```
// Load Sentinel-1 C-band SAR Ground Range collection (log scale, VV polarization, descending pass,
Swath Mode: IW)
var collectionVV = ee.ImageCollection('COPERNICUS/S1_GRD')
.filter(ee.Filter.eq('instrumentMode', 'IW'))
.filter(ee.Filter.listContains('transmitterReceiverPolarisation', 'VV'))
.filter(ee.Filter.eq('orbitProperties_pass', 'DESCENDING'))
.filterMetadata('resolution_meters', 'equals' , 10)
.filterBounds(roi)
.select('VV');
print(collectionVV, 'Collection VV');

// Load Sentinel-1 C-band SAR Ground Range collection (log scale, VH polarization, descending pass,
Swath Mode: IW))
var collectionVH = ee.ImageCollection('COPERNICUS/S1_GRD')
.filter(ee.Filter.eq('instrumentMode', 'IW'))
.filter(ee.Filter.listContains('transmitterReceiverPolarisation', 'VH'))
.filter(ee.Filter.eq('orbitProperties_pass', 'DESCENDING'))
.filterMetadata('resolution_meters', 'equals' , 10)
.filterBounds(roi)
.select('VH');
print(collectionVH, 'Collection VH');

//Filter by date
var first2016VV = collectionVV.filterDate('2016-08-01', '2016-08-10').mosaic();
var second2018VV = collectionVV.filterDate('2018-08-05', '2018-08-10').mosaic();
var third2019VV = collectionVV.filterDate('2019-08-01', '2019-08-15').mosaic();
var first2016VH = collectionVH.filterDate('2016-08-01', '2016-08-10').mosaic();
var second2018VH = collectionVH.filterDate('2018-08-05', '2018-08-10').mosaic();
var third2019VH = collectionVH.filterDate('2019-08-01', '2019-08-15').mosaic();
```

```
// Display map
Map.centerObject(roi,  7);
Map.addLayer(first2016VV,  {min:-15,max:0}, '2016 VV', 0);
Map.addLayer(second2018VV,  {min:-15,max:0}, '2018 VV', 0);
Map.addLayer(third2019VV,  {min:-15,max:0}, '2019 VV', 0);
Map.addLayer(first2016VH,  {min:-27,max:0}, '2016 VH', 0);
Map.addLayer(second2018VH,  {min:-27,max:0}, '2018 VH', 0);
Map.addLayer(third2019VH,  {min:-27,max:0}, '2019 VH', 0);
//RGB Combination r:2016 g: 2018 b: 2019
//green: High in 2018 and low in 2016 and 2019
//red: High in 2016
//Yellow: High in 2016 2018
Map.addLayer(first2016VH.addBands(second2018VH).addBands(third2019VH),  {min: -27,max: -8}, '2016/2018/2019  RGB', 0);
//Apply filter to reduce speckle  -- smoothy filter
var SMOOTHING_RADIUS = 50;
var first2016VV_filtered  = first2016VV.focal_mean(SMOOTHING_RADIUS,  'circle', 'meters');
var first2016VH_filtered  = first2016VH.focal_mean(SMOOTHING_RADIUS,  'circle', 'meters');
var second2018VV_filtered  = second2018VV.focal_mean(SMOOTHING_RADIUS,  'circle','meters');
var second2018VH_filtered  = second2018VH.focal_mean(SMOOTHING_RADIUS,  'circle','meters');
var third2019VV_filtered  = third2019VV.focal_mean(SMOOTHING_RADIUS,  'circle', 'meters');
var third2019VH_filtered  = third2019VH.focal_mean(SMOOTHING_RADIUS,  'circle', 'meters');
//Display filtered images  --- stretching the image (from-15 to 0)
Map.addLayer(first2016VV_filtered,  {min:-15,max:0}, '2016 VV Filtered',0);
Map.addLayer(first2016VH_filtered,  {min:-27,max:0}, '2016 VH Filtered',0);
Map.addLayer(second2018VV_filtered,  {min:-15,max:0}, '2018 VV Filtered',0);
Map.addLayer(second2018VH_filtered,  {min:-27,max:0}, '2018 VH Filtered',0);
Map.addLayer(third2019VV_filtered,  {min:-15,max:0}, '2019 VV Filtered',0);
Map.addLayer(third2019VH_filtered, {min:-27,max:0}, '2019 VH Filtered',0);
Map.addLayer(first2016VH_filtered.addBands(second2018VH_filtered).addBands(third2019VH_filtered)
, {min: -25, max: -8}, '2016/2018/2019 HV filtered RGB', 0);
// Calculate the difference between before and after
var diff1618VH= first2016VH_filtered.subtract(second2018VH_filtered);
var diff1618VV= first2016VV_filtered.subtract(second2018VV_filtered);
var diff1819VH= second2018VH_filtered.subtract(third2019VH_filtered);
var diff1819VV= second2018VV_filtered.subtract(third2019VV_filtered);
// Display the difference images
Map.addLayer(diff1618VH, {min: -9,max:9}, 'Difference VH 2016/2018', 0);
Map.addLayer(diff1618VV, {min: -9,max:9}, 'Difference VV 2016/2018', 0);
Map.addLayer(diff1819VH, {min: -9,max:9}, 'Difference VH 2018/2019', 0);
Map.addLayer(diff1819VV, {min: -9,max:9}, 'Difference VV 2018/2019', 0);
//Calculate histograms for each image
print(ui.Chart.image.histogram({image:diff1618VH, region:newroi, scale:300}));
print(ui.Chart.image.histogram({image:diff1819VH, region:newroi, scale:300}));
// Combine the mean and standard deviation reducers.
var reducers = ee.Reducer.mean().combine({
reducer2: ee.Reducer.stdDev(),
sharedInputs: true
});

//Calculate the mean and standard deviation for each difference image
var stats1618 = diff1618VH.reduceRegion({
reducer: reducers,
geometry: newroi,
scale: 10,
});
var stats1819 = diff1819VH.reduceRegion({
```

```
reducer: reducers,
geometry: newroi,
scale: 10,
});
//Print the mean and stdv for each difference image
print('stats:', stats1618, stats1819)
//Apply Thresholds based on ND <stdvx1.5 to create a vegetation regrowth mask
// recalculating the Thresholds because the ROI was adjusted. Multiplying ((1.296* 1.5) + 0.24). ((stdv * threshold) + mean)
var DIFF_UPPER_THRESHOLD1618 = 2.17;
var DIFF_UPPER_THRESHOLD1819 = 2.88;
var diff1618VH_thresholded = diff1618VH.gt(DIFF_UPPER_THRESHOLD1618);
var diff1819VH_thresholded = diff1819VH.gt(DIFF_UPPER_THRESHOLD1819);
//Display Masks based on threshold of previous step.
//anything above of the threshold set as one, that repesent vegetation lost.
// set the colors in order to check the difference among the years
// the threshold is only showing where the vegetation was lost. No were the vegetation was recover,
// to do that the threshold should be changed.
Map.addLayer(diff1618VH_thresholded.updateMask(diff1618VH_thresholded),{palette:"FF0000"},'Vegetation Loss 16/18',1);
Map.addLayer(diff1819VH_thresholded.updateMask(diff1819VH_thresholded),{palette:"FF0000"},'Vegetation Loss 18/19',1);
//Compare differences in vegetation loss between 16/18 and 18/19 in an small area in statistics.
var area_loss1618 = diff1618VH_thresholded.reduceRegion({
 reducer: ee.Reducer.sum(),
 geometry: stats_area,
 scale: 10,
 });
var area_loss1819 = diff1819VH_thresholded.reduceRegion({
 reducer: ee.Reducer.sum(),
 geometry: stats_area,
 scale: 10,
});
//Print the mean and stdv for each difference of image calculated in previous step
// it calculates the number of pixels were there was lost of vegetation 2016-2018
// in 2016-2018 983 pixels with less vegetation
// in 2018-2019 433pixels with less vegetation
print('stats:', area_loss1618, area_loss1819);

//The area of a pixel is 10 meters (which is 10 m × 10 m = 100 square meters or 0.01 ha
//983 pixels x 0.01 = 9.83 ha in 2016/2018
//433 pixels x 0.01 = 4.33 ha in 2018/2019

// Summary: Vegetation Lost using a threshold and mask to represent where was vegatation lost.
// This vegation lost does not mean necessry DEFORESTATION. Can be where vegetation degradation
//from one year to another. The calculated here is the vegetation loss



// Export Image
Export.image.toDrive({
 image: diff1618VH_thresholded,
 description: '2016_2018_mask',
 scale: 30,
 region: newroi,
  fileFormat: 'GeoTIFF',
 folder: 'GEE'
});
```