

Week 6: Spatial Clustering (Unsupervised Learning)

GEO2408

ผศ.ดร.มรกต วรชัยรุ่งเรือง

2025-12-12

```
# 1. โหลดแผนที่ กทม. (เหมือน Week 5)
th_districts <- gadm(country = "THA", level = 2, path = tempdir())
bkk <- st_as_sf(th_districts) %>%
  filter(str_detect(NAME_1, "Bangkok")) %>%
  select(name = NAME_2) %>%
  st_transform(32647) # UTM 47N

# 2. จำลองข้อมูล (Simulate Socio-Econ Data)
# สมมติเรามีข้อมูล 3 มิติ: ราคาที่ดิน, ความหนาแน่นประชากร, จำนวนห้าง
set.seed(123)
coords <- st_coordinates(st_centroid(bkk))

# Logic: ใจกลางเมือง (Center) แพงและคนแน่น, ขอบเมือง (Periphery) ถูกและคนน้อย
dist_to_center <- sqrt((coords[,1] - mean(coords[,1]))^2 + (coords[,2] - mean(coords[,2]))^2)
norm_dist <- (max(dist_to_center) - dist_to_center) / max(dist_to_center) # 1 = ใจกลาง, 0 = ขอบ

bkk$land_price <- (norm_dist * 100000) + runif(50, 10000, 30000) # บาท/ตร.วา
bkk$pop_density <- (norm_dist * 5000) + runif(50, 500, 2000) # คน/ตร.กม.
bkk$malls <- round((norm_dist * 5) + runif(50, 0, 2)) # จำนวนห้าง

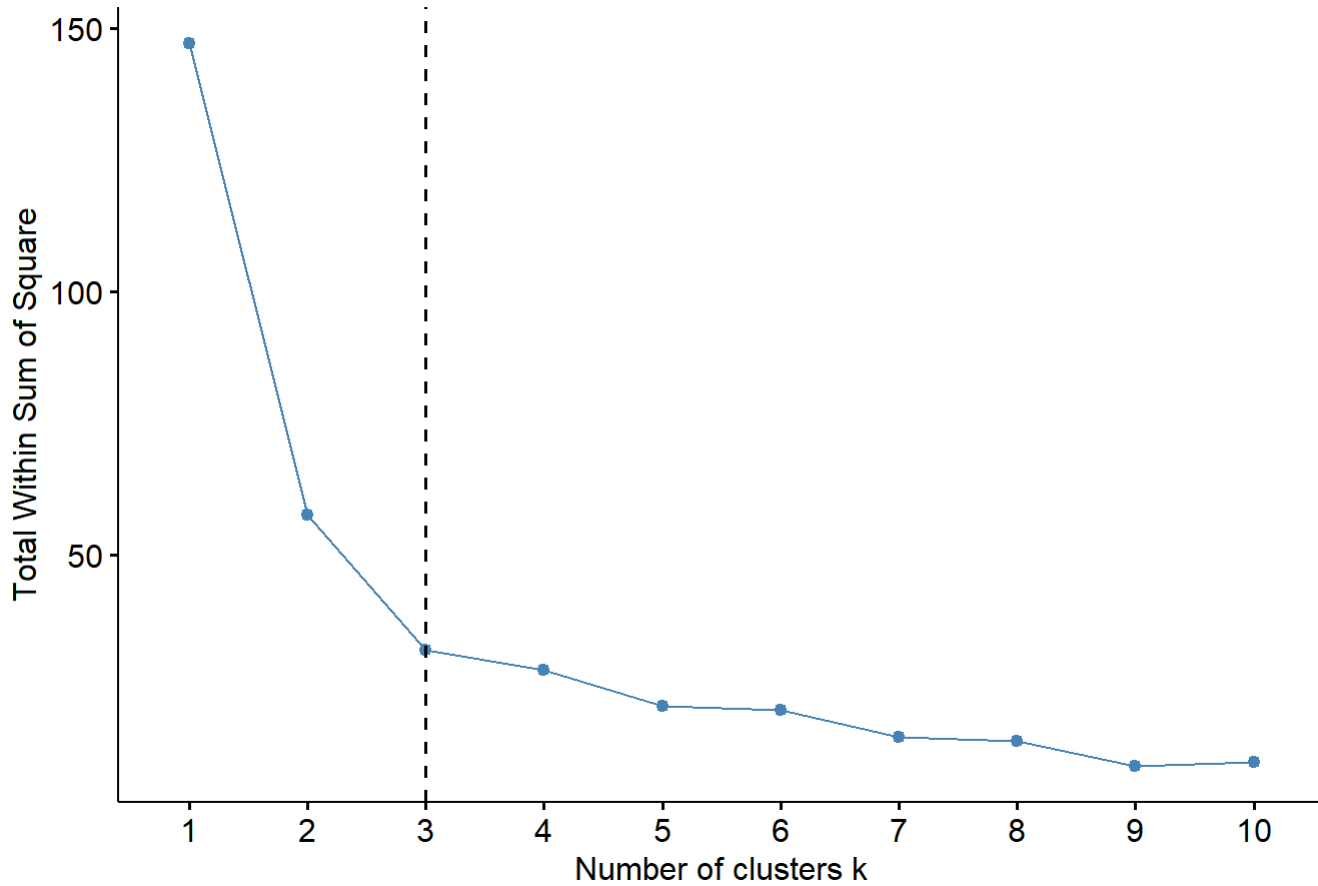
# ดูข้อมูล 5 แถวแรก
head(bkk %>% st_drop_geometry())
```

```
##           name land_price pop_density malls
## 1      Bang Bon  53653.61   2463.850     3
## 2      Bang Kapi 99335.25   4841.758     4
## 3      Bang Khae 65067.61   4042.791     3
## 4      Bang Khen 85096.65   3554.664     5
## 5 Bang Kho Laem 104562.59  5129.084     5
## 6      Bang Na  75953.21   4061.901     5
```

```
# 1. เตรียม Data (Scaling สำคัญมากสำหรับ K-Means!)
df_attr <- bkk %>%
  st_drop_geometry() %>%
  select(land_price, pop_density, malls) %>%
  scale() # ปรับฐานข้อมูลให้เป็น Z-score

# 2. หาจำนวน Cluster ที่เหมาะสม (Elbow Method)
fviz_nbclust(df_attr, kmeans, method = "wss") +
  geom_vline(xintercept = 3, linetype = 2)
```

Optimal number of clusters



```
# 3. รัน K-Means (สมมติเลือก 3 กลุ่ม: CBD, Urban, Suburban)
```

```
set.seed(99)
```

```
km_attr <- kmeans(df_attr, centers = 3, nstart = 25)
```

```
# 4. เอาผลลัพธ์ใส่กลับแผนที่
```

```
bkk$cluster_attr <- as.factor(km_attr$cluster)
```

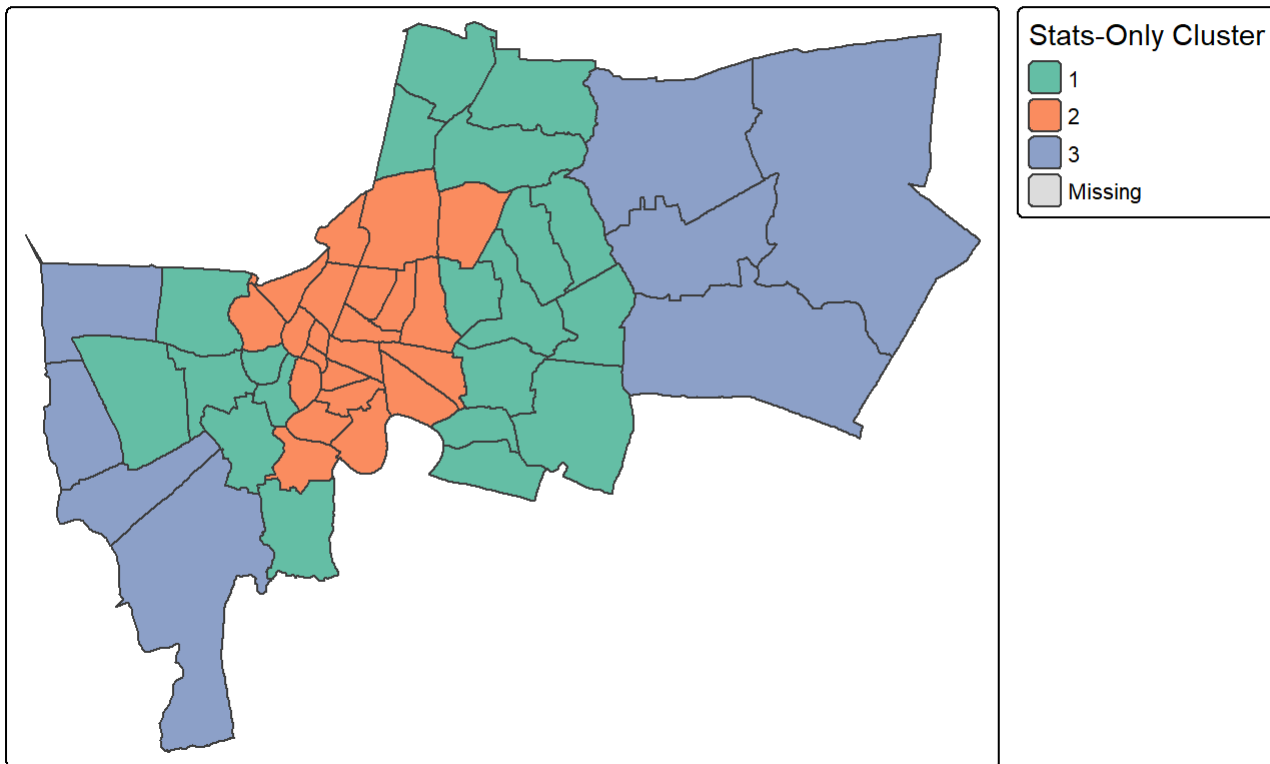
```
# แสดงผล
```

```
tm_shape(bkk) +
```

```
  tm_polygons("cluster_attr",  
             title = "Stats-Only Cluster",  
             palette = "Set2") +
```

```
  tm_layout(main.title = "Grouping by Attributes Only")
```

Grouping by Attributes Only



```
# 1. ดึงพิกัด XY (Centroid)
xy <- st_coordinates(st_centroid(bkk))

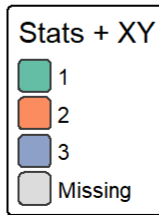
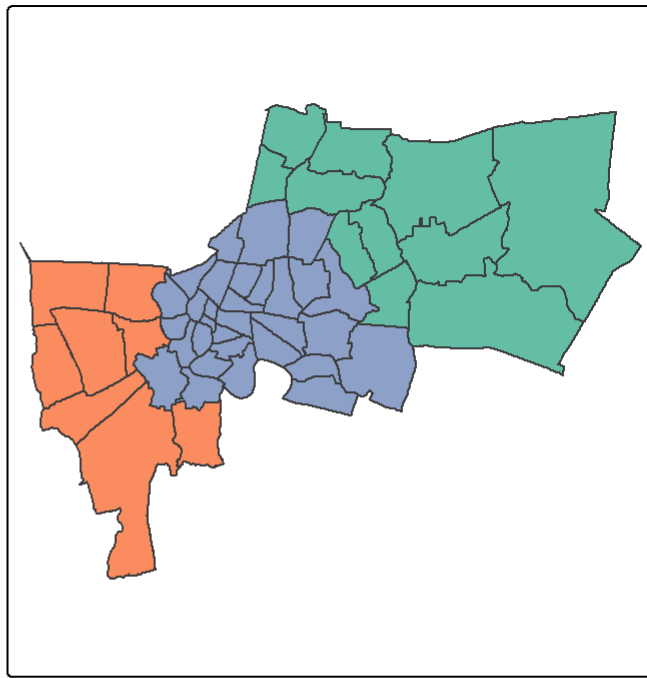
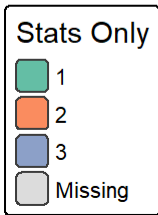
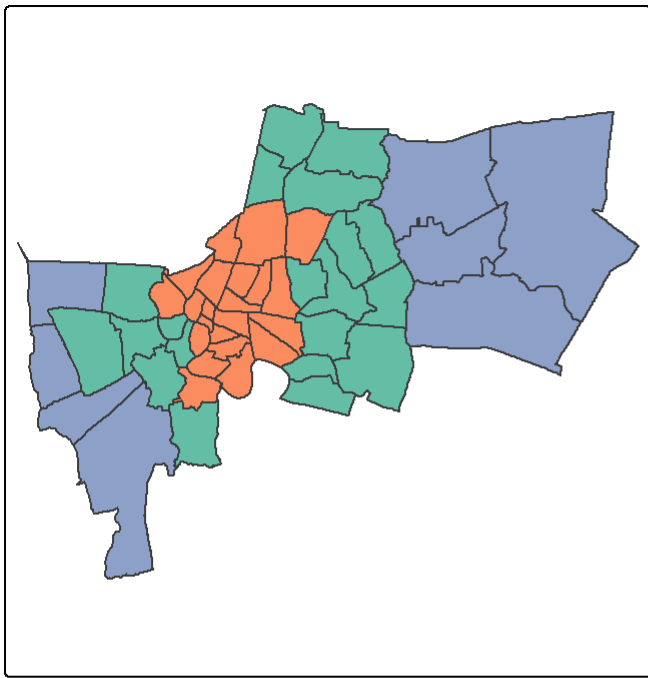
# 2. รวม Attribute + Location
#หมายเหตุ: เราต้อง Scale XY ด้วย ไม่งั้นค่าพิกัด (หลักแสน) จะกลบค่าอื่นๆ หมด
df_spatial <- cbind(df_attr, scale(xy))

# 3. รัน K-Means ใหม่
set.seed(99)
km_spatial <- kmeans(df_spatial, centers = 3, nstart = 25)

# 4. เก็บผลลัพธ์
bkk$cluster_spatial <- as.factor(km_spatial$cluster)

# แสดงผลเปรียบเทียบ
tmap_mode("plot")
map1 <- tm_shape(bkk) + tm_polygons("cluster_attr", palette="Set2", title="Stats Only")
map2 <- tm_shape(bkk) + tm_polygons("cluster_spatial", palette="Set2", title="Stats + XY")

tmap_arrange(map1, map2)
```

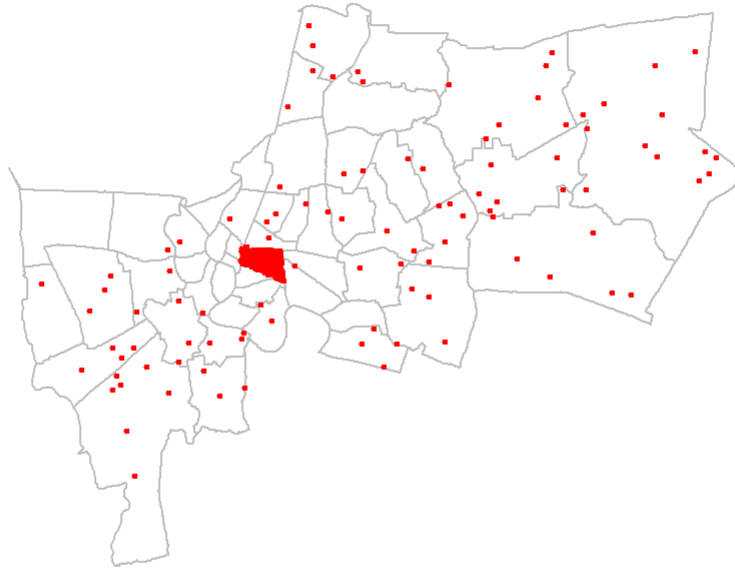


```
# 1. จำลองจุดคอนโด 500 จุด (เน้นเกาะกลุ่มกัน + มี Noise กระจายๆ)
# สร้างจุดสุ่ม 2 แบบผสมกัน
set.seed(123)
# แบบที่ 1: กระจายทั่วเมือง (Noise)
points_noise <- st_sample(bkk, size = 100)
# แบบที่ 2: กระจุกตัวกลางเมือง (Dense)
center_zone <- bkk %>% filter(name %in% c("Pathum Wan", "Watthana", "Khlong Toei"))
points_dense <- st_sample(center_zone, size = 400)

# รวมกัน
condos <- c(points_noise, points_dense) %>% st_as_sf()
coords_condo <- st_coordinates(condos)

# Plot ดูจุดตึ๊งๆ
plot(st_geometry(bkk), border="grey")
plot(condos, add=TRUE, col="red", pch=20, cex=0.5)
title("Simulated Condo Locations")
```

Simulated Condo Locations



#K-Means (Attribute): ใช้เมื่อต้องการหา “คู่แข่ง” เช่น #หาเขตที่มีพฤติกรรมลูกค้าเหมือนกัน เพื่อทำการตลาดแบบเดียวกัน #(ไม่จำเป็นต้องอยู่ติดกัน)

#Spatial K-Means: ใช้เมื่อต้องการ “แบ่งโซน” (Zoning) เช่น #การแบ่งเขตความรับผิดชอบของ Salesman หรือการแบ่งเขตจัดส่งสินค้า

#DBSCAN: ใช้เมื่อต้องการหา “Hotspot ตามธรรมชาติ” จากข้อมูลจุด เช่น #ย่านร้านอาหารกลางคืน, ย่านที่อยู่อาศัย, หรือจุดเสี่ยงอุบัติเหตุ #(โดยตัดจุดที่เกิดครั้งเดียวทิ้งไปได้)