//2022.01.28 อาจไม่ต้องสอนก็ได้

// emm lab 3 การประมวลผลข้อมูลปริมาณมากด้วยการลดจำนวนข้อมูลจาก NDVI และ median reducer

// จากลิ้งค์ https://www.gears-lab.com/emm\_lab\_3/

// จากยูทูป https://www.youtube.com/watch?v=Wik07UyjnDs&list=PLf6lu3bePWHCKAoGx0iDYElctyw3g07ge&index=3

//1. ให้ซูมไปภาคตะวันออกของประเทศไทย ระยอง

var roi = ee.Geometry.Point([101.26251, 12.68957]);

//2. นำเข้าข้อมูล Landsat 8 TOA ด้วยการค้นหาในช่องค้นหา แล้วเปลี่ยนชื่อตัวแปรเป็น L8

var L8=ee.ImageCollection("LANDSAT/LC08/C01/T1\_TOA");

Map.centerObject(roi,9);

//3. ก๊อปปี้โค้ดเพื่อกรองข้อมูลภาพแลนแซท 8

var filtered = L8.filterDate('2020-08-01', '2020-08-10');

Map.addLayer(filtered);

// กด run แล้วให้ซูมออก เพื่อดูภาพรวมว่า ช่วงเวลาดังกล่าวจะพบเมฆส่วนใหญ่ เนื่องจากเป็นฤดูมรสุม

// สังเกตว่าภาพจะเป็นสีดำ

/\*

//4.เล่นกับแบนด์คอมโพสิท ด้วยการผสมภาพสีจริงของแลนด์แซท 8 นั่นคือ RGB 432

// ให้ใช้การคอมเม้นแบบหลายบรรทัด /กับข้อ 3

var filtered = L8.filterDate('2020-08-01', '2020-08-10');

Map.addLayer(filtered, {min: 0, max: 0.3, bands:['B4', 'B3', 'B2']});

// กด run จะสังเกตได้ว่าภาพที่ได้เป็นภาพสีผสมจริงที่มีความสว่างกว่าข้อ 3

// อย่างไรก็ตามจำนวนภาพยังมีน้อยเกินไป เพราะช่วงเวลาที่ใช้กรองภาพนั้้นสั้นเกิน แลนด์แซทจะมี revisit time อยู่ที่ 16 วัน

//5.ทำภาพสีผสมเท็จที่แสดงพืชพรรณเป็นสีแดงด้วยการใส่แบนด์ nir ในแม่สีแดง

var filtered = L8.filterDate('2020-08-01', '2020-08-10');

Map.addLayer(filtered, {min: 0, max :0.3, bands:['B4', 'B3', 'B2']}, 'RGB');

Map.addLayer(filtered, {min: 0, max: 0.3, bands:['B5', 'B4', 'B3']}, 'False Color');

// กด run

// สังเกตภาพสีผสมเท็จ สีแดงจะแทนบริเวณที่เป็นพืชพรรณในการผสมครั้งนี้

//6.ค่าพารามิเตอร์ในการตั้งค่าเพื่อการปรับแต่งแผนที่ สามารถสร้างเป็นฟังค์ชันได้ เมื่อถูกใช้บ่อยๆ

var rgb\_vis = {min: 0, max: 0.3, bands:['B4', 'B3', 'B2']};

var filtered = L8.filterDate('2020-09-01', '2020-09-10');

Map.addLayer(filtered, rgb\_vis, 'RGB');

// กด run

//////////////////////////////////////////////////////////////////////////////

// ส่วนต่อไปจะเป็นการสาธิตหลักการ reducing and collections เพื่อนำมาใช้กับภาพปริมาณมากๆ

// 7.แสดงภาพที่บันทึกในรอบ 1 ปีของพื้นที่ที่สนใจ

var rgb\_vis = {min: 0, max: 0.3, bands: ['B4', 'B3', 'B2']};

var filtered = L8.filterDate('2020-01-01', '2021-01-01');

Map.addLayer(filtered, rgb\_vis, 'RGB');

// กด run แล้วซูมออก จะพบว่าในรอบ 1 ปีมีการบันทึกภาพแลนแซทที่เยอะมาก

// อย่างไรก็ตามบริเวณภาคใต้ของประเทศไทย เต็มไปด้วยเมฆปกคลุม

// เราจะใช้วิธีการ median เพื่อลดปริมาณภาพจำนวนมากเหล่านี้

//8.จะใช้วิธีการ median เพื่อลดปริมาณภาพจำนวนมากเหล่านี้

var rgb\_vis = {min: 0, max: 0.3, bands: ['B4', 'B3', 'B2']};

var filtered = L8.filterDate('2017-01-01', '2021-01-01');

Map.addLayer(filtered.median(), rgb\_vis, 'RGB - median reducer'); // ใช้ median

// กด run สังเกตว่าภูเก็ตเมื่อใช้ช่วงเวลา 1 ปีจะยังมีเมฆเต็มพื้นที่ แต่หากเปลี่ยนเป็น 2017-2021 จะลดเมฆไปได้

// นิสิตลอง pan ไปดูบริเวณอื่นหน่อย แล้วลองเล่นกับช่วงเวลาที่ใช้กรองข้อมูล

//9. การแยกภาพที่ต้องการ เพื่อนำมาเป็นชั้นข้อมูลเดี่ยวๆใน layers ให้ซุมชลบุรี

var rgb\_vis = {min: 0, max: 0.3, bands: ['B4', 'B3', 'B2']};

var filtered = L8.filterDate('2020-02-01', '2020-02-28');

Map.addLayer(filtered, rgb\_vis, 'RGB');

// กด run

//9.1 ค้นหาภาพด้วยการใช้ roi และเพิ่มเงื่อนไขการค้นหาเข้าไป ใช้บอลลูนในการเลือกตำแหน่ง

//ผลลัพธ์ที่ได้จากการค้นหาจะถูกเก็บไว้ใน image collections ซึ่งจะมีภาพเดียวหรือหลายภาพก็ได้ขึั้นอยู่กับระยะเวลาที่ใช้กรอง

var filtered = L8.filterDate('2020-02-01', '2020-02-28')

.filterBounds(roi);

var rgb\_vis = {min: 0, max: 0.3, bands: ['B4', 'B3', 'B2']};

Map.addLayer(filtered, rgb\_vis, 'RGB');

//9.2 เลือกภาพตัวอย่างจาก image collections เพื่อนำมาแสดงเป็นชั้นข้อมูลเดี่ยวใน Layers

var filtered = L8.filterDate('2020-02-01', '2020-02-28')

.filterBounds(roi);

var image = ee.Image(filtered.first()); //เลือกภาพแรกที่อยู่ใน image collections

var rgb\_vis = {min: 0, max: 0.3, bands: ['B4', 'B3', 'B2']};

Map.addLayer(image, rgb\_vis, 'RGB');

// 10. คำนวณดัชนีพืชพรรณ ด้วยการใช้ e.Image.normalizedDifference() method

var rgb\_vis = {min: 0, max: 0.3, bands: ['B4', 'B3', 'B2']};

var filtered = L8.filterDate('2020-02-01', '2020-02-28')

.filterBounds(roi);

var image = ee.Image(filtered.first());

var ndvi = image.normalizedDifference(['B5', 'B4']);

Map.addLayer(image, rgb\_vis, 'RGB');

Map.addLayer(ndvi, {min: 0, max: 1}, 'NDVI');

// กด run สังเกตว่าภาพที่ใช้จะมีเมฆ ซึ่งการ mask เมฆ สามารถทำได้ จะได้กล่าวภายหลัง

// หรือการเลือกเฉพาะภาพที่มีเปอร์เซ็นต์เมฆที่น้อยที่สุด มาทำ ndvi

//11. คำนวณ NDVI ด้วยการสร้างฟังค์ชัน

function addNDVI(image) {

var ndvi = image.normalizedDifference(['B5', 'B4']);

return image.addBands(ndvi);

}

// เรียกใช้ฟังค์ชัน

var rgb\_vis = {min: 0, max: 0.3, bands: ['B4', 'B3', 'B2']};

var filtered = L8.filterDate('2020-02-01', '2020-02-28')

.filterBounds(roi)

var image = ee.Image(filtered.first());

var ndvi = addNDVI(image);

Map.addLayer(image, rgb\_vis, 'RGB');

Map.addLayer(ndvi, {min: 0, max: 1}, 'NDVI');

// กด run แล้วใช้ inspector เลือกไปสักหนึ้่งพื้นที่ จะพบว่า ndvi มีการเพิ่มแบนด์ nd ที่เป็นผลลัพธ์ของการคำนวณ ndvi

// อย่างไรก็ตามแบนด์ 1 ถูกนำมาแสดงโดยอัตโนมัติ ดังนั้นจึงต้องมีการกำหนดค่าพารามิเตอร์เพื่อแสดงผล ndvi ด้วย nd

Map.addLayer(ndvi, {bands: 'nd', min: 0, max: 1}, 'NDVI\_adjusted');

//กด run แล้วจะพบว่ามีการแสดง ndvi ที่ถูกต้องจากการใช้แบนด์ nd

//12.การใช้ map function สำหรับ image collections

//คือการเรียกใช้ฟังค์ชันกับทุกๆภาพที่ถูกเก็บไว้ในชุด image collections

function addNDVI(image) {

var ndvi = image.normalizedDifference(['B5', 'B4']);

return image.addBands(ndvi);

}

var filtered = L8.filterDate('2020-02-01', '2020-02-28')

.filterBounds(roi);

var with\_ndvi = filtered.map(addNDVI);

var rgb\_vis = {min: 0, max: 0.3, bands: ['B4', 'B3', 'B2']};

Map.addLayer(filtered, rgb\_vis, 'RGB');

Map.addLayer(with\_ndvi, {bands: 'nd', min: 0, max: 1}, 'NDVI');

// กด run

//12.1 การใช้ median เพื่อหาค่า ndvi ของหลายช่วงเวลา

function addNDVI(image) {

var ndvi = image.normalizedDifference(['B5', 'B4']);

return image.addBands(ndvi);

}

var filtered = L8.filterDate('2020-01-01', '2020-12-31');

var with\_ndvi = filtered.map(addNDVI);//ใช้ map ฟังค์ชันเพื่อสร้าง ndvi ในแต่ละภาพที่เก็บไว้ใน image collection

var rgb\_vis = {min: 0, max: 0.3, bands: ['B4', 'B3', 'B2']};

Map.addLayer(filtered.median(), rgb\_vis, 'RGB');//ใช้ median เพื่อหาค่ามัธยฐานของจุดภาพหลายช่วงเวลา

Map.addLayer(with\_ndvi.median(), {bands: 'nd', min: 0, max: 1}, 'NDVI');//หาค่า median ของ ndvi ในแต่ละภาพที่เก็บไว้ใน image collection

// กด run

//13.การสร้างแผนที่โมเสคด้วยการใช้ greeness pixel ซึ่งใช้ ndvi ทำให้ได้แผนที่โมเสคที่ไม่มีเมฆ นอกเหนือจากวิธ๊ median ในข้อ 12.

function addNDVI(image) {

var ndvi = image.normalizedDifference(['B5', 'B4']);

return image.addBands(ndvi);

}

var filtered = L8.filterDate('2020-01-01', '2020-12-31');

var with\_ndvi = filtered.map(addNDVI);

var rgb\_vis = {min: 0, max: 0.3, bands: ['B4', 'B3', 'B2']};

Map.addLayer(filtered.median(), rgb\_vis, 'RGB (median)');

var greenest = with\_ndvi.qualityMosaic('nd');// ตรงนี้ใช้ qualityMosaic จะเรียกเอาค่าสูงสุดมาโมเสค ในที่นี้ใช้แบนด nd

Map.addLayer(greenest, rgb\_vis, 'RGB (greenest pixel)');

// กด run สังเกตว่าในท้องทะเลจะมีเมฆเต็ม ส่วนแผ่นดินจะไร้เมฆหรือมีเมฆน้อยมากเมื่อเทียบกับวิธีการ median

// 14. สร้างกราฟเพื่อวิเคราะห์NDVIแบบ time series

function addNDVI(image) {

var ndvi = image.normalizedDifference(['B5', 'B4']);

return image.addBands(ndvi);

}

var filtered = L8.filterDate('2020-01-01', '2020-12-31');

var with\_ndvi = filtered.map(addNDVI);

var greenest = with\_ndvi.qualityMosaic('nd');

var rgb\_vis = {min: 0, max: 0.3, bands: ['B4', 'B3', 'B2']};

Map.addLayer(filtered.median(), rgb\_vis, 'RGB (median)');

Map.addLayer(greenest, rgb\_vis, 'RGB (greenest pixel)');

print(Chart.image.series(with\_ndvi.select('nd'), roi));

// กด run เพื่อดูกราฟของค่า ndvi ที่ได้จากแบนด์ nd ในตำแหน่งของ roi ที่แสดงด้วยจุดบอลลูน

// หาก roi ไปเลือกในพื้นที่ที่ไม่ใช่พืชพรรณ ค่า ndvi จะไม่เปลี่ยนแปลงเช่นในเขตเมือง

// ค่า ndvi จะสูงในช่วงฤดูเพาะปลูก ส่วนฤดูแล้งจะต่ำ

// ค่า ndvi นี้ใช้วิเคราะห์ช่วงเวลาในการปลูกพืชได้ ในที่นี้ค่อนข้างจะได้ค่าที่ไม่ smooth เพราะใช้แค่จุด

// สามารถใช้ค่าเฉลี่ยจุดภาพจากรูปปิดได้ ซึ่งต้องมีสเตปเพิ่มเติมในการเขียนโค้ด

// กราฟที่ได้สา่มารถดาวน์โหลดข้อมูลและดาวน์โหลดข้อมูลภาพได้

// ให้นิสิตลองเปลี่ยนช่วงเวลาเป็น 2018-2020

// ให้นิสิตลองสร้าง roi ในพื้นที่ที่่ตนเองสนใจ อย่าลืม rename และแก้โค้ดด้วย

////////////////////////////////////////////////////////////

//ในส่วนต่อไปจะสาธิตวิธีการ export ข้อมูลภาพออกไปใช้ข้างนอก GEE

//15.เริ่มจากการสร้างภาพ 8 บิท ด้วยคำสั่ง ee.Image.visualize()

var rgb\_vis = {min: 0, max: 0.3, bands: ['B4', 'B3', 'B2']};

function addNDVI(image) {

var ndvi = image.normalizedDifference(['B5', 'B4']);

return image.addBands(ndvi);

}

var filtered = L8.filterDate('2020-01-01', '2020-12-31');

var with\_ndvi = filtered.map(addNDVI);

var greenest = with\_ndvi.qualityMosaic('nd');

var rgb = greenest.visualize(rgb\_vis); //ตรงนี้ละที่ทำภาพให้เป็น 8 บิท

Map.addLayer(rgb,{}, 'RGB');

// กด run จะได้แผนที่ 8 บิท สามารถใช้ inspector ตรวจคำตอบได้ ด้วยการเลือกที่ภาพ

// จะพบว่ามีทั้งหมดสามแบนด์ที่เหลืออยู่ เมื่อกดเข้าไปที่ Objects > RGB>bands จะพบว่าแต่ะละแบนด์มีค่าเป็น unsigned int8

\*/

//15.1 ส่งออกภาพผลที่ได้ไปยังกุเกิ้ลไดร์ฟ

var rgb\_vis = {min: 0, max: 0.3, bands: ['B4', 'B3', 'B2']};

function addNDVI(image) {

var ndvi = image.normalizedDifference(['B5', 'B4']);

return image.addBands(ndvi);

}

var filtered = L8.filterDate('2020-01-01', '2020-12-31');

var with\_ndvi = filtered.map(addNDVI);

var greenest = with\_ndvi.qualityMosaic('nd');

var rgb = greenest.visualize(rgb\_vis);

Map.addLayer(rgb, {}, 'RGB (greenest pixel)');

Export.image.toDrive(rgb, 'GreenestPixel'); //ตรวจสอบไดร์ฟ

// กด run แล้วไปที่แท็ป Tasks เพื่อกด run แล้วจะมีไดอะล็อคขึ้น ให้เซตค่าตามที่ต้องการเช่น ขนาดจุดภาพ

/\*

ให้นิสิตดัดแปลงโค้ด พื้นที่ที่นิสิตสนใจ เปลี่ยนปีที่นิสิตสนใจ แล้วเปรียบเทียบระหว่างปี

\*/