

FINAL REPORT

登革熱

H24101222_陳凱騫

2025-01-03

Table of contents

空間統計	1
library	1
DATA IMPORT	2
PREPROCESS	2
BOUNDARY	4
Density plot	5
RISK LEVEL PLOT	8
點格局識別	14
Quadrant Analysis(QA)	14
VMR (Variance-to-Mean Ratio)	14
Ripley's K function	16
格數據統計	20
Local Moran's I	22
Getis-Ord Gi	23
Model	24
空間滯後效應 (Spatial Lag Effect)	24
空間滯後效應的例子	24
空間滯後模型的形式	24
相鄰的定義	25
結果詮釋	25
空間滯後效應的解釋	25
為何需要考慮空間滯後效應？	25

空間統計

library

載入需要使用的package。

```
rm(list = ls())
# ===== 1. 載入必要的套件 =====
library(Hmisc)
library(DataExplorer)
library(sp)
library(sf)
library(spacetime)
library(gstat)
library(dplyr)
```

```
library(lubridate)
library(spdep)
library(ggplot2)
library(RColorBrewer)
library(showtext)
showtext_auto()
```

DATA IMPORT

```
# ===== 2. 數據讀取與合併 =====
# 創建年份向量
years <- sprintf("%d", 105:113)
# 使用lapply批量讀取數據
data_list <- lapply(years, function(year) {
  file_name <- paste0(year, ".csv")
  df <- read.csv(file_name)
  df$year <- year
  return(df)
})

# 合併所有數據框
data_combined <- do.call(rbind, data_list)

# ===== 3. 初步數據檢查 =====
# 檢查每個數據框的行數
rows_count <- sapply(data_list, nrow)
names(rows_count) <- paste0("data", years)
# print("各年度數據行數：")
# print(rows_count)

# 基本數據檢查
# print("合併後的數據概要：")
# print(paste("總行數:", nrow(data_combined)))
# print("列名：")
# print(colnames(data_combined))

# 數據類型檢查
# str(data_combined)
# plot_missing(data_combined$容器指數)
```

PREPROCESS

```
# ===== 4. 數據清理與轉換 =====
# 日期轉換
data_combined <- data_combined %>%
  mutate(日期 = as.Date(日期, format = "%Y%m%d")) %>%
  filter(!is.na(日期)) %>%
  arrange(日期)

# head(data_combined)
# str(data_combined)
```

```

# 經緯度處理
data_combined <- data_combined %>%
  # 移除非數字字符
  mutate(
    經度 = gsub("[^0-9.]", "", 經度),
    緯度 = gsub("[^0-9.]", "", 緯度)
  ) %>%
  # 轉換為數字並處理經緯度順序
  mutate(
    經度 = ifelse(as.numeric(經度) < 90,
      as.numeric(緯度),
      as.numeric(經度)),
    緯度 = ifelse(as.numeric(緯度) > 90,
      as.numeric(經度),
      as.numeric(緯度))
  ) %>%
  # 移除NA值
  filter(!is.na(經度) & !is.na(緯度)) %>%
  # 確保經緯度在合理範圍內
  filter(經度 >= 120 & 經度 <= 122) %>%
  filter(緯度 >= 21.5 & 緯度 <= 23.45)

data_combined <- data_combined %>%
  mutate(
    經度 = as.numeric(經度),
    緯度 = as.numeric(緯度),
    year = as.numeric(year)
  )

# 時間分析
data_combined <- data_combined %>%
  mutate(
    year = year(日期),
    month = month(日期),
    day = day(日期)
  )

# str(data_combined)
# head(data_combined)

data <- data_combined
# 創建sf對象
data_sf <- st_as_sf(data, coords = c("經度", "緯度"), crs = 4326)

# 檢查空間對象s
# 繪製第1到5個屬性
# ... existing code ...

# 获取所有年份
unique_years <- unique(data_sf$year)

# 遍历每个年份
yearly_data_105 <- data_sf %>% filter(year == unique_years[1])

```

```

yearly_data_106 <- data_sf %>% filter(year == unique_years[2])
yearly_data_107 <- data_sf %>% filter(year == unique_years[3])
yearly_data_108 <- data_sf %>% filter(year == unique_years[4])
yearly_data_109 <- data_sf %>% filter(year == unique_years[5])
yearly_data_110 <- data_sf %>% filter(year == unique_years[6])
yearly_data_111 <- data_sf %>% filter(year == unique_years[7])
yearly_data_112 <- data_sf %>% filter(year == unique_years[8])
yearly_data_113 <- data_sf %>% filter(year == unique_years[9])

```

```

# 定義函數來處理數據框
process_data <- function(data) {
  coords <- st_coordinates(data)
  # 添加經緯度到數據框
  data$經度 <- coords[, 1]
  data$緯度 <- coords[, 2]
  # 轉換和過濾容器指數和容器級數
  data$容器指數 <- as.numeric(data$容器指數)
  data <- data %>% filter(!is.na(容器指數))
  data$容器級數 <- as.numeric(data$容器級數)
  data <- data %>% filter(!is.na(容器級數))
  # 移除經度和緯度中的缺失值
  data <- data %>% filter(!is.na(經度) & !is.na(緯度))
  # 確保經度和緯度為數值類型
  data$經度 <- as.numeric(data$經度)
  data$緯度 <- as.numeric(data$緯度)
  return(data)
}

```

```

# 處理所有年份的數據框
yearly_data_106 <- process_data(yearly_data_106)
yearly_data_107 <- process_data(yearly_data_107)
yearly_data_108 <- process_data(yearly_data_108)
yearly_data_109 <- process_data(yearly_data_109)
yearly_data_110 <- process_data(yearly_data_110)
yearly_data_111 <- process_data(yearly_data_111)
yearly_data_112 <- process_data(yearly_data_112)
yearly_data_113 <- process_data(yearly_data_113)
data_sf <- process_data(data_sf)

```

先將一些非常少數量的缺失值去除，最重要的是有發現一些經緯度紀錄相反和超出台南市範圍的資料，因此將超出範圍的去除，而經緯度相反的把它轉回來。

BOUNDARY

```

# ===== 讀取台南市行政區邊界檔案 =====
tainan_boundary <- st_read("tainan_town.shp")

```

Reading layer `tainan_town' from data source
 `C:\Users\user\Desktop\Textbook\4-1\stat_consult\final_report\disease\final_report_kcchen\tainan_town.shp'
 using driver `ESRI Shapefile'
 Simple feature collection with 37 features and 5 fields

Geometry type: POLYGON
Dimension: XY
Bounding box: xmin: 149903.2 ymin: 2531973 xmax: 214819.9 ymax: 2590210
Projected CRS: Transverse_Mercator

```
# 檢查行政區的 CRS  
crs_boundary <- st_crs(tainan_boundary)
```

```
# ===== 數據對象列表 =====  
yearly_data_list <- list(  
  `105` = yearly_data_105,  
  `106` = yearly_data_106,  
  `107` = yearly_data_107,  
  `108` = yearly_data_108,  
  `109` = yearly_data_109,  
  `110` = yearly_data_110,  
  `111` = yearly_data_111,  
  `112` = yearly_data_112,  
  `113` = yearly_data_113  
)
```

Density plot

```
# ===== 4. 每年數據處理 =====
```

```
# 加載所需的套件  
library(sf)  
library(dplyr)  
library(ggplot2)  
library(patchwork)
```

```
# 讀取行政區邊界資料  
tainan_boundary <- st_read("tainan_town.shp")
```

Reading layer `tainan_town` from data source
`C:\Users\user\Desktop\Textbook\4-1\stat_consult\final_report\disease\final_report_kcchen\tainan_town.shp`
using driver `ESRI Shapefile`
Simple feature collection with 37 features and 5 fields
Geometry type: POLYGON
Dimension: XY
Bounding box: xmin: 149903.2 ymin: 2531973 xmax: 214819.9 ymax: 2590210
Projected CRS: Transverse_Mercator

```
# 確保邊界的 CRS 一致性  
crs_boundary <- st_crs(tainan_boundary)
```

```
# 儲存每年結果的清單  
yearly_density_plots <- list()
```

```
# 遍歷 105 至 113 年  
for (year in 105:113) {  
  # 動態獲取每年的數據  
  yearly_data <- get(paste0("yearly_data_", year))
```

```

# 確保數據有正確的 CRS，並轉換為與邊界一致
crs_yearly <- st_crs(yearly_data)
if (crs_yearly != crs_boundary) {
  yearly_data <- st_transform(yearly_data, crs_boundary)
}

# 計算密度指數，按行政區聚合
density_by_town <- yearly_data %>%
  st_join(tainan_boundary, join = st_within) %>%
  group_by(TOWN_ID) %>%
  summarise(density_index = sum(陽性容器合計, na.rm = TRUE) /
            sum(調查容器合計, na.rm = TRUE) * 100)

# 確保 CRS 一致性
crs_density <- st_crs(density_by_town)
if (crs_boundary != crs_density) {
  density_by_town <- st_transform(density_by_town, crs_boundary)
}

# 合併行政區邊界
merged_data <- density_by_town %>%
  st_drop_geometry() %>%
  left_join(tainan_boundary, by = "TOWN_ID") %>%
  st_as_sf()

# 使用 ggplot 繪製密度指數分布
density_plot <- ggplot(data = merged_data) +
  geom_sf(aes(fill = density_index), color = "black", lwd = 0.2) +
  scale_fill_viridis_c(option = "viridis", na.value = "grey50") +
  labs(title = paste("Year", year), fill = "Density Index") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))

# 儲存結果到清單
yearly_density_plots[[as.character(year)]] <- density_plot
}

# 組合所有年份的圖表為 3x3 網格
combined_plot <- wrap_plots(yearly_density_plots, ncol = 3, nrow = 3)

# 顯示圖形(圖形太大，因此用輸入的方式輸出在qmd上)
#print(combined_plot)

```

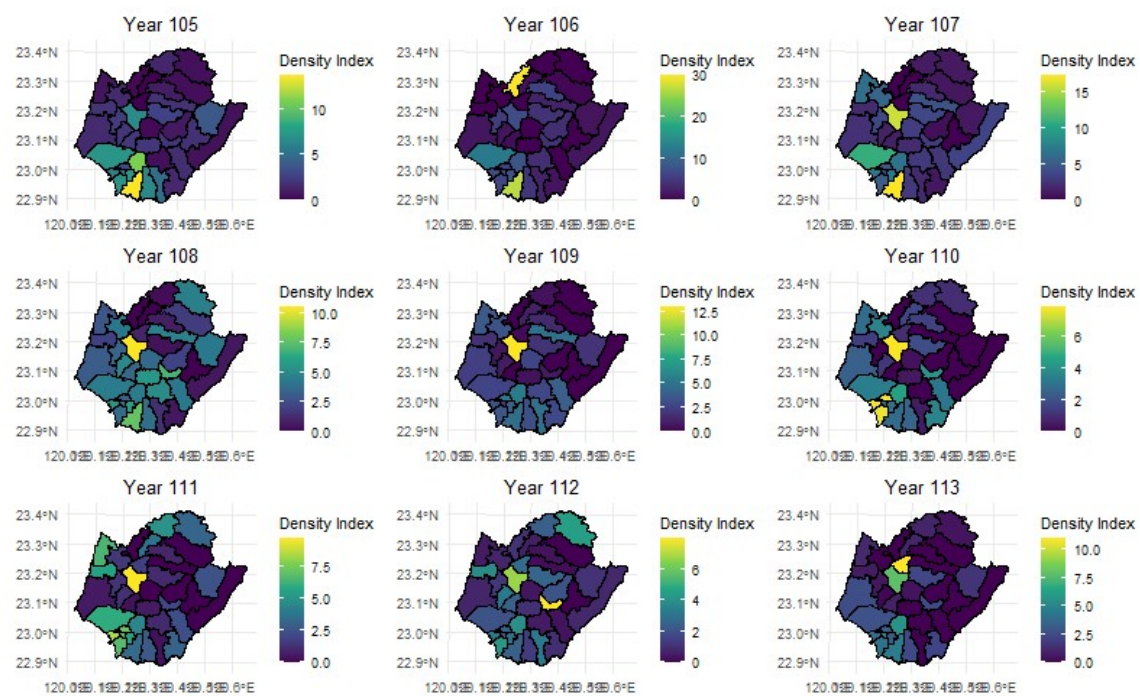


Figure 1: the plot of density for 105 to 113

我們選擇使用容器指數當作我們的一個風險指標，因為我們認為容器指數是一個陽性容器在整體容器中的比例，而陽性容器可以說是登革熱傳播的一個源頭之一。

而以上圖形告訴了我們每年容器指數在各個地區的數值，從中也可以看到像是麻豆區幾乎每一年都算容器指數相對較高的地區。

此外台南西南地區的容器指數也相對整個台南市算高的部分。

RISK LEVEL PLOT

```
# 儲存每年的結果
density_results <- list()

# ===== 進行迴圈處理每年的資料 =====
for (year in names(yearly_data_list)) {
  # 取得該年的數據
  yearly_data <- yearly_data_list[[year]]

  # ===== 檢查 CRS 並進行轉換 =====
  crs_yearly <- st_crs(yearly_data)
  if (crs_yearly != crs_boundary) {
    yearly_data <- st_transform(yearly_data, crs_boundary)
  }

  # ===== 計算密度指數 =====
  density_by_town <- yearly_data %>%
    st_join(tainan_boundary, join = st_within) %>%
    group_by(TOWN_ID) %>%
    summarise(
      density_index = sum(陽性容器合計, na.rm = TRUE) /
        sum(調查容器合計, na.rm = TRUE) * 100
    )

  # ===== 檢查 CRS 並轉換 =====
  crs_density <- st_crs(density_by_town)
  if (crs_boundary != crs_density) {
    density_by_town <- st_transform(density_by_town, crs_boundary)
  }

  # ===== 合併行政區資料 =====
  merged_data <- density_by_town %>%
    st_drop_geometry() %>%
    left_join(tainan_boundary, by = "TOWN_ID") %>%
    st_as_sf()

  # ===== 保存結果 =====
  density_results[[year]] <- merged_data

  # ===== 繪製每年的密度分布地圖 =====
  ggplot() +
    geom_sf(data = merged_data, aes(fill = density_index), color = "black", lwd = 0.3) +
    scale_fill_gradient(low = "green", high = "red", na.value = "gray") +
    labs(
      title = paste("台南市密度分佈圖 -", year),
```



```

    fill = "密度指數"
  ) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))
}

library(ggplot2)
library(patchwork)

# ===== 繪製每年的密度分布圖 =====
plots <- list()

for (year in names(density_results)) {
  merged_data <- density_results[[year]]

  # 創建圖表
  p <- ggplot() +
    geom_sf(data = merged_data, aes(fill = density_index), color = "black", lwd = 0.3) +
    scale_fill_gradient(low = "green", high = "red", na.value = "gray") +
    labs(
      title = paste("Year", year),
      fill = "Density Index"
    ) +
    theme_minimal() +
    theme(
      plot.title = element_text(hjust = 0.5),
      legend.position = "bottom"
    )

  # 將圖存入列表
  plots[[year]] <- p
}

# ===== 使用 patchwork 組合圖表 =====
combined_plot <- (plots[["105"]] | plots[["106"]] | plots[["107"]]) /
  (plots[["108"]] | plots[["109"]] | plots[["110"]]) /
  (plots[["111"]] | plots[["112"]] | plots[["113"]])

# 顯示組合圖
# print(combined_plot)

# ===== 計算所有年份的密度範圍 =====
all_density_values <- unlist(lapply(yearly_data_list, function(data) {
  # 檢查並統一 CRS
  if (st_crs(data) != st_crs(tainan_boundary)) {
    data <- st_transform(data, st_crs(tainan_boundary))
  }

  # 計算密度指數
  sum_density <- data %>%
    st_join(tainan_boundary, join = st_within) %>%
    group_by(TOWN_ID) %>%
    summarise(density_index = sum(陽性容器合計, na.rm = TRUE) /
      sum(調查容器合計, na.rm = TRUE) * 100)
})

```

```

    return(sum_density$density_index)
  )))

min_density <- min(all_density_values, na.rm = TRUE)
max_density <- max(all_density_values, na.rm = TRUE)

# ===== 進行迴圈處理每年的資料 =====
for (year in names(yearly_data_list)) {
  yearly_data <- yearly_data_list[[year]]

  # 檢查 CRS 並轉換
  crs_yearly <- st_crs(yearly_data)
  if (crs_yearly != crs_boundary) {
    yearly_data <- st_transform(yearly_data, crs_boundary)
  }

  # 計算密度指數
  density_by_town <- yearly_data %>%
    st_join(tainan_boundary, join = st_within) %>%
    group_by(TOWN_ID) %>%
    summarise(
      density_index = sum(陽性容器合計, na.rm = TRUE) /
        sum(調查容器合計, na.rm = TRUE) * 100
    )

  # 合併行政區資料
  merged_data <- density_by_town %>%
    st_drop_geometry() %>%
    left_join(tainan_boundary, by = "TOWN_ID") %>%
    st_as_sf()

  # 保存結果
  density_results[[year]] <- merged_data

  # 繪圖並設置統一的 scale
  ggplot() +
    geom_sf(data = merged_data, aes(fill = density_index), color = "black", lwd = 0.3) +
    scale_fill_gradient(
      low = "green",
      high = "red",
      na.value = "gray",
      limits = c(min_density, max_density) # 統一的顏色範圍
    ) +
    labs(
      title = paste("台南市密度分佈圖 - ", year),
      fill = "密度指數"
    ) +
    theme_minimal() +
    theme(plot.title = element_text(hjust = 0.5))
}
library(ggplot2)
library(patchwork)

```

```

# ===== 獲取所有年份的密度指數範圍 =====
all_density_values <- unlist(lapply(density_results, function(data) {
  return(data$density_index)
}))

# 找到全局最小值和最大值
global_min <- min(all_density_values, na.rm = TRUE)
global_max <- max(all_density_values, na.rm = TRUE)

# ===== 繪製每年的密度分布圖 =====
plots <- list()

for (year in names(density_results)) {
  merged_data <- density_results[[year]]

  # 創建圖表
  p <- ggplot() +
    geom_sf(data = merged_data, aes(fill = density_index), color = "black", lwd = 0.3) +
    scale_fill_gradient(
      low = "green", high = "red", na.value = "gray",
      limits = c(global_min, global_max) # 設定全局的範圍
    ) +
    labs(
      title = paste("Year", year),
      fill = "Density Index"
    ) +
    theme_minimal() +
    theme(
      plot.title = element_text(hjust = 0.5),
      legend.position = "bottom"
    )

  # 將圖存入列表
  plots[[year]] <- p
}

# ===== 使用 patchwork 組合圖表 =====
combined_plot <- (plots[["105"]] | plots[["106"]] | plots[["107"]]) /
  (plots[["108"]] | plots[["109"]] | plots[["110"]]) /
  (plots[["111"]] | plots[["112"]] | plots[["113"]])

# 顯示組合圖
# print(combined_plot)

```

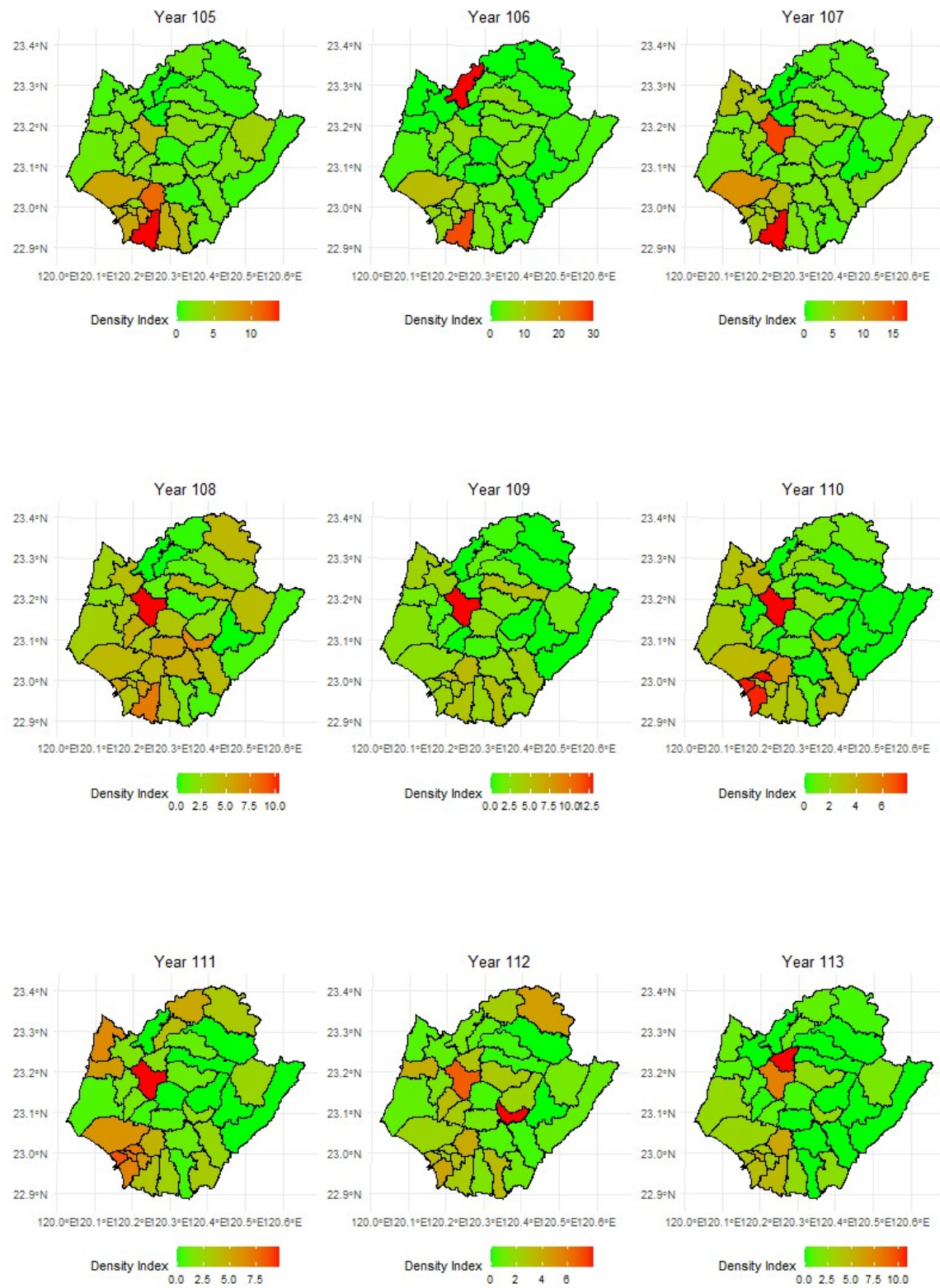


Figure 2: the plot showing the different risk level(region) with different scale.

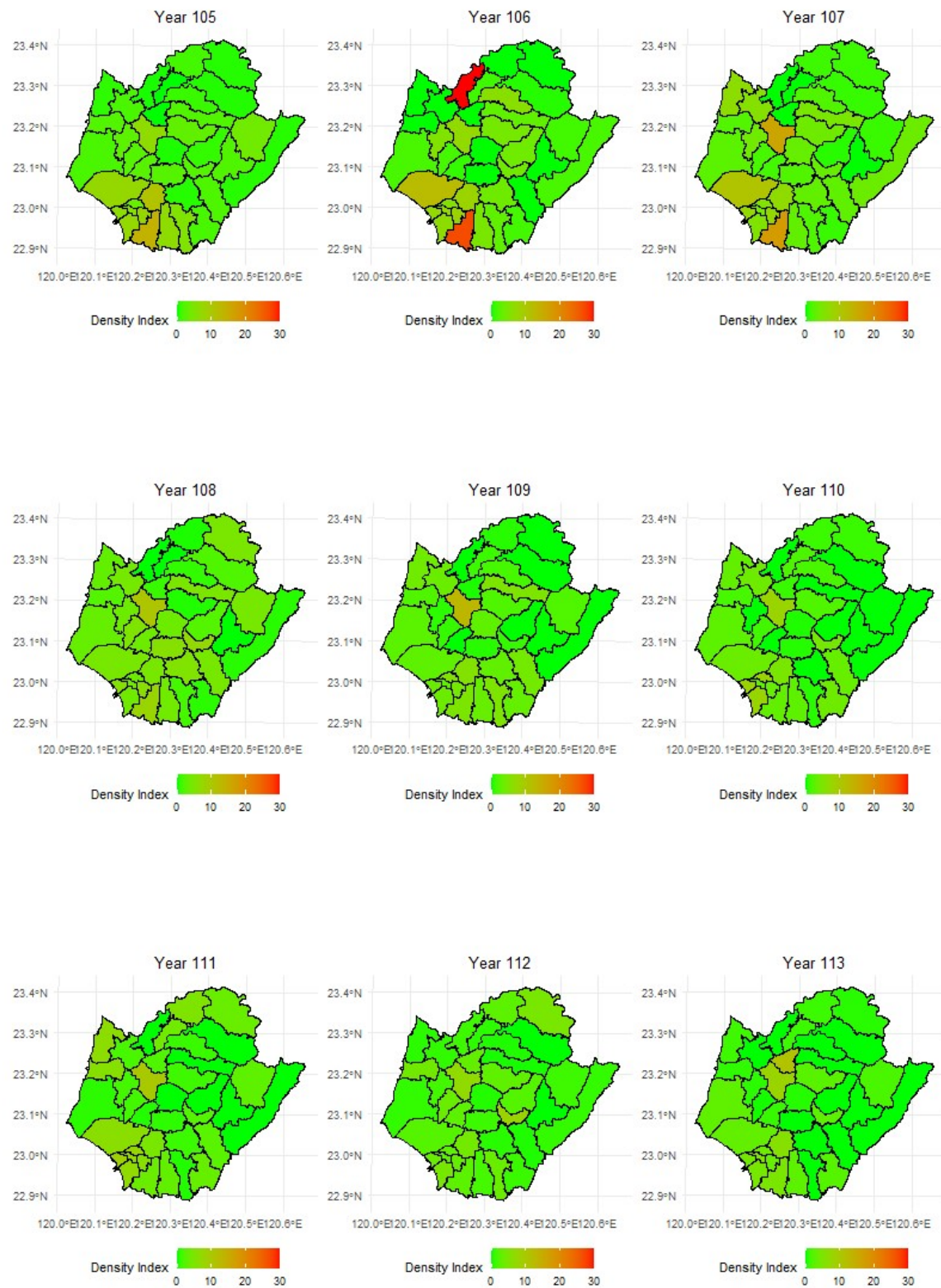


Figure 3: the plot showing the different risk level(region) with same scale(year106-113).

這2張圖代表容器指數在每個地區的嚴重程度，越綠的表示風險越低(容器指數低)，而越紅的代表風險越高(容器指數高)。從第一張圖可知道和前面 the density plot of 105 to 113 相似的結果，而第二張圖可以知道在105到107年時紅色的地區仍舊很多，然而隨著時間來到最近，紅色的地區越來越少，代表著病毒的來源密度越來越少，也就是得到登革熱的風險逐年降低。

點格局識別

Quadrant Analysis(QA)

Quadrant Analysis 和 VMR (Variance-to-Mean Ratio) 是兩個常用於統計與資料分析中的工具。以下是它們的介紹：

Quadrant Analysis 是一種分析點的分布情形之工具，常用於評估點為隨機分布、聚集分布或均勻分布。

在這次的分析中，我們將台南市劃分為9個格子，每個格子的長寬盡量相同，之後便使用 VMR 來判斷點分布。

VMR (Variance-to-Mean Ratio)

VMR 在空間統計中是個用來評估點分布的工具，其公式為：

$$\text{VMR} = \frac{\text{Variance}}{\text{Mean}},$$

where

- $\text{Variance} = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (X_i - \text{Mean})^2}$
- $\text{Mean} = \frac{1}{n} \sum_{i=1}^n X_i$
- X_i 為格子中點之數量

用途：

檢測數據的分佈類型：

- 如果 $\text{VMR} = 1$ ：資料呈現 poisson distribution(隨機分布)。
- 如果 $\text{VMR} < 1$ ：資料是低變異的，可能呈現較像 uniform distribution。
- 如果 $\text{VMR} > 1$ ：資料是高變異的，可能呈現較像 clustered distribution。

優點：

- 簡單易算，特別適合離散數據。
- 可用於評估隨機性或群聚性。

```
data_sf1 <- process_data(data_sf)
positive_data <- data_sf1 %>%
  filter(陽性戶數 > 0)
# positive_data
# summary(data_sf1$經度)
# summary(data_sf1$緯度)
# data_sf1
# 1
xd <- (max(data_sf1$經度) - min(data_sf1$經度)) / 3
yd <- (23.45 - min(data_sf1$緯度)) / 3
x_quartiles <- c(min(data_sf1$經度) + xd, (min(data_sf1$經度) + 2 * xd))
y_quartiles <- c(min(data_sf1$緯度) + yd, (min(data_sf1$緯度) + 2 * yd))
```

```

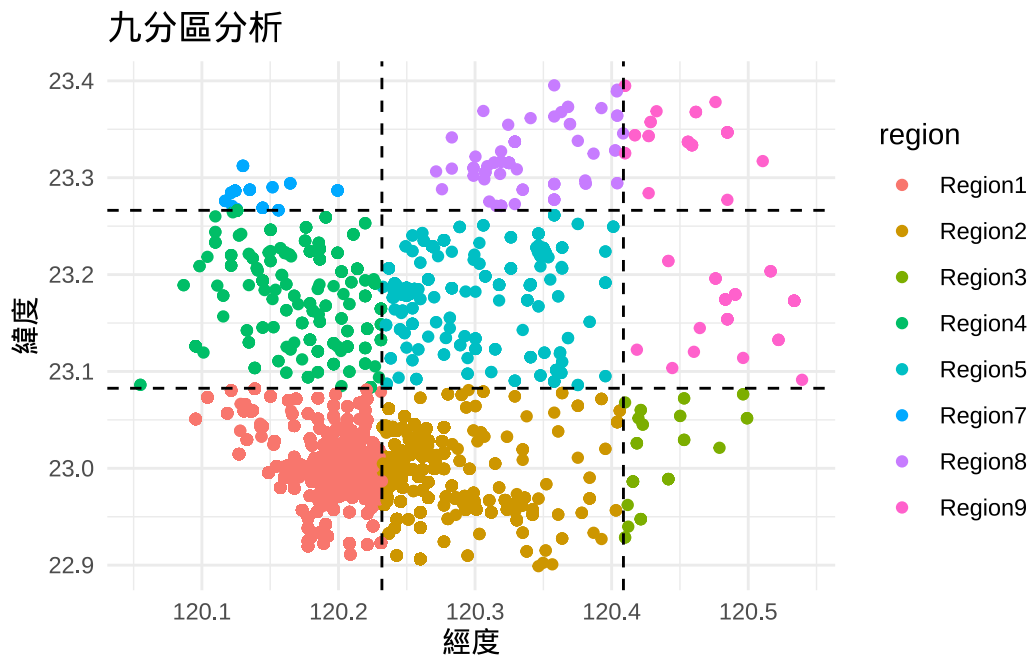
# 添加九區標籤
data_sf1 <- positive_data %>%
  mutate(
    region = case_when(
      經度 < x_quartiles[1] & 緯度 < y_quartiles[1] ~ "Region1",
      經度 >= x_quartiles[1] & 經度 < x_quartiles[2] & 緯度 < y_quartiles[1] ~ "Region2",
      經度 >= x_quartiles[2] & 緯度 < y_quartiles[1] ~ "Region3",
      經度 < x_quartiles[1] & 緯度 >= y_quartiles[1] & 緯度 < y_quartiles[2] ~ "Region4",
      經度 >= x_quartiles[1] & 經度 < x_quartiles[2] & 緯度 >= y_quartiles[1] & 緯度 < y_quartiles[2] ~ "Region5",
      經度 >= x_quartiles[1] & 經度 < x_quartiles[2] & 緯度 >= y_quartiles[1] & 緯度 < y_quartiles[2] ~ "Region6",
      經度 < x_quartiles[1] & 緯度 >= y_quartiles[2] ~ "Region7",
      經度 >= x_quartiles[1] & 經度 < x_quartiles[2] & 緯度 >= y_quartiles[2] ~ "Region8",
      TRUE ~ "Region9"
    )
  )

# 九區統計
region_count <- table(data_sf1$region)

# print(region_count)

# 繪製九區分布圖
ggplot(data_sf1, aes(x = 經度, y = 緯度, color = region)) +
  geom_point() +
  theme_minimal() +
  ggtitle("九分區分析") +
  geom_vline(xintercept = x_quartiles, linetype = "dashed") +
  geom_hline(yintercept = y_quartiles, linetype = "dashed")

```



```

# 計算每個區域的數據點數量
region_counts <- table(data_sf1$region)

```

```
# 計算 VMR
vmr_region <- var(region_counts) / mean(region_counts)

# 輸出結果
cat("Variance-to-Mean Ratio (VMR) for nine regions:", vmr_region, "\n")
```

Variance-to-Mean Ratio (VMR) for nine regions: 10376.35

```
# 解釋分布類型
if (vmr_region > 1) {
  cat("分布呈現聚集特性。 \n")
} else if (vmr_region < 1) {
  cat("分布呈現均勻特性。 \n")
} else {
  cat("分布呈現隨機特性。 \n")
}
```

分布呈現聚集特性。

```
# 使用卡方檢驗檢查數據分布
chisq_test <- chisq.test(region_counts)

# 輸出檢驗結果
print(chisq_test)
```

Chi-squared test for given probabilities

```
data: region_counts
X-squared = 72634, df = 7, p-value < 2.2e-16

if (chisq_test$p.value < 0.05) {
  cat("分布顯著偏離隨機性 ( p-value < 0.05 ) 。 \n")
} else {
  cat("分布未顯著偏離隨機性 ( p-value >= 0.05 ) 。 \n")
}
```

分布顯著偏離隨機性 (p-value < 0.05) 。

從上方的圖可以看到每個點代表著陽性戶數為非零的點，而從這個Quadrant analysis可以看出其陽性戶數並非分布隨機，而是有聚集的效應，此外我們也做了卡方檢定(適合度檢定)來看其陽性戶數的分布是否為隨機(Poisson distribution)。

其結果也如預期為拒絕虛無假設(陽性戶數的分布為隨機分布)。接著我們在看Ripley's K是否也有一樣的結果。

Ripley's K function

Ripley's K-Function 是一種用於分析空間點模式的統計方法，常用於檢測點的空間分布是隨機分布、聚集分布還是均勻分布。以下是它的原理和基本概念：

1. Ripley's K 的定義

Ripley's K 是描述點與點之間空間關係的函數，通過計算在距離 r 內點對的數量來評估點的分布模式。

公式：

$$K(r) = \frac{\lambda^{-1}}{n} \sum_{i=1}^n \sum_{j \neq i} I(d_{ij} \leq r)$$

- n ：點的總數
- λ ：點的密度 ($\lambda = \frac{n}{A}$ ， A 是區域的面積)
- d_{ij} ：點 i 和點 j 之間的距離
- $I(d_{ij} \leq r)$ ：指示函數，當 $d_{ij} \leq r$ 時為 1，否則為 0。

Ripley's K 衡量的是，對於每個點，在距離 r 內有多少其他點。

2. 功能與應用

Ripley's K 函數提供了一個完整的尺度範圍的分析，比單一距離的測度（例如最近鄰距離）更全面。它可以用於：

1. 隨機性檢測：

- 比較觀測到的 $K(r)$ 與理論下的隨機分布（Poisson 分布）的 $K(r)$ 。
- 在隨機分布下，理論值為 $K(r) = \pi r^2$ 。

2. 聚集或均勻性檢測：

- 如果 $K(r) > \pi r^2$ ，表明有聚集模式。
- 如果 $K(r) < \pi r^2$ ，表明有均勻模式。

Result

```
# ===== 載入必要套件 =====
library(dplyr)
library(sf)
library(RANN)
library(spatstat)
library(ggplot2)
library(patchwork)

# ===== 數據準備 =====
# 年份範圍
years <- sprintf("%d", 105:113)

# ===== 進行迴圈處理每年的數據 =====
# ===== 處理與分析 =====

# 儲存每年的圖表
k_func_plots <- list()

# ===== 進行迴圈分析每年的數據 =====
for (year in years) {
  # 動態獲取每年的數據
  yearly_data <- get(paste0("yearly_data_", year))

  # 過濾有陽性戶數的點
  positive_data <- yearly_data %>% filter(陽性戶數 > 0)

  # 檢查數據是否足夠進行分析
  if (nrow(positive_data) < 10) {
    warning(paste("Year", year, "陽性戶數點數量不足，跳過分析。"))
    next
  }
}
```

```

}

# 提取經緯度
coords <- st_coordinates(positive_data)

# 轉換為 ppp 格式
ppp_data <- as.ppp(
  coords,
  W = owin(xrange = range(coords[, 1]), yrange = range(coords[, 2]))
)

# 計算 K 函數
k_func <- Kest(ppp_data)

# 轉換為 data.frame
k_func_df <- as.data.frame(k_func)

# 繪製 K 函數圖表
k_func_plot <- ggplot(k_func_df, aes(x = r)) +
  geom_line(aes(y = border), color = "blue", size = 1) +
  geom_line(aes(y = pi * r^2), color = "red", linetype = "dashed") +
  labs(
    title = paste("Year", year, "陽性戶數 K 函數"),
    x = "距離 (r)",
    y = "K(r)"
  ) +
  theme_minimal()

# 儲存圖表
k_func_plots[[year]] <- k_func_plot
}

# ===== 組合圖表展示 =====
combined_plot <- (k_func_plots[["105"]] | k_func_plots[["106"]] | k_func_plots[["107"]]) /
  (k_func_plots[["108"]] | k_func_plots[["109"]] | k_func_plots[["110"]]) /
  (k_func_plots[["111"]] | k_func_plots[["112"]] | k_func_plots[["113"]])

# 顯示組合圖
# print(combined_plot)

```

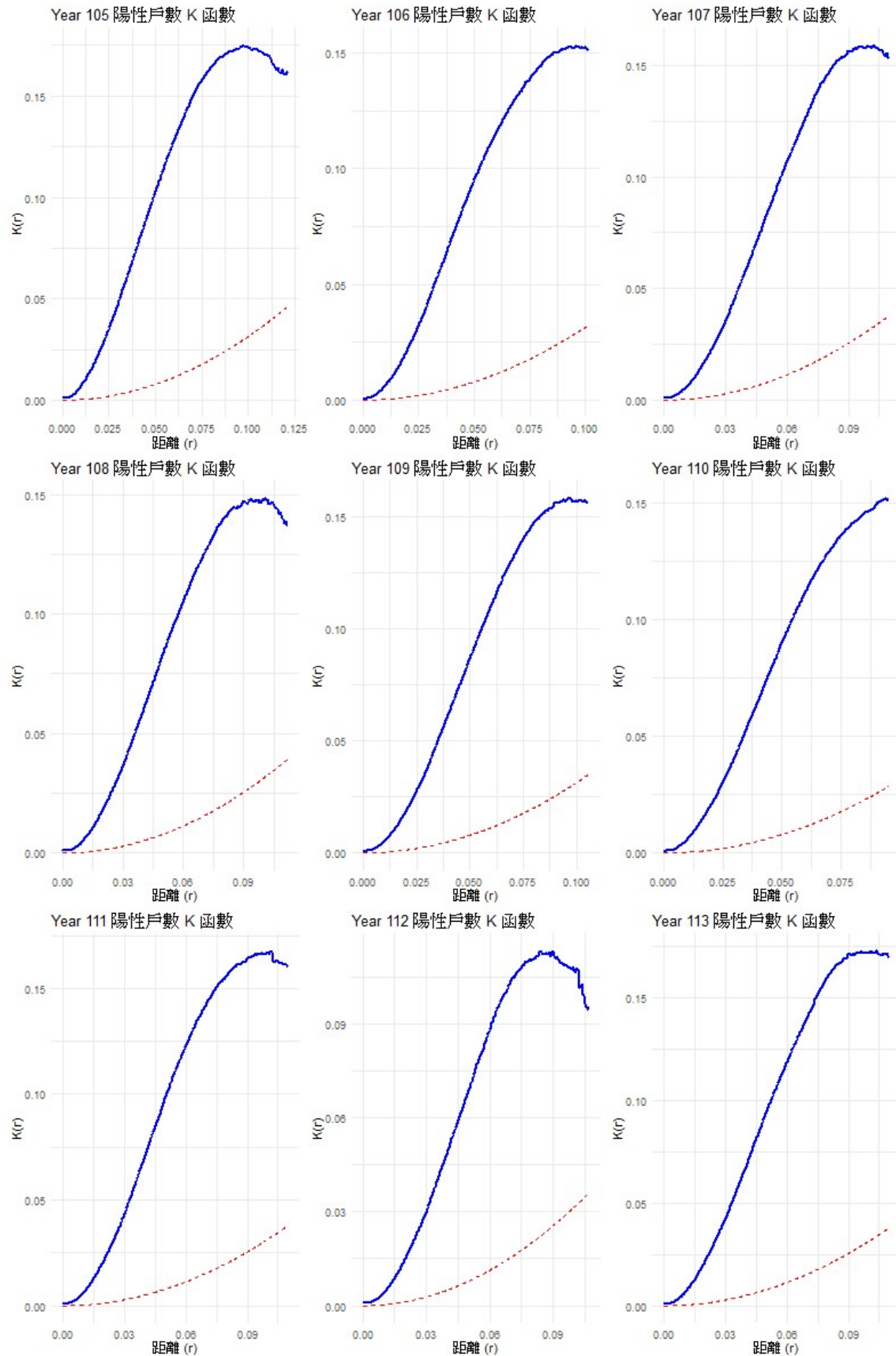


Figure 4: k function plot
19

上方的圖是每一年陽性戶數的k值的圖，藍色的為k值，而紅色的為 πr^2 ， πr^2 為隨機分布的k值。

而從上方的圖可以知道每一年其陽性戶數的分布都是有聚集效應的。在看完點分布後我們來看其點是否有空間自相關性。

格數據統計

這個統計分析主要在看是否有空間自相關，最有名的是這個分析可以畫出熱點圖，來直覺的看出每個地圖上的點的趨勢。

熱點圖主要有2個，一個是 Local Moran's I 和 Getis-ord Gi。

```
# ===== 1. 載入必要的套件 =====
library(Hmisc)
library(DataExplorer)
library(sp)
library(sf)
library(spacetime)
library(gstat)
library(dplyr)
library(lubridate)
library(spdep)
library(ggplot2)
library(RColorBrewer)

# ===== 2. 數據讀取與合併 =====
# 創建年份向量
years <- sprintf("%d", 105:113)

# 使用lapply批量讀取數據
data_list <- lapply(years, function(year) {
  file_name <- paste0(year, ".csv")
  df <- read.csv(file_name)
  df$year <- year
  return(df)
})

# 合併所有數據框
data_combined <- do.call(rbind, data_list)

# ===== 3. 初步數據檢查 =====
# 檢查每個數據框的行數
rows_count <- sapply(data_list, nrow)
names(rows_count) <- paste0("data", years)

# plot_missing(data_combined$容器指數)

# ===== 4. 數據清理與轉換 =====
# 日期轉換
data_combined <- data_combined %>%
  mutate(日期 = as.Date(日期, format = "%Y%m%d")) %>%
  filter(!is.na(日期)) %>%
  arrange(日期)

# 經緯度處理
data_combined <- data_combined %>%
  # 移除非數字字符
```

```

mutate(
  經度 = gsub("[^0-9]", "", 經度),
  緯度 = gsub("[^0-9]", "", 緯度)
) %>%
# 轉換為數字並處理經緯度順序
mutate(
  經度 = ifelse(as.numeric(經度) < 90,
    as.numeric(緯度),
    as.numeric(經度)),
  緯度 = ifelse(as.numeric(緯度) > 90,
    as.numeric(經度),
    as.numeric(緯度))
) %>%
# 移除NA值
filter(!is.na(經度) & !is.na(緯度)) %>%
# 確保經緯度在合理範圍內
filter(經度 >= 120 & 經度 <= 122) %>%
filter(緯度 >= 21.5 & 緯度 <= 23.45)

data_combined <- data_combined %>%
mutate(
  經度 = as.numeric(經度),
  緯度 = as.numeric(緯度),
  year = as.numeric(year)
)

# 時間分析
data_combined <- data_combined %>%
mutate(
  year = year(日期),
  month = month(日期),
  day = day(日期)
)

# ===== 5. 空間分析 =====
# ===== 5. 空間分析 =====

# 創建空間對象
# 移除經度或緯度為NA的行
data_combined_clean <- data_combined %>%
  filter(!is.na(經度) & !is.na(緯度) & !is.na(陽性戶數))

# 移除重複的坐標點
data_combined_clean <- data_combined_clean %>%
  distinct(經度, 緯度, .keep_all = TRUE)

coordinates(data_combined_clean) <- ~經度+緯度

# 創建鄰接矩陣
nb <- dnearneigh(coordinates(data_combined_clean), 0, 0.1) # 擴大距離閾值
listw <- nb2listw(nb, style = "W", zero.policy = TRUE)

# 計算 Local Moran's I

```

```

data_combined_clean$z_陽性戶數 <- as.numeric(scale(data_combined_clean$陽性戶數
, center = TRUE, scale = TRUE))

local_moran <- localmoran(data_combined_clean$z_陽性戶數, listw)

# 將結果加入數據框，並標準化 Local Moran's I
data_combined_clean$local_moran_l_raw <- local_moran[,1]
data_combined_clean$local_moran_l <- local_moran[,1] / max(abs(local_moran[,1])) # 標準化至 [-1, 1]

# 計算 Getis-Ord Gi*
gi_star <- localG(data_combined_clean$z_陽性戶數, listw)

# 將結果加入數據框
data_combined_clean$gi_star <- gi_star

# 將空間對象轉換回數據框
data_combined_clean_df <- as.data.frame(data_combined_clean)

# 確保列名是字符向量
colnames(data_combined_clean_df) <- make.names(colnames(data_combined_clean_df))

# ===== 6. 載入行政區邊界資料 =====
tainan_boundary <- st_read("tainan_town.shp")

```

Reading layer `tainan_town' from data source
`C:\Users\user\Desktop\Textbook\4-1\stat_consult\final_report\disease\final_report_kcchen\tainan_town.shp'
using driver `ESRI Shapefile'
Simple feature collection with 37 features and 5 fields
Geometry type: POLYGON
Dimension: XY
Bounding box: xmin: 149903.2 ymin: 2531973 xmax: 214819.9 ymax: 2590210
Projected CRS: Transverse_Mercator

```

# 數據轉換為 sf 對象
data_sf <- st_as_sf(data_combined_clean, coords = c("經度", "緯度"), crs = 4326)
st_crs(data_sf) <- 4326

# 確保數據和邊界的 CRS 一致
if (st_crs(data_sf) != st_crs(tainan_boundary)) {
  tainan_boundary <- st_transform(tainan_boundary, st_crs(data_sf))
}

```

Local Moran's I

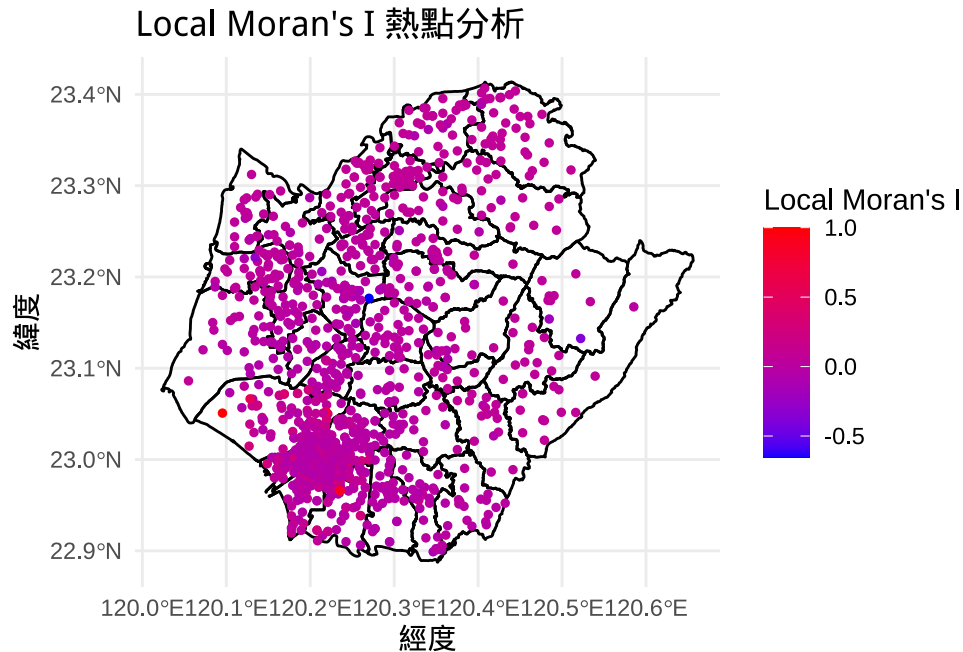
```

# ===== 3. Local Moran's I 熱點分析繪圖 =====
local_moran_plot <- ggplot() +
  # 繪製行政區邊界
  geom_sf(data = tainan_boundary, fill = NA, color = "black", lwd = 0.5) +
  # 繪製 Local Moran's I 熱點點位
  geom_sf(data = data_sf, aes(color = local_moran_l), size = 1) +
  scale_color_gradient(low = "blue", high = "red", name = "Local Moran's I") +
  labs(
    title = "Local Moran's I 熱點分析",

```

```
x = "經度",
y = "緯度"
) +
theme_minimal()
```

```
# 顯示 Local Moran's I 熱點分析圖
print(local_moran_plot)
```



以上的圖片是local moran's i 的圖形，越紅就代表其點與附近(距離<0.1(其單位))越相似，月藍則與鄰近的點差異越大。

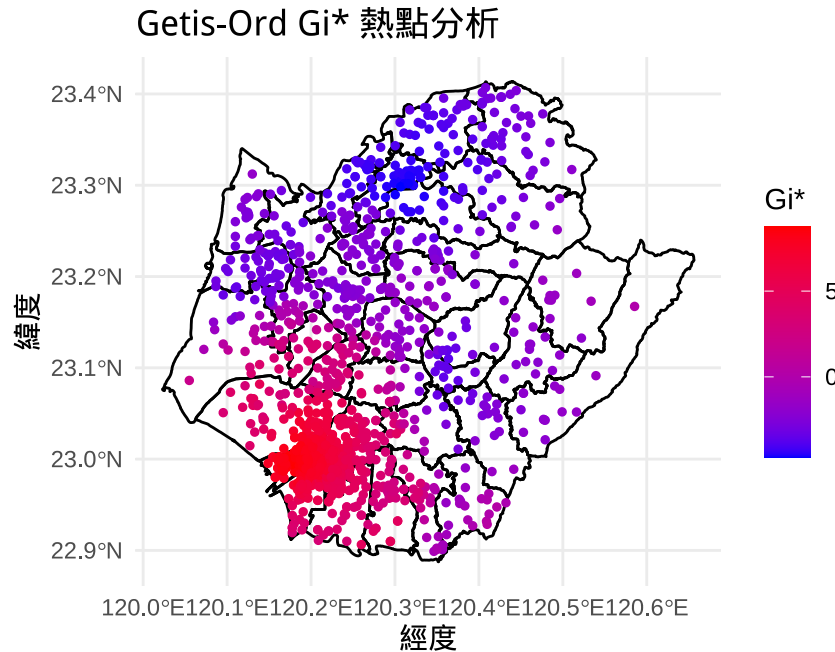
從以上圖片可以發現這個圖大部分的點為紫色偏紅，可以知道這個登革熱資料有一定程度的空間自相關。接著我們做Getis-Ord Gi(另一個常用的熱點圖)。

Getis-Ord Gi

```
# ===== 4. Getis-Ord Gi* 熱點分析繪圖 =====
# 確保 gi_star 為數值類型
data_sf$gi_star <- as.numeric(data_sf$gi_star)

gi_star_plot <- ggplot() +
  # 繪製行政區邊界
  geom_sf(data = tainan_boundary, fill = NA, color = "black", lwd = 0.5) +
  # 繪製 Getis-Ord Gi* 熱點點位
  geom_sf(data = data_sf, aes(color = gi_star), size = 1) +
  scale_color_gradient(low = "blue", high = "red", name = "Gi*") +
  labs(
    title = "Getis-Ord Gi* 熱點分析",
    x = "經度",
    y = "緯度"
  ) +
  theme_minimal()
```

```
# 顯示 Getis-Ord Gi* 熱點分析圖  
print(gi_star_plot)
```



以上這張圖是 Getis-Ord Gi 熱點分析圖，越紅色的點代表他與附近的點相比下，此點之數值高於附近的點，而越藍色的點也就是此點相比於附近的點，其數值較低。

從這張圖可發現以台南的西南部為中心，其點之顏色隨著與中心的距離越遠，其顏色從紅色逐步轉為藍色。有此現象可以知道其中心點屬於登革熱高風險區域。

Model

空間滯後效應 (Spatial Lag Effect)

空間滯後效應是指在空間數據中，某一地點的觀察值會受到其鄰近地點的觀察值影響。這種效應反映了地理或空間上的相互依賴性，即一個地點的變量不僅由當地的特徵決定，還可能受到與之相鄰的地點的特徵影響。

在空間統計學中，空間滯後模型 (Spatial Lag Model) 是一種常用的方法來描述這種空間依賴性。這些模型通常包括一個空間滯後項，用來捕捉鄰近地區的影響。

空間滯後效應的例子

假設你正在研究某個地區的登革熱病例數，並且知道病例數不僅取決於該地區的因素，還可能受到周邊地區登革熱流行情況的影響。例如，如果某個城市的鄰近地區登革熱病例增加，這可能會導致該城市的病例數也上升，因為蚊媒可能跨區域傳播病毒，或因人口流動增加了感染風險。這種現象即為空間滯後效應。

空間滯後模型的形式

在數學表達上，空間滯後模型通常是這樣的：

$$y_i = \rho \sum_{j \in W_i} w_{ij} y_j + X_i \beta + \epsilon_i$$

其中：

- y_i 是地點 i 的應變量。
- ρ 是空間滯後係數，表示鄰近地點對地點 i 的影響程度。
- $\sum_{j \in W_i} w_{ij} y_j$ 是鄰近地點 j 的應變量的加權和， w_{ij} 是加權矩陣的元素，表示地點 i 和地點 j 之間的關聯程度。
- X_i 是地點 i 的解釋變量。
- β 是解釋變量的回歸係數。
- ϵ_i 是誤差項。

相鄰的定義

在程式碼中，相鄰的定義是基於k-最近鄰 (k-nearest neighbors) 方法來確定的。這部分的程式碼如下：

- **k-最近鄰 (k-nearest neighbors)**：這裡使用 $k = 4$ ，表示對每個地理點（每個區域），找出其最近的4個鄰近點（根據空間坐標）。
- **knn2nb**：這個函數將從k-最近鄰中建立一個鄰接矩陣 (nb)，該矩陣告訴你哪些區域彼此相鄰（即，哪些區域的距離在4個最近鄰範圍內）。
- **listw**：這是將鄰接矩陣轉換為空間權重矩陣的步驟，`style = "W"`表示使用標準化的權重矩陣（每行的總權重為1），這樣可以消除因為區域數量不同而導致的規模效應。

結果詮釋

- **相鄰區域的影響**：該模型假設應變量（如陽性戶數）不僅受區域內的因素影響，還會受到鄰近區域的影響。這些鄰近區域的影響通過空間滯後項 $\rho \sum_{j \in N(i)} W_{ij} Y_j$ 被引入。
- **鄰接矩陣**：在這段程式碼中，相鄰是基於空間坐標進行定義的，每個區域選取其最接近的4個鄰區作為相鄰區域。這意味著每個區域的應變量會受到這些相鄰區域的影響。

這種基於空間滯後模型的分析，可以用來分析和捕捉地理區域之間的空間自相關，並衡量某些地理變數對周邊區域的影響。

空間滯後效應的解釋

- **正向影響**：如果 $\rho > 0$ ，則表示鄰近地點對地點 i 的影響是正向的，即當鄰近地區的變量上升時，地點 i 的變量也會上升。
- **負向影響**：如果 $\rho < 0$ ，則表示鄰近地區對地點 i 的影響是負向的，即當鄰近地區的變量上升時，地點 i 的變量會下降。

為何需要考慮空間滯後效應？

在許多現實情況中，地理鄰近的地點會有互動關係，忽略這些關聯可能會導致模型錯誤和解釋不當。例如：

- **登革熱防疫**：鄰近地區的疫情情況可能影響本地疫情的發展，透過分析空間滯後效應，可以更有效地制定防疫措施。

空間滯後效應幫助我們理解並量化空間關係對研究對象的影響，是空間數據分析中非常重要的一部分。

```
# ===== 載入必要套件 =====
```

```
library(dplyr)
library(sf)
library(spatialreg)
library(spdep)
```

```

# ===== 年份範圍 =====
years <- sprintf("%d", 105:113)

# 儲存回歸結果
results <- list()

# ===== 迴圈處理每年的資料 =====
# ===== 迴圈處理每年的資料 =====
for (year in years) {
  # 動態獲取每年的數據
  yearly_data <- get(paste0("yearly_data_", year))

  # 移除重複點
  yearly_data <- yearly_data %>% distinct(geometry, .keep_all = TRUE)

  # 提取經緯度
  coords <- st_coordinates(yearly_data)
  yearly_data$經度 <- coords[, 1]
  yearly_data$緯度 <- coords[, 2]

  # 確保數值類型並過濾 NA
  yearly_data <- yearly_data %>%
    mutate(
      經度 = as.numeric(經度),
      緯度 = as.numeric(緯度),
      容器指數 = as.numeric(容器指數),
      布氏指數 = as.numeric(布氏指數),
      布氏級數 = as.numeric(布氏級數),
      陽性容器合計 = as.numeric(陽性容器合計),
      陽性戶數 = as.numeric(陽性戶數)
    ) %>%
    filter(
      !is.na(經度), !is.na(緯度),
      !is.na(容器指數), !is.na(布氏指數),
      !is.na(布氏級數), !is.na(陽性容器合計),
      !is.na(陽性戶數)
    )

  # 如果資料點不足，跳過該年份
  if (nrow(yearly_data) < 10) {
    warning(paste("Year", year, "資料點數量不足，跳過回歸分析"))
    next
  }

  # 創建鄰接矩陣並轉換為權重列表
  nb <- knn2nb(knearneigh(coords, k = 4))
  listw <- nb2listw(nb, style = "W")

  # 空間滯後模型
  lag_model <- lagsarlm(
    陽性戶數 ~ 經度 + 緯度 + 陽性容器合計 + 布氏指數 + 容器指數,
    data = yearly_data,
    listw = listw
  )
}

```

```

)

# 儲存模型結果 ( 僅保存 lag_model )
results[[year]] <- list(
  lag_model = summary(lag_model)
)
}
# results

library(knitr)
# 創建數據框
data <- data.frame(
  年份 = c(105, 105, 105, 105, 105, 106, 106, 106, 106, 106, 107, 107, 107, 107, 107,
    108, 108, 108, 108, 108, 109, 109, 109, 109, 109, 110, 110, 110, 110, 110,
    111, 111, 111, 111, 111, 112, 112, 112, 112, 112, 113, 113, 113, 113, 113),
  變數 = c("經度", "緯度", "陽性容器合計", "布氏指數", "容器指數",
    "經度", "緯度", "陽性容器合計", "布氏指數", "容器指數",
    "經度", "緯度", "陽性容器合計", "布氏指數", "容器指數",
    "經度", "緯度", "陽性容器合計", "布氏指數", "容器指數",
    "經度", "緯度", "陽性容器合計", "布氏指數", "容器指數",
    "經度", "緯度", "陽性容器合計", "布氏指數", "容器指數",
    "經度", "緯度", "陽性容器合計", "布氏指數", "容器指數",
    "經度", "緯度", "陽性容器合計", "布氏指數", "容器指數",
    "經度", "緯度", "陽性容器合計", "布氏指數", "容器指數"),
  估計值 = c(-0.12, -0.12, 0.63, -0.006, 0.01,
    -0.12, -0.2, 0.85, -0.18, 0.12,
    -0.05, -0.04, 0.22, 0.26, 0.03,
    0.02, -0.03, 0.44, 0.2, 0.03,
    -0.04, -0.06, -0.24, 0.45, 0.06,
    -0.03, -0.02, 1.01, 0.13, 0.04,
    0.03, -0.02, 0.74, 0.06, 0.04,
    0.05, -0.04, 0.16, 0.30, 0.03,
    0.02, -0.01, 1.78, -0.67, 0.11),
  p_value = c(0.5, 0.35, "< 0.001", "< 0.001", "< 0.001",
    0.49, 0.1, "< 0.001", "< 0.001", "< 0.001",
    0.64, 0.55, "< 0.001", "< 0.001", "< 0.001",
    0.76, 0.52, "< 0.001", "< 0.001", "< 0.001",
    0.62, 0.29, 0.03, "< 0.001", "< 0.001",
    0.59, 0.71, "< 0.001", "< 0.001", "< 0.001",
    0.39, 0.49, "< 0.001", "< 0.001", "< 0.001",
    0.42, 0.41, 0.01, "< 0.001", "< 0.001",
    0.64, 0.63, "< 0.001", "< 0.001", "< 0.001"),
  備註 = c("", "", "****", "****", "****",
    "", "", "****", "****", "****",
    "", "", "****", "****", "****",
    "", "", "****", "****", "****",
    "", "", "****", "****", "****",
    "", "", "****", "****", "****",
    "", "", "****", "****", "****",
    "", "", "****", "****", "****",
    "", "", "****", "****", "****")
)

# 使用 knitr::kable() 生成表格

```

```
kable(data, caption = "各年份模型之獨立變數表", align = "c", digits = 2)
```

Table 1: 各年份模型之獨立變數表

年份	變數	估計值	p_value	備註
105	經度	-0.12	0.5	
105	緯度	-0.12	0.35	
105	陽性容器合計	0.63	< 0.001	***
105	布氏指數	-0.01	< 0.001	***
105	容器指數	0.01	< 0.001	***
106	經度	-0.12	0.49	
106	緯度	-0.20	0.1	
106	陽性容器合計	0.85	< 0.001	***
106	布氏指數	-0.18	< 0.001	***
106	容器指數	0.12	< 0.001	***
107	經度	-0.05	0.64	
107	緯度	-0.04	0.55	
107	陽性容器合計	0.22	< 0.001	***
107	布氏指數	0.26	< 0.001	***
107	容器指數	0.03	< 0.001	***
108	經度	0.02	0.76	
108	緯度	-0.03	0.52	
108	陽性容器合計	0.44	< 0.001	***
108	布氏指數	0.20	< 0.001	***
108	容器指數	0.03	< 0.001	***
109	經度	-0.04	0.62	
109	緯度	-0.06	0.29	
109	陽性容器合計	-0.24	0.03	*
109	布氏指數	0.45	< 0.001	***
109	容器指數	0.06	< 0.001	***
110	經度	-0.03	0.59	
110	緯度	-0.02	0.71	
110	陽性容器合計	1.01	< 0.001	***
110	布氏指數	0.13	< 0.001	***
110	容器指數	0.04	< 0.001	***
111	經度	0.03	0.39	
111	緯度	-0.02	0.49	
111	陽性容器合計	0.74	< 0.001	***
111	布氏指數	0.06	< 0.001	***
111	容器指數	0.04	< 0.001	***
112	經度	0.05	0.42	
112	緯度	-0.04	0.41	
112	陽性容器合計	0.16	0.01	*
112	布氏指數	0.30	< 0.001	***
112	容器指數	0.03	< 0.001	***
113	經度	0.02	0.64	
113	緯度	-0.01	0.63	
113	陽性容器合計	1.78	< 0.001	***
113	布氏指數	-0.67	< 0.001	***
113	容器指數	0.11	< 0.001	***

這個表顯示了幾個現象:

- 經度和緯度：這些變數顯示了地理位置的影響。而之前顯示之結果意味著地理位置對這些年份的變數影響相對較小，我們推測是因為其數據僅有台南地區，而台南的地區差異並沒有很大，其下雨量及溫度都沒有明顯的差異。
- 陽性容器合計：這個變數在大部分年份中顯示出強烈的統計顯著性（ p 值均小於0.001），並且估計值較大和其大部分的年份為正的，表示陽性容器合計對目標變數有顯著影響，可能是指與陽性容器相關的因素對結果有顯著的正向關聯。
- 布氏指數：這個指數在所有年份中也顯示出顯著的影響（ p 值通常小於0.001）。它的估計值的正負號表示了布氏指數對目標變數的-正向或負向影響，通常與空間環境或特徵的多樣性相關。
- 容器指數：這個指數也在所有年份中顯示出顯著影響（ p 值通常小於0.001）且正向，表明容器數量或類型的變化對目標變數有顯著的影響。

```
# 創建數據框
data <- data.frame(
  年份 = c(105, 106, 107, 108, 109, 110, 111, 112, 113),
  Rho = c(0.09, 0.03, 0.09, 0.03, 0.03, 0.02, 0.01, 0.02, 0.01),
  `LR Test 值` = c(10.39, 2.71, 19.69, 3.31, 1.58, 1.95, 0.77, 1.24, 0.18),
  `LR Test p 值` = c(0.00, 0.10, "< 0.001", 0.07, 0.21, 0.16, 0.38, 0.27, 0.67),
  `Wald 檢定值` = c(11.14, 2.69, 20.10, 3.22, 1.67, 1.94, 0.74, 1.24, 0.18),
  `Wald 檢定 p 值` = c(0.00, 0.10, "< 0.001", 0.07, 0.20, 0.16, 0.39, 0.27, 0.67)
)

# 使用 knitr::kable() 生成表格**
kable(data, caption = "各年度之空間滯後項係數表", align = "c", digits = 2)
```

Table 2: 各年度之空間滯後項係數表

年份	Rho	LR.Test.值	LR.Test.p.值	Wald.檢定值	Wald.檢定.p.值
105	0.09	10.39	0	11.14	0
106	0.03	2.71	0.1	2.69	0.1
107	0.09	19.69	< 0.001	20.10	< 0.001
108	0.03	3.31	0.07	3.22	0.07
109	0.03	1.58	0.21	1.67	0.2
110	0.02	1.95	0.16	1.94	0.16
111	0.01	0.77	0.38	0.74	0.39
112	0.02	1.24	0.27	1.24	0.27
113	0.01	0.18	0.67	0.18	0.67

從各年度之空間滯後項係數表中可以知道在前三年(民國105-107年)的 ρ 值都較大，而越接近最近幾年的 ρ 值越來越低，這展現了空間滯後的效果越來越不顯著，我們認為其原因主要為：

- 台南防疫措施有效，使得區域間的疾病傳播可能受到控制，導致空間滯後效應減弱。包括：
 - 及時噴藥阻止了病例的地理擴散。
 - 健康宣傳、蚊蟲消滅等措施降低了疾病在鄰近地區的傳播風險。
 - 隨著時間推進， ρ 檢定結果的顯著性下降，可能反映了這些防疫工作的逐漸成效。