วันที่ 1

มองป่าจากอวกาศ

พื้นฐานภาพดาวเทียม Earth Engine และวิธีให้ AI เขียนโค้ดให้เรา

2 ชั่วโมง · พื้นฐาน + สิ่งปกคลุมดิน

Lab 1 และ Lab 2 · ไม่ต้องมีพื้นฐานเขียนโปรแกรม

ก่อนเริ่ม

สองชั่วโมงต่อจากนี้

1. **แบบทดสอบก่อนเรียน** — 10 นาที ไม่มีผลต่อคะแนน ข้อสอบนี้วัดว่าคอร์สสอนได้ดีแค่ไหน ไม่ได้วัดว่าคุณเก่งแค่ไหน
2. **ดาวเทียมมองเห็นป่าอย่างไร** — พิกเซล แบนด์ และเทคนิคเดียวที่ทำให้การทำแผนที่พืชพรรณเป็นไปได้
3. **Earth Engine คืออะไร** — และทำไมคุณจะไม่ต้อง ดาวน์โหลดภาพดาวเทียมเลย
4. **Vibe coding** — วิธีให้ AI เขียนโค้ด Earth Engine ที่ถูกต้อง และวิธีจับได้เมื่อมันเขียนผิด
5. **Lab 1 และ Lab 2** — แผนที่แรกของคุณ และตัวเลขจริงตัวแรก คือพื้นที่ป่าเป็นเฮกตาร์ในจังหวัดที่คุณเลือกเอง

สัปดาห์นี้คุณจะไม่ได้เรียนเขียนโค้ด

สิ่งที่คุณจะได้เรียนคือการ **ระบุความต้องการ** ให้ชัดพอที่เครื่องจะเขียนโค้ดถูก และ **ตรวจสอบ** คำตอบได้ดีพอที่คุณจะกล้าเซ็นชื่อรับรอง อย่างหลังคือส่วนที่ยากกว่า และเป็นส่วนที่ยังมีค่าอยู่เสมอ

ส่วนที่หนึ่ง

ดาวเทียมมองเห็นป่าอย่างไร

มีสี่แนวคิดหลัก บวกเหตุผลว่าทำไมต้องใช้ RS และตระกูลดัชนีที่จะเจออีกในสัปดาห์นี้

แรงจูงใจ

แปลงสำรวจหนึ่งบอกได้แค่ 400 ตร.ม. แต่ป่ามีเป็นล้านเฮกตาร์

แปลงสำรวจหนึ่งแปลง
400 ตร.ม. ต่อแปลง

ภาพ Sentinel-2 หนึ่งภาพ
≈ 12,100 ตร.กม. ต่อภาพ

≈ แปลงสำรวจ 30 ล้านแปลง ใส่ในภาพเดียวได้พอดี

ทำไมแค่แปลงสำรวจไม่พอ

ทีมสำรวจภาคสนามที่วัดพื้นที่หนึ่งเฮกตาร์อย่างละเอียด — เส้นผ่านศูนย์กลาง ความสูง ชนิดพันธุ์ — อาจใช้เวลาหลายวัน ภูมิประเทศชันหรือเข้าถึงไม่ได้ทำให้บางภูมิภาคอยู่นอกเหนือขอบเขต และการสำรวจทั่วไปครอบคลุมพื้นที่ไม่ถึง 1% ของภูมิประเทศทั้งหมด

สิ่งที่ดาวเทียมให้คุณได้

รอบการบินซ้ำ — Sentinel-2 กลับมาทุก 5 วัน **คลังภาพย้อนหลัง** — Landsat บันทึกมาตั้งแต่ปี 1972 **ความครอบคลุม** — คำสั่งเดียวตอบได้ทั้งจังหวัด ไม่ใช่แค่ตัวอย่างส่วนหนึ่ง

ภาพดาวเทียมคือตารางของค่าที่วัดได้

ภาพดาวเทียมไม่ใช่ภาพถ่าย แต่เป็น **ตารางของตัวเลข** และตัวเลขแต่ละตัวคือค่าที่วัดได้จริง ว่าแสงสะท้อนกลับมาจากพื้นดินตรงนั้นมากแค่ไหน

พื้นที่หนึ่งช่อง = หนึ่ง **พิกเซล (pixel)** สำหรับ Sentinel-2 หนึ่งพิกเซลครอบคลุม **10 × 10 เมตร** ประมาณสนามแบดมินตันหนึ่งสนาม

แปลว่าพิกเซลเดียวบอกเรื่องต้นไม้ต้นเดียวไม่ได้ แต่บอกเรื่องต้นไม้ยี่สิบต้นที่ยืนอยู่ในช่องนั้นรวมกัน

ความละเอียดที่จะเจอสัปดาห์นี้

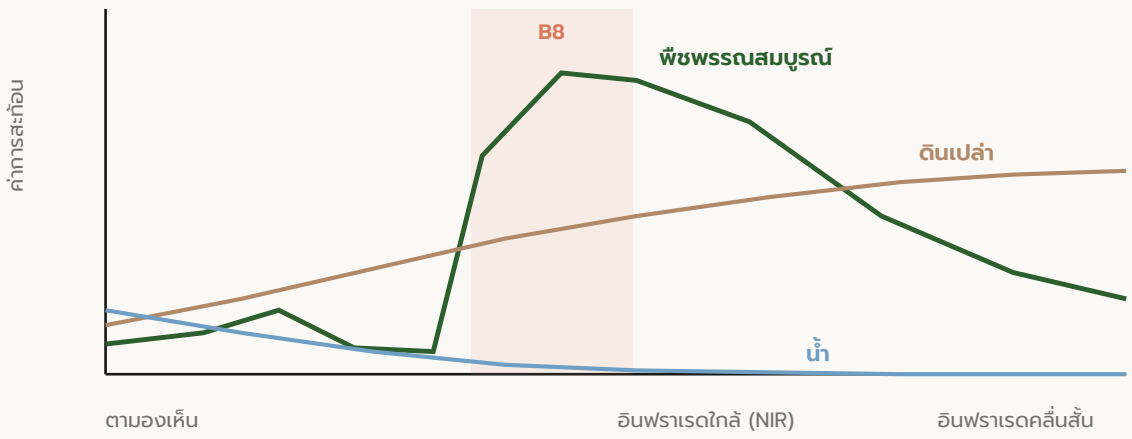
Sentinel-2	10 ม.	หย่อมป่า
Landsat	30 ม.	ย้อนถึงปี 1984
MODIS	500 ม.	ไฟป่า ระดับภูมิภาค
CHIRPS	5.5 กม.	ปริมาณฝน

พิกเซลเล็กกว่าไม่ได้ดีกว่าเสมอไป มันกินค่าประมวลผลมากกว่า และบ่อยครั้งก็ตอบคำถามได้ไม่ดีขึ้นเลย

ภาพหนึ่งภาพมีหลายชั้น ชั้นละหนึ่งช่วงคลื่น

ตาคนเห็นสามสี ดาวเทียมวัดได้ **สิบสาม** ช่วงคลื่น รวมช่วงที่ตาเรามองไม่เห็นเลยหลายช่วง แต่ละช่วงเรียกว่าหนึ่ง **แบนด์ (band)**

แบนด์ที่สำคัญที่สุดสำหรับงานป่าไม้คือ **B8 อินฟราเรดใกล้ (near-infrared)** ใบไม้ที่ยังมีชีวิตสะท้อนช่วงคลื่นนี้ออกมาแรงมาก และไม่มีอะไรอื่นในภูมิประเทศทำแบบนั้น



ทำไมเรื่องนี้ถึงสำคัญ

พืชพรรณที่สมบูรณ์คือสิ่งที่สว่างที่สุดในภาพช่วง NIR และค่อนข้างมืดในช่วงสีแดง ช่องว่างระหว่างสองค่านี้กว้าง วัดได้ และไม่ซ้ำกับอะไร นั่นแหละคือเหตุผลเดียวที่ทำให้เราทำแผนที่ป่า จากวงโคจรได้

แนวคิดที่ 3 จาก 4

ดัชนีเปลี่ยนหลายแบนด์ให้เหลือตัวเลขเดียวที่ใช้งานได้

NDVI — ดัชนีพืชพรรณ (Normalised Difference Vegetation Index) คือหัวใจทั้งหมด ของงาน remote sensing ด้านพืชพรรณ ย่อได้เหลือบรรทัดเดียว

$$NDVI = \frac{(NIR - Red)}{(NIR + Red)}$$

สว่างมากในช่วง NIR และมืดในช่วงแดง จะได้ค่าบวกสูง นั่นคือต้นไม้ การลบวัดช่องว่าง ส่วนการหารตัดผลของว่าวันนั้นแดดจัดแค่ไหนออกไป

อ่านค่า NDVI

-1 ถึง 0	น้ำ เงาเมฆ
0 ถึง 0.2	ดินเปล่า หิน สิ่งปลูกสร้าง
0.2 ถึง 0.5	หญ้า พืชไร่ พื้นที่ปกคลุมบาง
0.7 ถึง 0.9	**ป่าสมบูรณ์**

NDVI เกิน 1 ไม่ได้เด็ดขาด ถ้าของคุณเกิน แปลว่ามีอะไรผิด และวันนี้คุณจะได้เห็นว่าผิดตรงไหน

แนวคิดที่ 4 จาก 4

ศัตรูตัวจริงคือเมฆ และค่ามัธยฐานเอาชนะมันได้

เมืองไทยเมฆเยอะ ภาพเดี่ยวๆ ในหน้าฝนมักขาวโพลนทั้งภาพ

ทางแก้ไม่ใช่การไล่หาวันที่ฟ้าใสสมบูรณ์แบบสักวัน แต่คือเอาภาพ**ทุกภาพ** ในช่วงสามเดือน มาทิ้งพิกเซลที่เป็นเมฆออก แล้วเอาค่า**มัธยฐาน (median)** ของแต่ละพิกเซลจากที่เหลือ

เมฆสว่างและเกิดเป็นครั้งคราว ส่วนพื้นดินมืดกว่าและอยู่คงที่ ค่ามัธยฐานจึงตกลงบนพื้นดินแทบทุกครั้ง

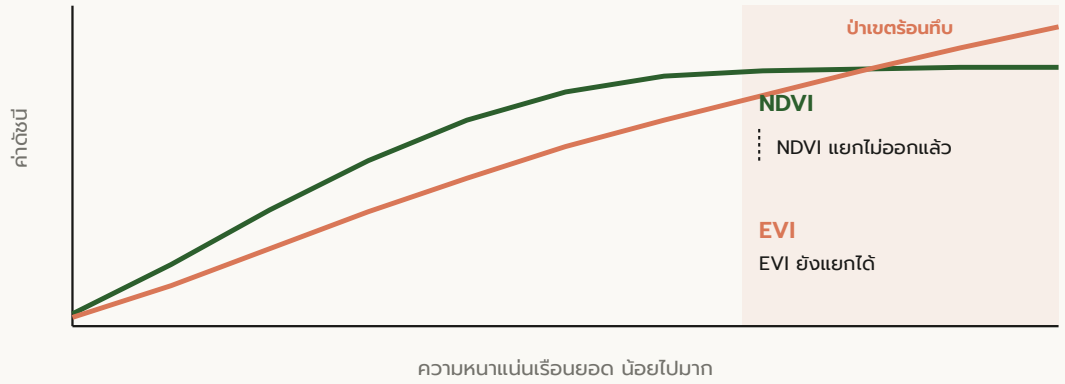
รูปแบบที่จะใช้ตลอดทั้งสัปดาห์

```
collection
  .filterBounds(aoi)          # ที่ไหน
  .filterDate(start, end)   # เมื่อไหร่
  .map(mask_clouds)         # ทิ้งพิกเซลเสีย
  .median()                  # ได้ภาพสะอาด 1 ภาพ
```

สี่บรรทัด นี่คือกระดูกสันหลังของงานวิเคราะห์ ภาพดาวเทียมแทบทุกงานที่คุณจะทำต่อจากนี้

เหนือกว่า NDVI

NDVI มีจุดบอด — มันอิ่มตัวในป่าทึบ



NDVI อ่านค่าได้ **0.85–0.95 แทบทุกจุด** ในเรือนยอดเขตร้อนที่ปิดทึบ และหมดพื้นที่ให้แยกป่าออกจากกัน การเพิ่ม **แบนด์สีน้ำเงิน** — คือ EVI — แก้ผลจากหมอกควันและคืนพื้นที่นั้นกลับมา

แผนที่ EVI จริง ดูผ่าน EO Browser

```
EVI = G × (NIR − Red) / (NIR + C1·Red − C2·Blue + L)
```

นี่คือสิ่งที่คุณจะคำนวณได้จริงใน Colab

ตระกูล NDVI

สูตรเดียวกัน แค่เปลี่ยนแบนด์

NDVI คือหนึ่งในตระกูลดัชนีเล็กๆ ที่มีรูปแบบเดียวกันหมด — **(แบนด์ A - แบนด์ B) / (แบนด์ A + แบนด์ B)** — แค่เล็งไปคนละคำถาม

ดัชนี	แบนด์	ตอบคำถามอะไร
NDVI	NIR, Red	เขียวแค่ไหน
NDWI	NIR, SWIR	เรือนยอดมีน้ำอยู่เท่าไหร่
NBR	NIR, SWIR	ไหม้รุนแรงแค่ไหน

NDWI จับสัญญาณความเครียดจากภัยแล้งได้ก่อน เรือนยอดจะเปลี่ยนสีให้เห็นเป็นสัปดาห์ — ปริมาณน้ำในใบลดลงก่อนสีจะเปลี่ยนเสมอ

คุณจะคำนวณดัชนีตัวนี้จริงในวันที่ 3

NBR คือตัวเลขที่อยู่เบื้องหลัง **dNBR** แผนที่ความรุนแรงของไฟ ที่คุณจะสร้างใน Lab 5 สำหรับฤดูไฟจริงในเชียงใหม่

สูตรรูปแบบเดียวกับ NDVI แค่เปลี่ยนแบนด์ เปลี่ยนคำถาม — รูปแบบนี้ซ้ำอยู่แทบทุกที่ในสาขานี้

ทิศทางต่อจากนี้

จากนับพิกเซล สู่การนับต้นไม้อัตโนมัติ

- **Deep learning** โครงข่ายประสาทเทียมเริ่มตรวจจับและนับ ต้นไม้หรือสัตว์แต่ละตัวในภาพได้โดยตรง แทนที่จะให้คนไล่ลากเส้นทีละภาพ
- **การประมวลผลแบบ cloud-native** ข้อมูลระดับเพตะไบต์กับ เครื่องประมวลผลอยู่ที่เดียวกันแล้ว ไม่มีใครต้องดาวน์โหลดภาพมาก่อน
- **เซนเซอร์ไฮเปอร์สเปกตรัม** แบนด์แคบๆ หลายร้อยแบนด์แทนที่จะมี แค่ไม่กี่แบนด์ ทำให้วัดองค์ประกอบทางเคมีของใบได้ — ไนโตรเจน ปริมาณน้ำ ไม่ใช่แค่ความเขียว

ส่วน cloud-native ไม่ใช่แนวโน้มอนาคตสำหรับคุณ แต่คือ **สิ่งที่คุณกำลังจะใช้เวลาที่เหลือของวันนี้เรียนรู้** Earth Engine คือสิ่งนี้เป๊ะๆ — ข้อมูลกับเครื่องประมวลผลอยู่ด้วยกันแล้ว รอแค่สูตรจากคุณ

ส่วนที่สอง

Earth Engine คืออะไรกันแน่

แนวคิดที่ทำให้ทั้งหมดนี้ทำได้บนโน้ตบุ๊กเครื่องเดียว

แนวคิดหลัก

คลังข้อมูลบวกเครื่องประมวลผล ทั้งคู่อยู่บนคลาวด์

คลังข้อมูล
กว่า 90 เพตะไบต์
ชุดข้อมูลกว่า 1,000 ชุด
ภาพย้อนหลัง 50 ปี

—

เครื่องของ Google
ประมวลผลที่นั่น
ไม่ใช่บนเครื่อง
ของคุณ

—

คุณ
ส่งสูตรไป
รับแผนที่
หรือตัวเลขกลับมา

คุณไม่ต้องดาวน์โหลดภาพดาวเทียมเลยสักภาพ

วิธีเดิม

หาภาพ ดาวน์โหลด 800 MB รอ ติดตั้งโปรแกรม ทำซ้ำทุกวันที่และทุกไทล์ ดิสก์เต็ม แล้วก็ล้มเลิก

Earth Engine

เขียนบอกว่าอยากได้อะไร Google รันให้ข้างๆ ตัวข้อมูล แล้วส่งกลับมาเฉพาะคำตอบ งานย้อนหลังยี่สิบปีใช้เวลาไม่กี่วินาที

ข้อแม้

ประมวลผลฟรี แต่ไม่ได้ไม่จำกัด

บัญชีของคุณได้ **150 EECU-hours ต่อเดือน** ถือว่าเหลือเฟือ แต่มันคืองบประมาณจริง และเป็นไปได้ที่จะใช้หมดไปมากในเซลล์เดียวถ้าไม่ระวัง

มีสองนิสัยที่จะทำให้คุณปลอดภัยตลอดทั้งคอร์ส

- **ทำทีละจังหวัด ไม่ใช่ทั้งประเทศ**
- **ใส่ `scale=` เสมอ** ใช้ 100 เมตรสำหรับการคำนวณพื้นที่ ขอ 10 เมตรคือขอให้เครื่องทำงานมากขึ้นร้อยเท่าเพื่อคำตอบเท่าเดิม

scale มีราคาเท่าไหร่

SCALE	จำนวนพิกเซลในหนึ่งจังหวัด
`scale=10`	~120 ล้าน
`scale=30`	~14 ล้าน
`scale=100`	~1.2 ล้าน
`scale=500`	~5 หมื่น

สำหรับคำถาม "มีป่ากี่เฮกตาร์" ทั้งสี่ค่าให้คำตอบเท่ากันโดยต่างกันไม่ถึงเศษเสี้ยวของเปอร์เซ็นต์ แต่มีแค่ค่าเดียวที่คุ้มกับเงินที่จ่าย

ส่วนที่สาม

Vibe coding

วิธีให้ AI เขียนโค้ด Earth Engine ที่ถูกต้อง และวิธีจับได้เมื่อมันเขียนผิด

AI เร็ว มั่นใจ และไม่รู้ว่าตัวเองตอบถูกหรือเปล่า

คุณช้ากว่า แต่คุณเป็นคนเดียวในวงจรนี้ที่บอกได้ นั่นแหละคืองานของคุณ และมันไม่ต้องอาศัยการเขียน Python เลย

เครื่องมือของคุณ

Antigravity — ฟรี และแบ่งหน้าที่เป็นสองส่วนโดยตั้งใจ

Antigravity คือ AI ผู้ช่วยฟรีของ Google สำหรับคอร์สนี้ ติดตั้งครั้งเดียว (วันที่ 0) ล็อกอินด้วย Gmail ส่วนตัว แล้วเปิดแผงแชทของมัน

มันเปิดและรันไฟล์ได้เองด้วย **แต่คุณจะไม่ใช้ส่วนนั้น** ใช้แค่แชทอย่างเดียว — ถามหาโค้ด รับโค้ดกลับมา เหมือนคุยกับ Claude หรือ ChatGPT ทุกประการ

1. วาง prompt ลงในแชทของ Antigravity
2. คัดลอกโค้ดที่ได้มา
3. วางลงในเซลล์ Colab แล้วรันที่นั่น

กฎข้อเดียว

Antigravity **เขียน**โค้ด **Colab เป็นคนรัน** คัดลอกคำตอบทุกครั้งข้ามไปเอง — ห้ามปล่อยให้ Antigravity รันอะไรกับบัญชี Earth Engine ของคุณโดยตรงเด็ดขาด

วิธีการ

สูตร PROMPT

P	**Purpose**	อยากได้แผนที่หรือตัวเลขอะไร
R	**Resource**	ใช้ชุดข้อมูลไหน ระบุ ID ให้ตรง
O	**Outline**	พื้นที่ไหน ช่วงเวลาใด
M	**Must-haves**	scale หน่วย ชุดสี คำอธิบายแผนที่
P	**Platform**	Colab, ee + geemap, ล็อกอินแล้ว
T	**Test**	พิมพ์ตัวเลข 1 ค่าให้ตรวจสอบได้

ใส่ให้ครบทั้งหกช่อง แล้วโค้ดมักจะใช้ได้ตั้งแต่ครั้งแรก

ช่องที่ทุกคนชอบข้าม

Test ถ้าไม่มีช่องนี้ คุณจะได้แผนที่สวยๆ มาหนึ่งใบ โดยไม่มีทางรู้เลยว่ามันถูกไหม ขอตัวเลขที่พิมพ์ออกมาสักค่าเสมอ เป็นค่าที่คุณเทียบกับสิ่งที่รู้อยู่แล้วได้

ตัวอย่าง prompt ที่กรอกครบแล้ว

```
You are a Google Earth Engine Python expert.

PURPOSE   Build one cloud-free Sentinel-2 image of my study area.
RESOURCE  Use dataset COPERNICUS/S2_SR_HARMONIZED.
OUTLINE   Area is in the variable `aoi`. Dates 2024-11-01 to 2025-02-28.
MUST      Mask cloud and shadow with the SCL band (drop values 3, 8, 9, 10).
          Divide by 10000 for reflectance. Take the median.
PLATFORM  Google Colab, `ee` and `geemap` imported and authenticated.
TEST      Print how many scenes went into the composite.
```

สังเกตสิ่งที่ **ไม่มี** อยู่ในนั้น คือ Python สักบรรทัด คุณแค่อธิบายปัญหาเป็นภาษาคน ด้วยคำศัพท์ของสาขาตัวเอง นั่นคือทักษะทั้งหมด

เขียน prompt เป็นภาษาอังกฤษ เพราะชื่อชุดข้อมูล ชื่อแบนด์ และเอกสารของ Earth Engine เป็นภาษาอังกฤษทั้งหมด — AI จะตอบแม่นกว่ามาก

วงจรแก้ error

1. **หยุดก่อน** อย่าแก้สามอย่างพร้อมกันแล้วรันใหม่เผื่อฟลุก
2. **คัดลอก error ทั้งหมด** ไม่ใช่แค่บรรทัดสุดท้าย บรรทัดข้างบนบอกว่าพังที่เซลล์ไหนบรรทัดไหน
3. **วางกลับไปให้ AI** พร้อมประโยคเดียวว่ากำลังพยายามทำอะไร
4. **ขอโค้ดที่แก้แล้ว** ไม่ใช่ขอคำอธิบายยาวๆ
5. **รันใหม่แล้วตรวจใหม่** error หายไม่ได้แปลว่าคำตอบถูก

```
This is the error I got. Please give me
the corrected code.

I was trying to: calculate hectares of
forest in my province

[paste the ENTIRE red error message]
```

อย่าพิมพ์โค้ดใหม่ด้วยมือเด็ดขาด

ทุกตัวอักษรที่พิมพ์เองคือโอกาสพิมพ์ผิดเพิ่ม แล้วคุณจะต้องแก้สองปัญหาแทนที่จะเป็นหนึ่ง

ทักษะที่สำคัญที่สุด

การตรวจสอบความสมเหตุสมผล

โค้ดที่รันได้ไม่ใช่โค้ดที่ถูก ก่อนจะเชื่อตัวเลขไหน ให้เทียบกับสิ่งที่คุณรู้อยู่แล้วเสมอ

ค่าที่วัด	ช่วงที่เป็นไปได้	ถ้าเจอแบบนี้แปลว่าผิด
NDVI	-1 ถึง 1 · ป่า 0.7–0.9	**3000** — ลืมหารด้วย 10000 · **ศูนย์หมด** — mask กินไปหมดแล้ว
พื้นที่แต่ละชั้น	น้อยกว่าพื้นที่จังหวัด	**ป่ามากกว่าแผ่นดิน** — บวก ตร.ม. แล้วเรียกว่าเฮกตาร์
ชีวมวล	100–400 Mg/ha	**5** — เฉลี่ยรวมผืนน้ำ · **20,000** — หน่วยผิด
อุณหภูมิ	15–35 °C	**ราว 300** — นั่นคือเคลวิน
ความถูกต้อง	0.75–0.92	**0.99 ขึ้นไป** — ข้อมูลทดสอบรั่วเข้าไปในชุดฝึก

คำถามที่ต้องถามทุกครั้งไม่มีข้อยกเว้น นี่คือชีวมวลหรือคาร์บอน เฮกตาร์หรือตารางเมตร เซลเซียสหรือเคลวิน คำตอบที่ผิดในสาขานี้เกือบทั้งหมดคือความผิดพลาดเรื่องหน่วย ที่ใส่หน้ากากเป็นแผนที่สวยๆ

ความเสี่ยงอีกอย่าง

AI กุสิ่งที่ดูเหมือนจริงขึ้นมาเอง

มันมักทำตอนที่ของจริงเป็นเรื่องเฉพาะทาง ซึ่งตรงกับชื่อชุดข้อมูลของ Earth Engine แทบทั้งหมด

ป้องกันได้สามชั้น เรียงตามความเร็ว

- เทียบ ID กับ **cheat sheet** ก่อนรันอะไรทั้งสิ้น
- ใช้ `print(img.bandNames().getInfo())` แทนการเชื่อชื่อแบนด์
- ค้นใน catalog ทางการ ถ้าไม่มีหน้าของ Google ก็แปลว่าไม่มีชุดข้อมูลนั้น

ตัวอย่างจริงจากตอนสร้างคอร์สนี้

- `...global_forest_change_2024_v1_12`
  มีจริง แต่ตกรุ่นแล้ว ปัจจุบันคือ `2025_v1_13`
- `ee.Image('NASA/ORNL/biomass_carbon_density/v1')`
  ID ถูก แต่ชนิดผิด มันเป็น ImageCollection **AI ทุกตัวที่เราทดสอบตอบผิดข้อนี้**
- แบนด์ชื่อ `NDVI` บนภาพ Sentinel-2 ดิบ
  ไม่มีอยู่จริง ต้องคำนวณเอง

ส่วนที่สี่

Lab 1 และ Lab 2

เปิด Colab ได้แล้ว นี่คือส่วนที่คุณมาเพื่อสิ่งนี้

LAB 1

แผนที่ดาวเทียมใบแรกของคุณ

● สาธิตสด — ทำตามไปด้วยกัน

1. เลือกจังหวัด ของคุณเอง ไม่ใช่ของผม
2. ขอภาพ Sentinel-2 ทุกภาพเหนือจังหวัดนั้นในหน้าแล้งนี้
3. ทิ้งเมฆ แล้วเอาค่ามัธยฐาน
4. แปลงเป็น NDVI
5. **ตรวจสอบตัวเลข**

ค่าที่ควรได้ — จังหวัดน่าน

0.754
NDVI เฉลี่ย
ป่าหนาแน่นมาก

194
ภาพ Sentinel-2
ที่ใช้ผสม

จังหวัดของคุณจะได้ค่าต่างออกไป นั่นคือประเด็น แต่ถ้า NDVI เกิน 1 หรือต่ำกว่า 0.3 ในพื้นที่ที่มีป่า ให้หยุดแล้วหาสาเหตุก่อนไปต่อ

LAB 2

จากภาพ สู่ตัวเลข

แผนที่ไม่ใช่ผลลัพธ์ **ตัวเลขต่างหากคือผลลัพธ์**

มีคนจำแนกสิ่งปกคลุมดินทั้งโลกไว้ให้แล้ว หน้าที่ของคุณคือเอาแผนที่ของเขามาแปลงเป็นเฮกตาร์ เพราะ "มีป่าอยู่เท่าไหร่" คือคำถามที่งานจริงจะถามคุณ

```
ee.Image.pixelArea()   # ตร.ม. ต่อพิกเซล
  .divide(1e4)         # -> เฮกตาร์
  .addBands(landcover)
  .reduceRegion(grouped sum)
```

จังหวัดน่าน · ESA WorldCover 2021

ประเภท	เฮกตาร์	สัดส่วน
พื้นที่ต้นไม้ปกคลุม	1,033,493	84.4%
ทุ่งหญ้า	126,798	10.4%
พื้นที่เกษตร	52,639	4.3%
สิ่งปลูกสร้าง	7,497	0.6%
แหล่งน้ำ	3,100	0.3%

ผลรวมต้องใกล้เคียงกับพื้นที่ จังหวัดที่พิมพ์ออกมาก่อนหน้านี้ ถ้าคลาดสิบเท่า แปลว่าหารผิดตัว

สองชุดข้อมูลที่น่าเชื่อถือทั้งคู่ จังหวัดเดียวกัน ต่างกัน 52,000 เฮกตาร์

WorldCover บอกว่าน่านมีต้นไม้ปกคลุม **84%** · Dynamic World บอก **88%** ไม่มีใครโกหกสักคน

บทเรียน

ทำไมสองแผนที่ถึงไม่ตรงกัน และต้องทำอย่างไร

สามเหตุผลที่ตรงไปตรงมา

- **นิยามของคำว่า "ต้นไม้" ต่างกัน** วัดความสูงเรือนยอด หรือร้อยละการปกคลุม แต่ละทีมขีดเส้นคนละที่
- **ช่วงเวลาต่างกัน** WorldCover เป็นภาพนิ่งปี 2021 ส่วน Dynamic World ผสมจาก ช่วงวันที่ *คุณ* เลือกเอง
- **ระบบชั้นข้อมูลต่างกัน** WorldCover มีทุ่งหญ้าชั้นเดียว ส่วน Dynamic World แยกหญ้ากับไม้พุ่มออกจากกัน

อย่าเขียนแบบนี้

"พื้นที่ป่าของจังหวัดน่านคือ 1,033,493 เฮกตาร์"

เขียนแบบนี้

*"พื้นที่ป่าของจังหวัดน่านคือ 1,033,493 เฮกตาร์ **ตามข้อมูล ESA WorldCover v200 ปี 2021**"*

ชื่อชุดข้อมูลและปีคือ **ส่วนหนึ่งของคำตอบ** ไม่ใช่เชิงอรรถ นิสัยข้อเดียวนี้จะช่วยคุณจากความน่าอายได้มากกว่าอย่างอื่นทั้งหมดในคอร์สนี้

ก่อนกลับ

สรุปวันที่ 1

ตอนนี้คุณทำอะไรได้แล้วบ้าง

- สร้างภาพดาวเทียมปลอดเมฆของที่ไหนก็ได้ในไทย
- แปลงเป็นแผนที่ความเขียว
- ดึงพื้นที่เป็นเฮกตาร์แยกตามประเภทสิ่งปกคลุมดิน
- เขียน prompt ที่ได้โค้ดใช้งานได้
- **บอกได้ว่าคำตอบผิดเมื่อไหร่**

การบ้าน — 15 นาที

รัน Lab 1 และ Lab 2 อีกครั้งโดยตั้ง `PROVINCE` เป็น **จังหวัดบ้านเกิด** ของคุณ

จดสองตัวเลขมาพรุ่งนี้

- NDVI เฉลี่ย
- พื้นที่ต้นไม้ปกคลุมเป็นเฮกตาร์

พรุ่งนี้ คุณจะเลิกใช้แผนที่ของคนอื่น แล้วสร้างเอง โดยแยกป่าดิบออกจากป่าผลัดใบ จากนั้นใส่ตัวเลขคาร์บอนลงไป