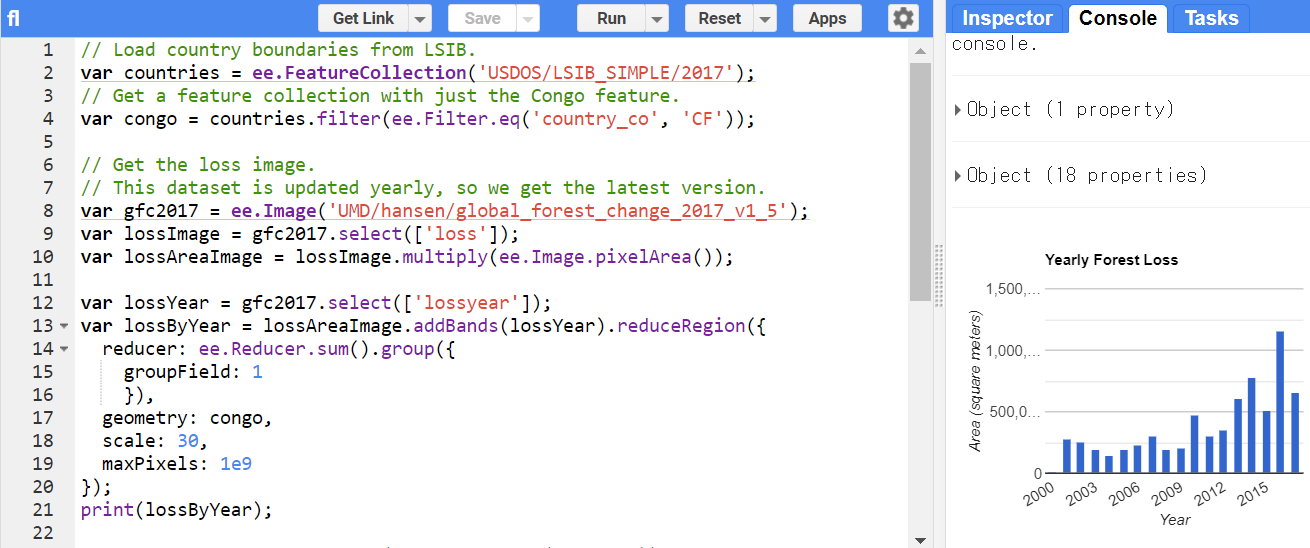
Calculating Yearly Forest Loss and Charting Yearly Forest Loss.



// Load country boundaries from LSIB.

var countries = ee.FeatureCollection('USDOS/LSIB\_SIMPLE/2017');

// Get a feature collection with just the Congo feature.

var congo = countries.filter(ee.Filter.eq('country\_co', 'CF'));

// Get the loss image.

// This dataset is updated yearly, so we get the latest version.

var gfc2017 = ee.Image('UMD/hansen/global\_forest\_change\_2017\_v1\_5');

var lossImage = gfc2017.select(['loss']);

var lossAreaImage = lossImage.multiply(ee.Image.pixelArea());

var lossYear = gfc2017.select(['lossyear']);

var lossByYear = lossAreaImage.addBands(lossYear).reduceRegion({

reducer: ee.Reducer.sum().group({

groupField: 1

}),

geometry: congo,

scale: 30,

maxPixels: 1e9

});

print(lossByYear);

var statsFormatted = ee.List(lossByYear.get('groups'))

.map(function(el) {

var d = ee.Dictionary(el);

return [ee.Number(d.get('group')).format("20%02d"), d.get('sum')];

});

var statsDictionary = ee.Dictionary(statsFormatted.flatten());

print(statsDictionary);

var chart = ui.Chart.array.values({

array: statsDictionary.values(),

axis: 0,

xLabels: statsDictionary.keys()

}).setChartType('ColumnChart')

.setOptions({

title: 'Yearly Forest Loss',

hAxis: {title: 'Year', format: '####'},

vAxis: {title: 'Area (square meters)'},

legend: { position: "none" },

lineWidth: 1,

pointSize: 3

});

print(chart);