

# 疾病傳播與應對政策之模擬分析

## The Simulation of COVID-19

組員：陳昱詮、謝富伍、卓彥霖

### 研究動機

在過去一年，COVID-19 在全球範圍大肆傳播，使全球人們受到了巨大損失以及不便。這段期間，各國政府採取了許多不同的措施如：保持社交距離、戴口罩、隔離檢疫、限制旅行等方法，期望能防止疫情擴大而造成更大的傷害。因此我們希望能夠藉由程式模擬流行病在各種條件下的傳播情況，並以此為基礎預測及分析各式預防措施對於流行病增長情形的可能效果，希望以此作為與疫情對抗的參考。

### 數值分析

#### ● 基本設定

SIR 模型：

1. 將人群分為未感染的人 (Susceptible, S)、感染者 (Infectious, I) 和已從感染中恢復的人 (Recovered, R)。
2. 每單位時間中，未感染的人處於感染者的傳染半徑 (infection radius,  $r$ ) 時，便有一定機率 (probability,  $p$ ) 遭受感染。以此方式來簡單量化現實中人與人之間的各種互動可能造成的傳染，例如如果人們更常洗手、戴口罩代表傳染半徑、感染機率會因此變小。
3. 感染者在一段時間後便會從 Covid-19 中康復，並且不再傳播疾病。

初始參數設定：

- (a) 以顏色區分不同狀態的人，未感染的人(S)為藍色、感染者(I)為紅色、已從感染中恢復的人(R)為灰色。
- (b) 感染半徑  $r =$
- (c) 感染機率  $p =$

#### ● 行為模式

1. 隨機行走：  
所有人往各方向的機率皆相同 (除非會超出邊界)。
2. 社區間移動：  
存在許多的社區 (只會在社區中移動、和社區內的人有接觸)，存在一定機率(旅行率)會有部分人移動至其他社區。  
在病例數高於一定值後，將旅行率減少，代表當疫情升溫的時候人們會減少外出、在不同社區間的移動。

初始參數設定：

- (a) 旅行率  $travel\_rate =$
- (b) 觸發旅行率減少之病例數  $trigger\_number =$

● 識別、隔離和治療

1. 假設我們有良好的篩檢和隔離，一旦病例達到的某個值（表現出症狀），我們便會在他們感染 Covid-19 後的 N 天開始，將感染者送到一個單獨的區域（對應現實中的隔離）。
2. 感染者也有可能因為沒有表現出任何症狀並且沒有接受檢測而不會被隔離。
3. 假設存在病毒解藥，我們可以降低感染持續時間。此方法與隔離案例具有相同的效果。

初始參數設定：

(a) 感染者未受隔離比例

(b) 感染者具感染力時間 Infection\_duration =

● 社交距離

1. 在病例數高於一定值後，我們將社交距離因子作為人與人之間的“排斥力”，當人們與鄰近的人之間的距離太近時，會有一互斥的傾向並以將個體顏色改變表示出來，以此模擬（政策使然或是自主的）社交距離。
2. 當然有些人可能會不遵守社交距離，因此在不同地方履行社交距離的人口比例有所不同。
3. 我們設置一個社交距離因子，以此判斷是否有社交距離以及做為控制社交距離斥力大小的係數，此為一無因次之係數。

初始參數設定：

(a) 社交距離因子 Social\_distance\_factor =

(b) 人群中遵守社交距離的比例：

Percentage\_of\_social\_distancing\_population =

● 傳播率

考慮一個個體在受到感染後、具有感染力期間將疾病傳染給多少人，計算此數值在群體中的平均數，以此判斷此傳染病現階段的傳播狀況。

$$R = \frac{\text{感染者人數}}{\text{感染他人的人人數}}$$

$R > 1$  時，感染人數成指數型增長，表示處於大流行階段。

當  $R$  穩定在 1 左右時，表示疫情受控制。

如果  $R < 1$ ，則表示疫情趨緩（也有可能代表社會中幾乎所有人皆已染疫）。

## 程式架構

### ● Class Person

#### 1. 初始位置設置

程式一開始使所有點在各個社區中為隨機分布。

分別沿著 x, y 方向設置上下限，形成一個矩形的邊界。將個體的 x, y 座標值以 random 函數分別產生一個沿 x, y 方向、介於上下限之間的隨機數寫入，將此 x, y 座標值以陣列形式寫入自身位置，回傳至自身位置。

#### 2. 更新位置機制

藉由個體所受合力的形式，來控制各個體的行動。

每經過時間微分量 dt，個體之位移 x 以及速度 v 之微分量分別為：

$$dx = dv \times dt$$

$$dv = a \times dt$$

$$a = \frac{F}{m}$$

其中設  $m = 1$ 、F 為合力、a 為加速度，並且若計算過程中個體速度 speed

大於設定之速度上限 max\_speed，則使其乘以  $\frac{\text{max\_speed}}{\text{speed}}$  來令任一個體速度

不會無上限的增加。

#### 3. 各分力之作用機制

##### I. Gravity well 重力井(暫譯)

每經過一段時間(wander\_step\_duration)更新，隨機指向一個方向之特定位置並受距離平方反比的力。

實際做法

當前時刻距離上一次改變重力井作用力計算時時刻若大於

wander\_step\_duration，則開始此次重力井作用力計算。產出一角度隨機之單位向量，並令重力井位於由自身方向沿指向此單位向量、距離

wander\_step\_size 處。重力井位置減去自身位置後計算此長度(即距重力井

之距離)。最後計算  $\frac{\text{位置差}^3}{\text{距離}}$  即為我們希望得到之與距離平方成反比的力，然後

加入合力。

##### II. Wall avoid 邊界作用

為了讓所有個體都能在特定範圍中活動，我們設定邊界以及讓個體受到一個與邊界距離成反比的力來緩衝

實際做法

###### (a) 防止超出邊界

由於各點以一個含二數的數列 [x y] 為其座標表示，因此分別取此數列之第一項與第二項(python 中為第零項與第一項)，即在 x 與 y 方向的位置。

若此位置減去邊界最小值為負數，表示個體位置已超出下邊界，則將其速度設為當前速度之絕對值(正值)，使其在此方向上向範圍內反彈。

若邊界最大值減去此位置為負數，表示個體位置已超出上邊界，則將其速度設為當前速度之負絕對值(負值)，使其在此方向上向範圍內反彈。

(b) 邊界緩衝力

給予個體一個與邊界距離倒數成比例的力後加入合力。

III. 社交距離排斥力

將社交距離排斥力設置為一個零向量，對於處在保持距離範圍內的個體，當社交距離係數不為零時，該個體位置減去自身位置後計算此長度，若此距離較最短距離小，則更新此為最短距離(初始為無限大)，

然後總社交距離排斥力增加一個由該點指向自身、大小為 $\frac{\text{社交距離係數}}{\text{距離}^2}$ 的

斥力，最後將總社交距離排斥力加入合力。

4. 感染半徑

若為感染者(或是前一瞬間  $dt$  受感染而成為感染者)，則感染半徑的有無、大小會在此更新。

● Class SIR\_Simulation

1. 建立社區列表

建立一個空列表作為社區列表，將第  $n$  個社區依次加入社區列表直到  $n$  為設定的總社區數量，將此列表寫入社區列表，回傳至社區列表。

2. 在社區中加入個體

在第  $n$  個社區中，將第  $i$  個個體依次加入社區直到  $i$  為設定的社區人口數，加入的個體屬於 Person 類別。將個體加入城市中個體列表。

3. 感染源

在社區中隨機選取一個個體，將其作為第一位感染者，其感染狀態設為 I。

4. 個體感染狀態更新

將未感染的人與感染者各自作一個空列表，一一檢視第  $n$  個社區中的第  $i$  個個體，狀態為 S 的加入未感染的人列表，狀態為 I 的加入感染者列表，以 linalg 得出 S 點與其他 I 點距離，若此距離小於感染半徑且由 random 函數生成的 0~1 間的隨機數小於感染機率，則將該狀態為 I 的個體感染他人人數+1，並且將該此狀態為 S 的個體狀態更新為 I，代表未感染的人在與感染者接觸後受感染。狀態為 I 的個體若時間減去感染開始時間大於感染期，則將其狀態改為 R，代表感染者已復原。

5. 旅行率

會有部分人移動到不同的社區。

用 random 函數生成的 0~1 間的隨機數若是小於旅行率，則隨機選擇一個社區定為目的地，將旅行者加入選擇社區，並消去原社區列表中的旅行者。

6. 社交距離排斥力

考慮現實中人與人之間的距離，一個人實際能注意並保持刻意保持距離的人數有限——假設為十人，因此我們在計算因社交距離而造成的互斥力時，便只會計算與自身距離最近的十人。

將單一社區中的各個體位置(x,y)列表，把這些位置訂為斥力中心，若是此時社交距離係數大於零，則以其他點位置減自身斥力中心位置，用 linalg 得出與其他各點距離，將此結果以 argsort 由值最小的開始排列，取十個來計算其他人對自身造成的社交距離斥力。

7. 感染狀態人數紀錄

建立一個三個元素的零陣列，第一個元素代表 S 狀態人數、第二個元素代表 I 狀態人數、第三個元素代表 R 狀態人數。依次檢視城市中各個體的狀態，若一個個體狀態為 S，則將第一個元素(S 狀態人數)數值加一，以此類推。最後此陣列為各感染狀態總人數的陣列，將此陣列回傳至個體感染狀態人數紀錄陣列。