

Week 3: Raster Data Processing with {terra}

GEO2408

ผศ.ดร.มรกต วรชัยรุ่งเรือง

2025-12-10

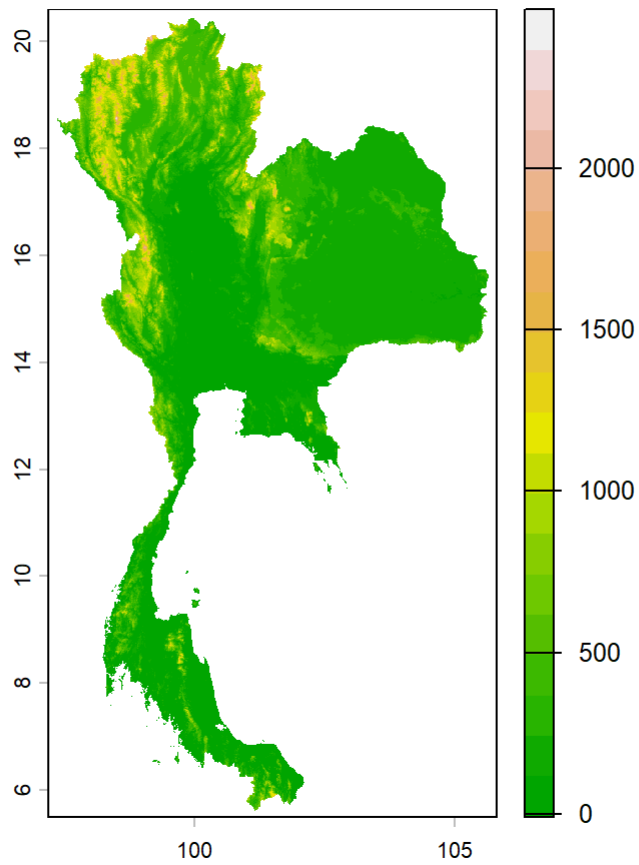
```
# 1. โหลดข้อมูลความสูง (Elevation) ความละเอียด 30 arc-seconds (~1 km)
# path=tempdir() คือโหลดเก็บไว้ชั่วคราว ไม่รกเครื่อง
dem_th <- geodata::elevation_30s(country = "THA", path = tempdir())
```

```
# 2. ตรวจสอบข้อมูล
print(dem_th)
```

```
## class      : SpatRaster
## size       : 1812, 1032, 1 (nrow, ncol, nlyr)
## resolution : 0.008333333, 0.008333333 (x, y)
## extent     : 97.2, 105.8, 5.5, 20.6 (xmin, xmax, ymin, ymax)
## coord. ref.: lon/lat WGS 84
## source     : THA_elv_msk.tif
## name       : THA_elv_msk
## min value  :          -9
## max value  :          2552
```

```
# คำอธิบายผลลัพธ์:
# dimensions: จำนวนแถว x คอลัมน์ (Pixels)
# resolution: ขนาดของ 1 Pixel (หน่วยเป็นองศา)
# crs        : ระบบพิกัด (WGS84)
# min/max    : ความสูงต่ำสุด/สูงสุด
# Plot ดูภาพรวมทั้งประเทศ
plot(dem_th, main = "Elevation of Thailand (Meters)", col = terrain.colors(20))
```

Elevation of Thailand (Meters)



1. โหลดขอบเขตประเทศไทย(Vector)

```
cm_boundary <- gadm(country = "THA", level = 1, path = tempdir()) %>%  
  st_as_sf()
```

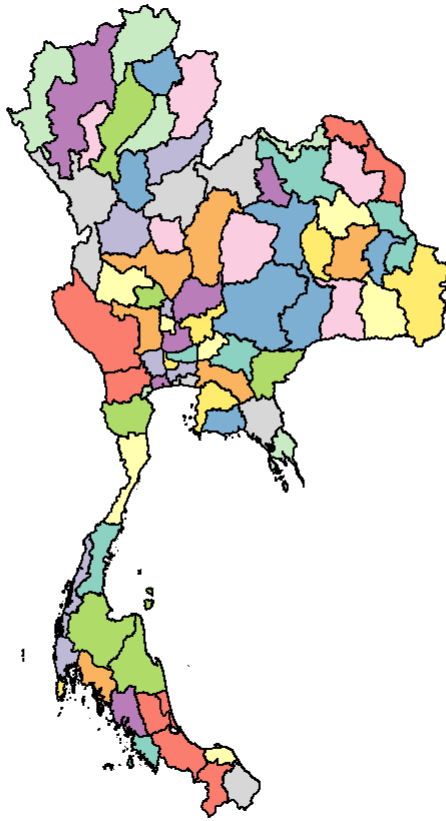
Plot เช็คความถูกต้อง

```
print(cm_boundary$NAME_1)
```

## [1]	"Amnat Charoen"	"Ang Thong"
## [3]	"Bangkok Metropolis"	"Bueng Kan"
## [5]	"Buri Ram"	"Chachoengsao"
## [7]	"Chai Nat"	"Chaiyaphum"
## [9]	"Chanthaburi"	"Chiang Mai"
## [11]	"Chiang Rai"	"Chon Buri"
## [13]	"Chumphon"	"Kalasin"
## [15]	"Kamphaeng Phet"	"Kanchanaburi"
## [17]	"Khon Kaen"	"Krabi"
## [19]	"Lampang"	"Lamphun"
## [21]	"Loei"	"Lop Buri"
## [23]	"Mae Hong Son"	"Maha Sarakham"
## [25]	"Mukdahan"	"Nakhon Nayok"
## [27]	"Nakhon Pathom"	"Nakhon Phanom"
## [29]	"Nakhon Ratchasima"	"Nakhon Sawan"
## [31]	"Nakhon Si Thammarat"	"Nan"
## [33]	"Narathiwat"	"Nong Bua Lam Phu"
## [35]	"Nong Khai"	"Nonthaburi"
## [37]	"Pathum Thani"	"Pattani"
## [39]	"Phangnga"	"Phatthalung"
## [41]	"Phayao"	"Phetchabun"
## [43]	"Phetchaburi"	"Phichit"
## [45]	"Phitsanulok"	"Phra Nakhon Si Ayutthaya"
## [47]	"Phrae"	"Phuket"
## [49]	"Prachin Buri"	"Prachuap Khiri Khan"
## [51]	"Ranong"	"Ratchaburi"
## [53]	"Rayong"	"Roi Et"
## [55]	"Sa Kaeo"	"Sakon Nakhon"
## [57]	"Samut Prakan"	"Samut Sakhon"
## [59]	"Samut Songkhram"	"Saraburi"
## [61]	"Satun"	"Si Sa Ket"
## [63]	"Sing Buri"	"Songkhla"
## [65]	"Sukhothai"	"Suphan Buri"
## [67]	"Surat Thani"	"Surin"
## [69]	"Tak"	"Trang"
## [71]	"Trat"	"Ubon Ratchathani"
## [73]	"Udon Thani"	"Uthai Thani"
## [75]	"Uttaradit"	"Yala"
## [77]	"Yasothon"	

```
plot(cm_boundary["NAME_1"])
```

NAME_1



1. โหลดขอบเขตจังหวัดเชียงใหม่ (Vector)

```
Chiang_Mai=cm_boundary%>%
```

```
  st_as_sf() %>%
```

```
  filter(NAME_1 == "Chiang Mai")
```

Plot เช็คความถูกต้อง

```
plot(st_geometry(Chiang_Mai), main = "Chiang Mai Boundary")
```

Chiang Mai Boundary

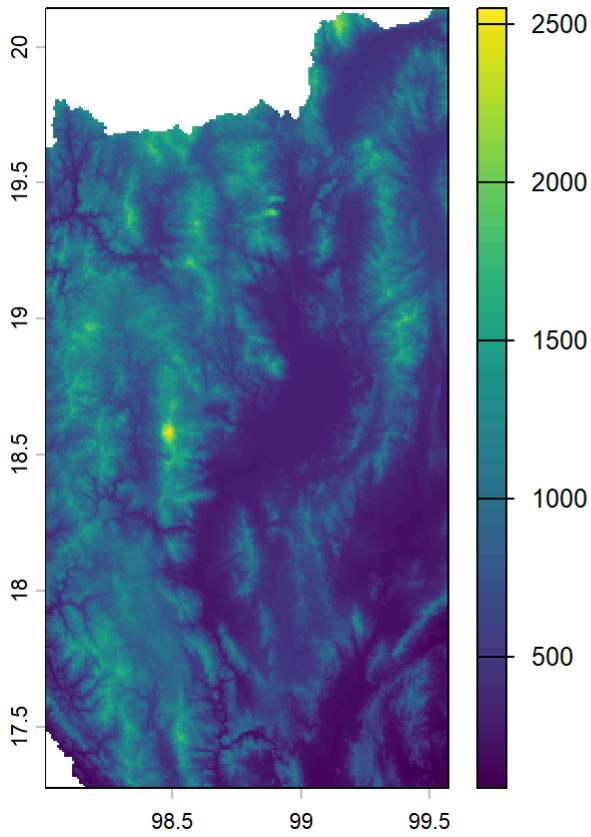


```
# 2. Crop (ตัดกรอบสี่เหลี่ยม)
dem_cm_crop <- crop(dem_th, Chiang_Mai)

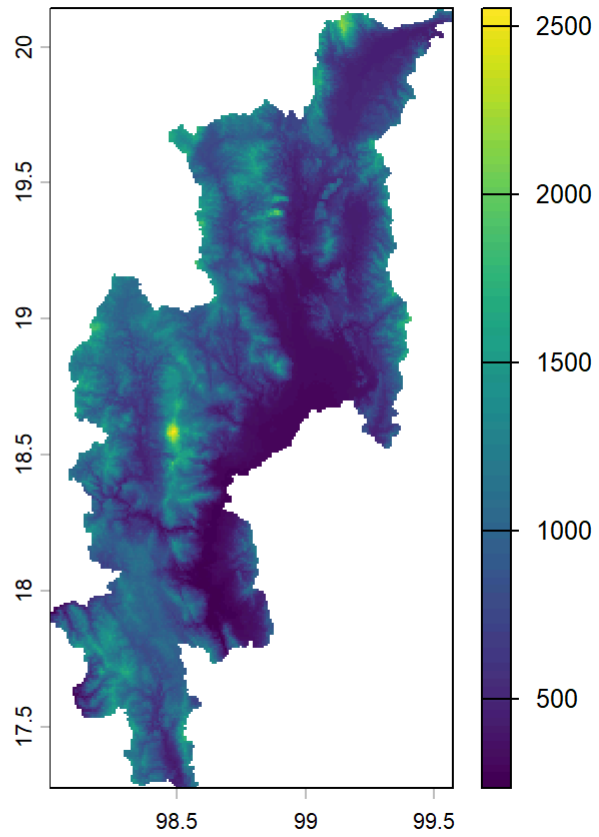
# 3. Mask (ตัดตามรูปร่างจังหวัด)
dem_cm <- mask(dem_cm_crop, Chiang_Mai)

# แสดงผลเปรียบเทียบ
par(mfrow = c(1, 2)) # แบ่งหน้าจอ 2 ช่อง
plot(dem_cm_crop, main = "Cropped (Rectangular)")
plot(dem_cm, main = "Masked (Exact Shape)")
```

Cropped (Rectangular)



Masked (Exact Shape)



```
par(mfrow = c(1, 1)) # คืนค่าหน้าจอ
```

```
# คำนวณ Slope (หน่วยเป็นองศา)
```

```
slope <- terrain(dem_cm, v = "slope", unit = "degrees")
```

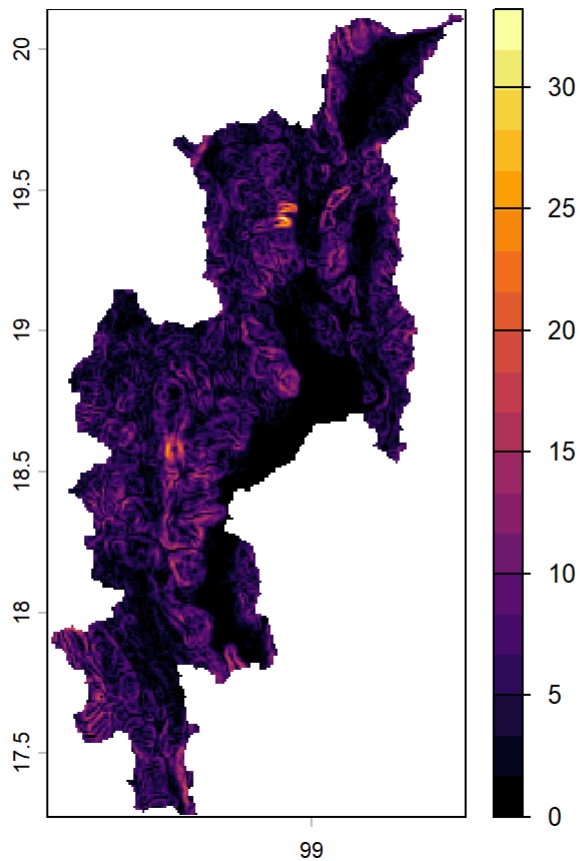
```
# คำนวณ Aspect (ทิศทางการลาดเขาหันหน้าไป)
```

```
aspect <- terrain(dem_cm, v = "aspect", unit = "degrees")
```

```
# Plot
```

```
plot(slope, main = "Slope Map (Degrees)", col = inferno(20))
```

Slope Map (Degrees)



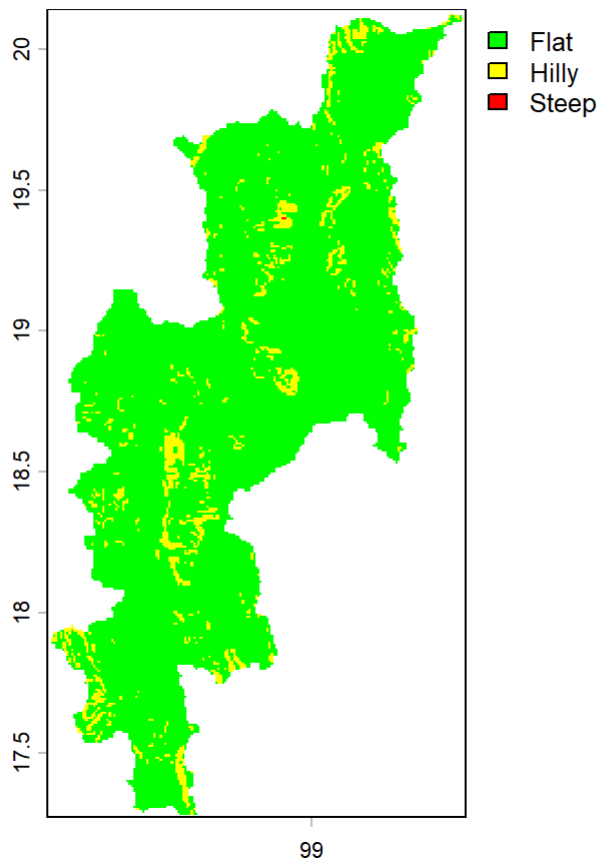
```
#0-10 องศา = พื้นที่ราบ (Flat) -> ให้ค่า 1
#10-30 องศา = เนินเขา (Hilly) -> ให้ค่า 2
#30 องศา = ภูเขาสูง (Steep) -> ให้ค่า 3
# สร้าง Matrix กฎการจัดกลุ่ม: [min, max, new_value]
rcl_matrix <- matrix(c(0, 10, 1,
                      10, 30, 2,
                      30, 90, 3), ncol = 3, byrow = TRUE)

# สั่ง Reclassify
slope_class <- classify(slope, rcl_matrix)

# เปลี่ยนตัวเลขเป็น Label (เพื่อให้ Plot สวย)
levels(slope_class) <- data.frame(id=1:3, label=c("Flat", "Hilly", "Steep"))

plot(slope_class, main = "Risk Classification", col = c("green", "yellow", "red"))
```

Risk Classification



```
# สร้าง Raster ว่างๆ ขนาดเท่า DEM เชียงใหม่
r_template <- dem_cm

# 1. จำลอง Band สีแดง (Red) - สมมติว่าพืชดูดซับแสงแดง (ค่าน้อย = พืชเยอะ)
# ใช้ Random noise ผสมกับค่าความสูง (สมมติที่สูงป่าเยอะกว่าเมือง)
set.seed(42)
band_red <- r_template
values(band_red) <- runif(ncell(r_template), 0.1, 0.3) + (values(dem_cm)/5000)

# 2. จำลอง Band อินฟราเรด (NIR) - พืชสะท้อนแสงนี้ได้ดี (ค่ามาก = พืชเยอะ)
band_nir <- r_template
values(band_nir) <- runif(ncell(r_template), 0.4, 0.8)

# รวมเป็นก้อนเดียวกัน (Multi-band Raster)
satellite_img <- c(band_red, band_nir)
names(satellite_img) <- c("Red", "NIR")

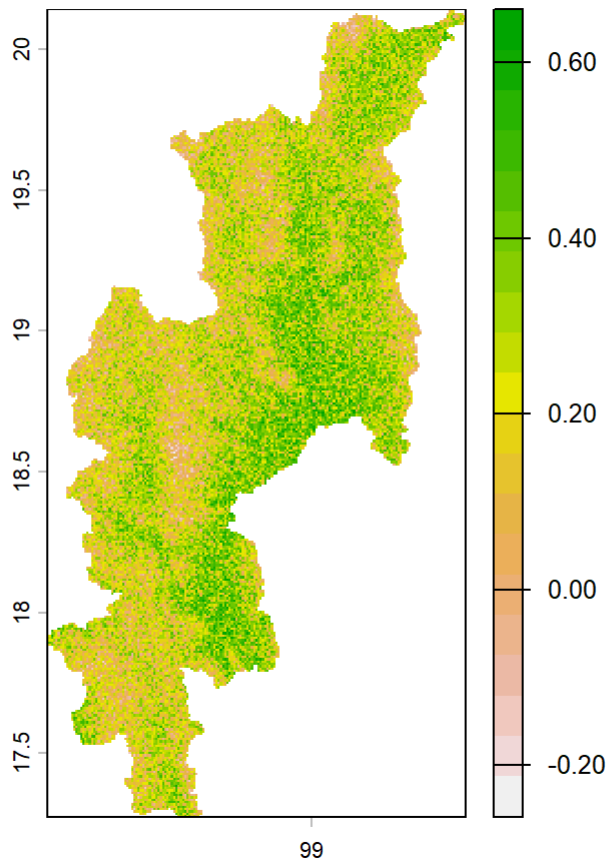
print(satellite_img)
```

```
## class      : SpatRaster
## size       : 344, 188, 2 (nrow, ncol, nlyr)
## resolution : 0.008333333, 0.008333333 (x, y)
## extent     : 98.00833, 99.575, 17.275, 20.14167 (xmin, xmax, ymin, ymax)
## coord. ref.: lon/lat WGS 84
## source(s)  : memory
## varnames   : THA_elv_msk
##            : THA_elv_msk
## names      :      Red,      NIR
## min values  : 0.1484519, 0.4000129
## max values  : 0.7589654, 0.7999788
```

```
# สูตร: (NIR - Red) / (NIR + Red)
ndvi <- (satellite_img$NIR - satellite_img$Red) / (satellite_img$NIR + satellite_img$Red)

plot(ndvi, main = "Simulated NDVI (Vegetation Health)", col = rev(terrain.colors(20)))
```

Simulated NDVI (Vegetation Health)



```

# สมมติเราสนใจเฉพาะพื้นที่ "Steep" (slope class 3) ที่เป็นพื้นที่เสี่ยงภัย
# 1. เลือกเฉพาะ Pixel ที่เป็น Class 3 (Steep)
risk_area <- slope_class == 3

# แก้ไข: ตั้งชื่อ Layer ให้ชัดเจนก่อนแปลง (กันความสับสน)
names(risk_area) <- "is_high_risk"

# 2. แปลง 1 (True) ให้เป็น Polygon
risk_poly <- as.polygons(risk_area, values = TRUE) %>%
  st_as_sf() %>%
  # ตอนนี้เรามั่นใจแล้วว่าคอลัมน์ชื่อ 'is_high_risk' แน่นนอน
  filter(is_high_risk == 1)

# 3. แสดงผล
mapview(risk_poly, col.regions = "red", layer.name = "High Risk Areas")

```

