

Week 7: Spatial Regression & GWR

GEO2408

ผศ.ดร.มรกต วรชัยรุ่งเรือง

2026-01-29

```
# 1. โหลดขอบเขต กทม.
bkk <- gadm(country = "THA", level = 2, path = tempdir()) %>%
  st_as_sf() %>%
  filter(str_detect(NAME_1, "Bangkok")) %>%
  st_union() %>% # รวมเขตทั้งหมดเป็นชั้นเดียว (Outline)
  st_transform(32647) # UTM 47N

# 2. สร้างจุดสุ่ม 400 จุด (ตัวแทนหมู่บ้านจัดสรร)
set.seed(555)
houses <- st_sample(bkk, size = 400) %>% st_as_sf()

# 3. สร้างตัวแปรทำนาย (Predictors)
coords <- st_coordinates(houses)
# - Distance to CBD (สยาม): ยิ่งใกล้ยิ่งถูก
center_point <- st_sfc(st_point(c(666000, 1520000)), crs = 32647) # พิกัดสยามโดยประมาณ
houses$dist_cbd <- as.numeric(st_distance(houses, center_point)) / 1000 # หน่วย กม.

# - Green Space (พื้นที่สีเขียว): สุ่มค่า (0-100%)
houses$green_score <- runif(400, 0, 100)

# 4. สร้างตัวแปรตาม (Price) โดยให้ความสัมพันธ์เปลี่ยนตามพื้นที่ (GWR Effect)
# - โชนเหนือ: พื้นที่สีเขียวมีผลต่อราคามาก (คนชอบธรรมชาติ)
# - โชนใต้: พื้นที่สีเขียวเฉยๆ เน้นใกล้เมืองมากกว่า
lat_effect <- (coords[,2] - min(coords[,2])) / (max(coords[,2]) - min(coords[,2])) # 0=South, 1=North

# สูตรจำลองราคา (หน่วยล้านบาท):
# Price = 5 + (-0.2 * Dist_CBD) + (Coefficient_Green * Green_Score) + Error
# โดยที่ Coefficient_Green จะแปรผันตาม Latitude (ยิ่งขึ้นเหนือ ยิ่งมีค่ามาก)
coef_green <- 0.02 + (lat_effect * 0.05)

houses$price <- 5 + (-0.2 * houses$dist_cbd) + (coef_green * houses$green_score) + rnorm(400, 0, 0.5)

# ดูข้อมูล
head(houses)
```

```
## Simple feature collection with 6 features and 3 fields
## Geometry type: POINT
## Dimension: XY
## Bounding box: xmin: 656398.3 ymin: 1522532 xmax: 703834.1 ymax: 1541292
## Projected CRS: WGS 84 / UTM zone 47N
##           x dist_cbd green_score price
## 1 POINT (667484.3 1522532) 2.935307 91.992856 9.8367395
## 2 POINT (656398.3 1523219) 10.126864 9.037171 2.4143289
## 3 POINT (699344.4 1540415) 39.097827 1.203006 -4.0379486
## 4 POINT (671966.3 1541292) 22.112253 71.630152 5.9054787
## 5 POINT (703834.1 1536012) 41.082763 46.466573 0.1657531
## 6 POINT (677113.8 1532378) 16.634909 80.088083 6.0797539
```

```
# สร้าง Linear Model ปกติ
ols_model <- lm(price ~ dist_cbd + green_score, data = houses)

summary(ols_model)
```

```
##
## Call:
## lm(formula = price ~ dist_cbd + green_score, data = houses)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.4128 -0.4797 -0.0157  0.5066  2.4200
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.712047   0.108191  43.55  <2e-16 ***
## dist_cbd     -0.185897   0.003927  -47.34  <2e-16 ***
## green_score   0.049961   0.001396   35.79  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7896 on 397 degrees of freedom
## Multiple R-squared:  0.8957, Adjusted R-squared:  0.8952
## F-statistic: 1704 on 2 and 397 DF,  p-value: < 2.2e-16
```

```
# การแปลผล:
# - dist_cbd ตัดลบ (สมเหตุสมผล: ยิ่งไกลเมือง ราคาบ้านยิ่งถูก)
# - green_score เป็นบวก (พื้นที่สีเขียวเพิ่มราคาบ้าน)
# - R-squared: ดูว่าโมเดลแม่นยำแค่ไหน (เช่น 0.6 คือทำนายถูก 60%)
# เอาค่า Error ใส่กลับไปในแผนที่
houses$ols_resid <- residuals(ols_model)
tm_shape(bkk) + tm_borders() +
  tm_shape(houses) +
  tm_dots("ols_resid",
    style = "jenks",
    midpoint = 0,
    palette = "RdBu",
    title = "OLS Residuals")
```



```
# ถ้าเห็นสีแดงกระจุกตัว หรือสีน้ำเงินกระจุกตัว แสดงว่ามี Spatial Autocorrelation (ต้องใช้ GWR)
# เอาค่า Error ใส่กลับไปในแผนที่
houses$ols_resid <- residuals(ols_model)
tm_shape(bkk) + tm_borders() +
  tm_shape(houses) +
  tm_dots("ols_resid",
    style = "jenks",
    midpoint = 0,
    palette = "RdBu",
    title = "OLS Residuals")
```



```
# ถ้าเห็นสีแดงกระจุกตัว หรือสีน้ำเงินกระจุกตัว แสดงว่ามี Spatial Autocorrelation (ต้องใช้ GWR)
```

```
library(GWmodel)
```

```
# 1. แปลงข้อมูลเป็น Spatial Object (sp format)
```

```
# เพราะ package GWmodel เวอร์ชันเก่าชอบ sp มากกว่า sf
```

```
houses_sp <- as(houses, "Spatial")
```

```
# 2. หา Bandwidth ที่เหมาะสม (Optimal Bandwidth)
```

```
# คือรัศมีที่จะกวาดเอาเพื่อนบ้านมาสร้างสมการ (เช่น 5 กม. หรือ 10 กม.)
```

```
bw <- bw.gwr(price ~ dist_cbd + green_score,  
             data = houses_sp,  
             approach = "AICc",  
             kernel = "gaussian",  
             adaptive = TRUE)
```

```
## Adaptive bandwidth (number of nearest neighbours): 254 AICc value: 894.3066
```

```
## Adaptive bandwidth (number of nearest neighbours): 165 AICc value: 855.418
```

```
## Adaptive bandwidth (number of nearest neighbours): 108 AICc value: 796.1799
```

```
## Adaptive bandwidth (number of nearest neighbours): 75 AICc value: 755.7917
```

```
## Adaptive bandwidth (number of nearest neighbours): 52 AICc value: 716.637
```

```
## Adaptive bandwidth (number of nearest neighbours): 40 AICc value: 687.7678
```

```
## Adaptive bandwidth (number of nearest neighbours): 30 AICc value: 664.6637
```

```
## Adaptive bandwidth (number of nearest neighbours): 26 AICc value: 655.7483
```

```
## Adaptive bandwidth (number of nearest neighbours): 21 AICc value: 645.2807
```

```
## Adaptive bandwidth (number of nearest neighbours): 21 AICc value: 645.2807
```

```
print(paste("Optimal Bandwidth (Number of Neighbors):", bw))
```

```
## [1] "Optimal Bandwidth (Number of Neighbors): 21"
```

```
# 3. รัศมี GWR Model
```

```
gwr_model <- gwr.basic(price ~ dist_cbd + green_score,  
                      data = houses_sp,  
                      bw = bw,  
                      kernel = "gaussian",  
                      adaptive = TRUE)
```

```
print(gwr_model)
```

```

## *****
## *                               Package   Gwmodel                               *
## *****
## Program starts at: 2026-01-29 22:23:12
## Call:
## gwr.basic(formula = price ~ dist_cbd + green_score, data = houses_sp,
##           bw = bw, kernel = "gaussian", adaptive = TRUE)
##
## Dependent (y) variable:  price
## Independent variables:   dist_cbd green_score
## Number of data points:  400
## *****
## *                               Results of Global Regression                               *
## *****
## Call:
## lm(formula = formula, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.4128 -0.4797 -0.0157  0.5066  2.4200
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.712047   0.108191   43.55  <2e-16 ***
## dist_cbd     -0.185897   0.003927  -47.34  <2e-16 ***
## green_score   0.049961   0.001396   35.79  <2e-16 ***
##
## ---Significance stars
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Residual standard error: 0.7896 on 397 degrees of freedom
## Multiple R-squared:  0.8957
## Adjusted R-squared:  0.8952
## F-statistic: 1704 on 2 and 397 DF,  p-value: < 2.2e-16
## ***Extra Diagnostic information
## Residual sum of squares: 247.5285
## Sigma(hat): 0.788626
## AIC: 951.1754
## AICc: 951.2766
## BIC: 591.1071
## *****
## *                               Results of Geographically Weighted Regression                               *
## *****
## *****Model calibration information*****
## Kernel function: gaussian
## Adaptive bandwidth: 21 (number of nearest neighbours)
## Regression points: the same locations as observations are used.
## Distance metric: Euclidean distance metric is used.
##
## *****Summary of GWR coefficient estimates:*****
##              Min.    1st Qu.    Median    3rd Qu.    Max.
## Intercept    3.524746  4.469417  4.916774  5.177891  6.1139
## dist_cbd     -0.262247 -0.214970 -0.195375 -0.178214 -0.1378
## green_score   0.033344  0.043795  0.049357  0.056602  0.0645
## *****Diagnostic information*****
## Number of data points: 400
## Effective number of parameters (2trace(S) - trace(S'S)): 33.90789
## Effective degrees of freedom (n-2trace(S) + trace(S'S)): 366.0921

```

```

## AICc (GWR book, Fotheringham, et al. 2002, p. 61, eq 2.33): 645.2807
## AIC (GWR book, Fotheringham, et al. 2002,GWR p. 96, eq. 4.22): 615.7902
## BIC (GWR book, Fotheringham, et al. 2002,GWR p. 61, eq. 2.34): 335.6422
## Residual sum of squares: 102.8256
## R-square value: 0.9566655
## Adjusted R-square value: 0.9526408
##
## *****
## Program stops at: 2026-01-29 22:23:12

```

เที่ยบ AICc: ถ้า AICc ของ GWR ต่ำกว่า OLS แสดงว่า GWR ดีกว่า

```

# 1. ดึงผลลัพธ์จาก GWR object กลับมาเป็น sf
# ผลลัพธ์จะอยู่ใน @SDF
gwr_results <- st_as_sf(gwr_model$SDF)

# ดูชื่อคอลัมน์ผลลัพธ์
# green_score (เฉยๆ) = ค่า Coefficient ของตัวแปร green_score ณ จุดนั้น
# green_score_SE = Standard Error
# green_score_TV = T-value (Significance)

# 2. Map ค่าสัมประสิทธิ์ของ Green Score
tm_shape(bkk) + tm_borders() +
  tm_shape(gwr_results) +
  tm_dots("green_score",
    style = "cont",
    palette = "Viridis",
    size = 0.5,
    title = "Coef: Green Score") +
  tm_layout(main.title = "Where does Green Space matter most?")

```

Where does Green Space matter most?

