

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC - KỸ THUẬT MÁY TÍNH



MÔ HÌNH HÓA TOÁN HỌC (CO2011)

Bài tập lớn

“Mô hình SIR trong dự báo COVID-19”

GVHD: Nguyễn An Khương
Nguyễn Tiến Thịnh

SV thực hiện: Nguyễn Huỳnh Hữu Khiêm – 1810998
Nguyễn Anh Tú – 1920075
Trần Đình Đăng Khoa – 1810248
Lê Quang Duy – 1811709
Đặng Huỳnh Minh Trí – 1810610

Tp. Hồ Chí Minh, Tháng 7/2020



Mục lục

1	Bài toán 1	2
1.1	Mô hình SIR rời rạc	2
1.2	Mô hình SIR liên tục	5
1.2.1	Phương pháp xấp xỉ Euler	6
1.3	Những vấn đề liên quan đến mô hình SIR	6
1.4	Mở rộng của mô hình SIR - mô hình SIRD	7
2	Bài toán 2	8
2.1	Mô tả	8
2.2	Viết chương trình sử dụng thuật toán Euler tìm nghiệm của hệ SIR.	8
2.2.1	Ví dụ 1 mô hình SIR	8
2.2.2	Kết quả ví dụ 1	10
2.2.3	Nhận xét ví dụ 1	10
2.2.4	Ví dụ 2 mô hình SIRD	11
2.2.5	Kết quả ví dụ 2	12
2.2.6	Nhận xét ví dụ 2	12
3	Bài toán 3	14
3.1	Mô tả	14
3.2	Giải	14
3.3	Tổng hợp Code	15
3.4	Kết quả	16
4	Bài toán 4	17
4.1	Mô tả	17
4.2	Hiện thực code	17
4.3	Kết quả và đánh giá	19
	Tài liệu	20

1 Bài toán 1

Trình bày lại chi tiết cách xây dựng mô hình SIR (cả trường hợp rời rạc lẫn liên tục) hoặc mở rộng của nó và những vấn đề liên quan.

1.1 Mô hình SIR rời rạc

Hãy xem xét một căn bệnh đang lan rộng khắp nước Mỹ, chẳng hạn như bệnh cúm mới. Các Trung tâm kiểm soát và phòng ngừa dịch bệnh quan tâm đến việc tìm hiểu và thử nghiệm với một mô hình cho căn bệnh mới này trước khi nó trở thành một dịch bệnh thực sự. Ta chia dân số thành ba loại: miễn cảm, nhiễm bệnh và đã loại bỏ. Chúng ta làm các giả định sau đây cho mô hình:

- Không ai vào hoặc ra khỏi cộng đồng, và không có liên hệ bên ngoài cộng đồng.
- Mỗi người đều dễ mắc bệnh S (có thể mắc bệnh cúm mới này); bị nhiễm bệnh I (hiện có cảm cúm và có thể lây cúm); hoặc loại bỏ R (đã hết nhiễm và sẽ không tái phát, trong đó bao gồm trường hợp tử vong).
- Ban đầu, mỗi người trong cộng đồng chỉ có thể là S hoặc I. Một khi ai đó đã bị cúm, người đó không thể mắc lại.
- Giả sử độ dài trung bình của bệnh là $5/3$ tuần, I là người bị coi là bị nhiễm bệnh và có thể truyền bệnh.
- Khoảng thời gian của chúng ta xem xét cho mô hình là mỗi tuần.

Mô hình mà chúng ta sẽ xem xét liên quan đến những người dễ mắc bệnh, những người bị nhiễm bệnh và những người đã hồi phục, thường được gọi là mô hình SIR. Ta định nghĩa cho các biến như sau:

$S(t)$ = số người trong dân số dễ mắc bệnh sau thời điểm t

$I(t)$ = số người trong dân số đã mắc bệnh và có khả năng lây bệnh tại thời điểm t

$R(t)$ = số người trong dân số đã không còn mắc bệnh (bao gồm cả trường hợp tử vong)

Hãy bắt đầu mô hình của chúng ta bằng việc xử lý $R(t)$. Giả định của chúng ta có thời gian người trong giai đoạn cúm là $5/3$ (1 và $2/3$) tuần. Như vậy, $3/5$ số người nhiễm bệnh sẽ được loại bỏ mỗi tuần:

$$R(n+1) = R(n) + 0.6I(n)$$

Giá trị 0,6 được gọi là tỷ lệ loại bỏ mỗi tuần. Tỷ lệ loại bỏ đại diện cho tỷ lệ của những người bị nhiễm được loại bỏ khỏi nhiễm bệnh mỗi tuần. Nếu dữ liệu thực có sẵn ta sẽ lấy nó, nếu không sau đó chúng ta có thể thực hiện "phân tích dữ liệu" để lấy tỷ lệ loại bỏ này.

$I(t)$ sẽ có khả năng tăng và giảm số lượng của chính nó theo thời gian. $I(t)$ bị giảm bởi số lượng bị loại bỏ mỗi tuần: $-0.6 * I(t)$. Và $I(t)$ được tăng lên bởi số người nhạy cảm tiếp xúc với một người bị nhiễm bệnh và mắc bệnh, $a * S * I(t)$. Chúng ta xác định tỷ lệ a là tỷ lệ mà bệnh lây lan, hoặc hệ số truyền. Chúng ta nhận ra đây là một xác suất hệ số. Ban đầu, chúng ta sẽ cho rằng tỷ lệ này là một giá trị không đổi có thể được tìm thấy từ điều kiện ban đầu. Hãy giả sử minh họa sau: Giả sử chúng ta có 1000 sinh viên cư trú các ký túc xá. Y tá đã tìm thấy 5 học sinh báo cáo cho bệnh xá ban đầu: $I(0)=5$ và $S(0)=995$. Sau một tuần, tổng số người bị cúm là 9. Chúng ta tính a như sau:

$$I(0) = 5, I(1) = I(0) - 0.6 * I(0) + aI(0) * S(0)$$

$$I(1) = 9 = 5 - 3 + a * 5 * 995$$

$$7 = a(4957)$$

$$a = 0,001407$$

Hãy xem xét $S(n)$. Con số này chỉ giảm bởi số lượng bị nhiễm bệnh. Chúng ta có thể sử dụng tỷ lệ a như trước đây để có được mô hình:

$$S(n+1) = S(n) - aS(n)I(n)$$

Khi đó, chúng ta thu được mô hình kết hợp:

$$R(n+1) = R(n) + 0.6I(n)$$

$$I(n+1) = I(n) - 0.6I(n) + 0.001407I(n)S(n)$$

$$S(n+1) = S(n) - 0.001407S(n)I(n)$$

$$I(0) = 5, S(0) = 995, R(0) = 0$$

Phương trình mô hình SIR trên, có thể được giải quyết lặp lại và xem xét bằng đồ thị. Ta lặp lại giải pháp và thu được biểu đồ để quan sát hành vi để có được một số thông tin cần biết.

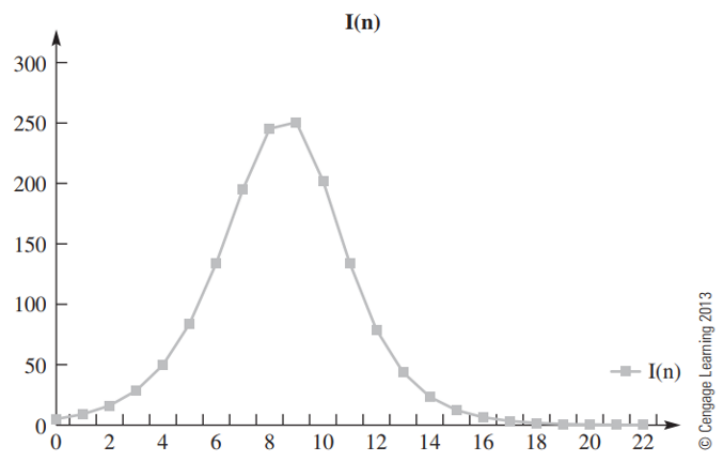
Week	$R(n)$	$I(n)$	$S(n)$
0	0	5	995
1	3	8.999825	988.0002
2	8.399895	16.11073	975.4894
3	18.06633	28.55649	953.3772
4	35.20023	49.72832	915.0714
5	65.03722	83.91682	851.046
6	115.3873	134.0505	750.5621
7	195.8176	195.1831	608.9993
8	312.9275	245.3182	441.7543
9	460.1184	250.6044	289.2772
10	610.481	202.241	187.2779
11	731.8256	134.1869	133.9874
12	812.3378	78.97173	108.6905
13	859.7208	43.66564	96.61352
14	885.9202	23.40195	90.67782
15	899.9614	12.34649	87.69211
16	907.3693	6.46194	86.16877
17	911.2465	3.368218	85.38533
18	913.2674	1.751936	84.98068
19	914.3185	0.910249	84.7712
20	914.8647	0.472668	84.66264
21	915.1483	0.245372	84.60633
22	915.2955	0.127358	84.57712

Hình 1: Biểu đồ phân bố số liệu SIR rời rạc

Giải thích: Chúng ta phân tích cả bảng số và Hình 1.31 để kiểm tra những gì đang xảy ra trong dịch. Đỉnh điểm của dịch cúm xảy ra vào khoảng tuần 9, trong đó đồ thị của những người bị nhiễm đạt giá trị tối đa. Số lượng tối đa là lớn hơn 250 một chút; nhìn từ bảng ta thấy là 250.6044. Sau khoảng 17 tuần, ít nhất 85 người vẫn chưa bị cúm.

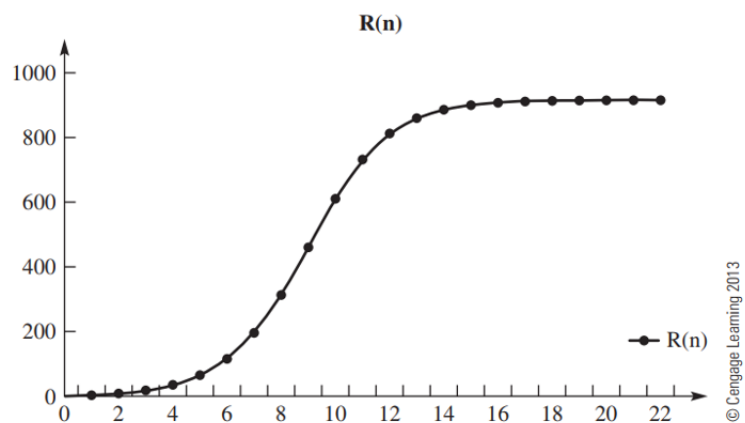
■ **Figure 1.31**

Plot of $I(n)$ versus time



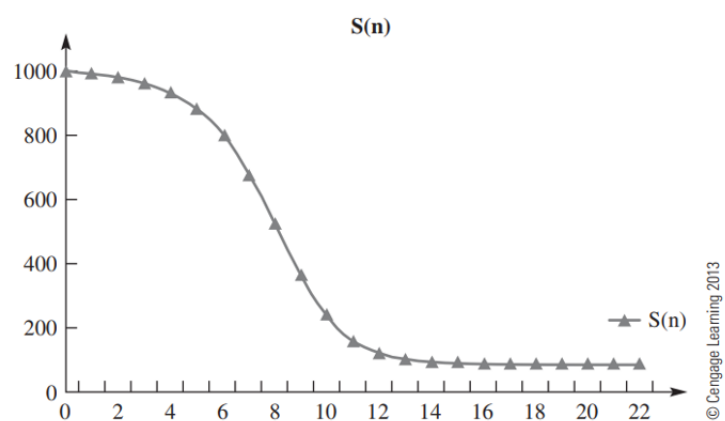
■ **Figure 1.32**

Plot of $R(n)$ over time



■ **Figure 1.33**

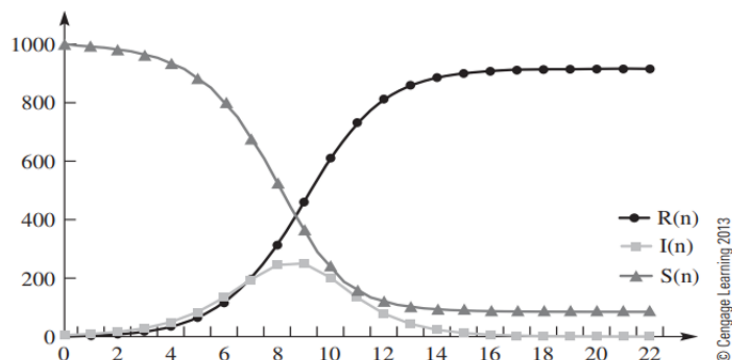
Plot of $S(n)$ over time



Hình 2

■ Figure 1.34

Plot of $I(n)$, $S(n)$, and $R(n)$ overlaid together



1.2 Mô hình SIR liên tục

Mô hình SIR (Susceptible - Infectious - Recovered) là một trong các mô hình cách ly cơ bản được sử dụng nhiều nhất trong quá khứ và hiện tại để mô tả dịch bệnh. Mô hình được lần đầu tiên phát biểu vào thế kỷ 20. Mô hình thể hiện ba trạng thái (Có nguy cơ mắc bệnh - Mắc bệnh - Hồi phục) cho một nhóm người được cách ly với giả thiết rằng sẽ miễn dịch với bệnh nếu đã phục hồi. Mô hình SIR là một hệ động lực gồm ba phương trình vi phân sau:

$$\begin{aligned}\frac{dS}{dt} &= -\frac{\beta}{N}IS \\ \frac{dI}{dt} &= \frac{\beta}{N}IS - \gamma I \\ \frac{dR}{dt} &= \gamma I\end{aligned}$$

trong đó tại mỗi thời điểm $t \geq t_0 \geq 0$ với t_0 là thời điểm đầu ghi nhận,

$S(t)$: Số người có nguy cơ mắc bệnh

$I(t)$: Số người nhiễm bệnh

$R(t)$: Số người phục hồi sau bệnh

$\beta(t)$: Tỷ lệ tiếp xúc của mỗi người trong nhóm $S(t)$ với người trong nhóm $I(t)$

$\gamma(t)$: Tỷ lệ hồi phục khi mắc bệnh

$N(t)$: Tổng số người trong cộng đồng bị cách ly được tính bằng: $N(t) := S(t) + I(t) + R(t)$

Hệ phương trình vi phân trên có thể được hiểu như sau:

- Phương trình (1) thể hiện sự suy giảm số người có nguy cơ mắc bệnh tại thời điểm $t \geq t_0$. Sự suy giảm được tính theo xác suất lây bệnh khi có tiếp xúc giữa nhóm $S(t)$ và nhóm $I(t)$;
- Phương trình (2) thể hiện độ biến thiên số người mắc bệnh tại thời điểm $t \geq t_0$. Sự biến thiên này được tính bằng cách lấy số người ở nhóm $S(t)$ đã bị lây nhiễm sau khi tiếp xúc với người bệnh nhóm $I(t)$ và trừ đi số người ở nhóm $I(t)$ đã phục hồi với tỷ lệ $\gamma I(t)$;
- Phương trình (3) thể hiện số người đã hồi phục từ nhóm $I(t)$ theo tỷ lệ hồi phục là γ .

Lấy minh họa với các số liệu tương tự như mô hình SIR rời rạc, ta có hệ các phương trình kết hợp sau:

$$\begin{aligned}\frac{dR}{dt} &= 0.6I(t) \\ \frac{dI}{dt} &= -0.6I(t) + 0.001407I(t)S(t) \\ \frac{dS}{dt} &= 0.001407I(t)S(t)\end{aligned}$$

$$I(0)=5, S(0)=995, R(0)=0$$

Chúng ta sẽ giải mô hình SIR liên tục này bằng phương pháp xấp xỉ Euler.

1.2.1 Phương pháp xấp xỉ Euler

Phương pháp Euler là một phương pháp bậc một thường được sử dụng trong việc giải các phương trình vi phân thường. Phương pháp được đặt tên theo Leonhard Euler, người đã giới thiệu phương pháp trong quyển sách Institutionum Calculi Integralis cùng tên xuất bản trong khoảng thời gian 1768 đến 1770. Giả sử ta có phương trình vi phân bậc nhất:

$$y' = f(t, y(t))$$

Khi đó ý tưởng của phương pháp Euler là xấp xỉ nghiệm y bằng dãy y_n sao cho

$$y_{n+1} := y_n + f(t_n, y_n) \Delta t$$

với Δt là xấp xỉ đủ nhỏ và $f(t_n, y_n)$ là độ dốc của đường cong y tính tại thời điểm t

Ở dạng tổng quát, một hệ phương trình vi phân được viết dưới dạng

$$y'_1 = f_1(t, y_1, \dots, y_N)$$

\vdots

$$y'_N = f_N(t, y_1, \dots, y_N)$$

Khi đó, áp dụng xấp xỉ Euler vào mô hình SIR liên tục ta được:

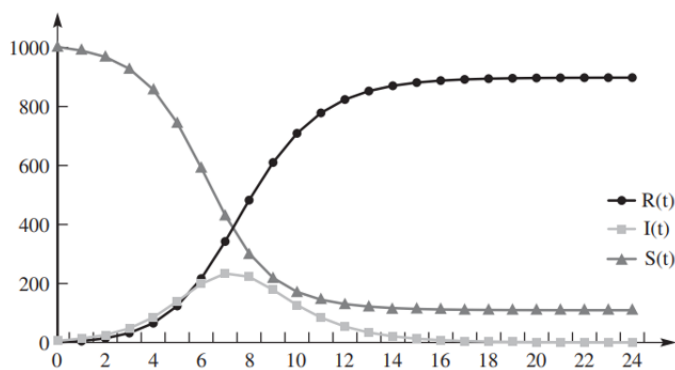
$$S_n = S_{n-1} - b S_{n-1} I_{n-1} \Delta t$$

$$I_n = I_{n-1} + (b S_{n-1} I_{n-1} - k I_{n-1}) \Delta t$$

$$R_n = R_{n-1} + k I_{n-1} \Delta t$$

Sau khi giải bằng phương pháp Euler với $\Delta t = 0,5$ ta được kết quả sau:

Week	$R(t)$	$I(t)$	$S(t)$
0	0	5	995
1	4.29623	13.0107	982.693
2	13.6301	25.0714	961.299
3	31.503	47.0769	921.42
4	64.6671	84.2088	851.124
5	122.74	138.163	739.097
6	214.755	197.109	588.136
7	339.747	231.908	428.346
8	478.985	221.114	299.902
9	605.748	176.888	217.364
10	704.076	125.809	170.115
11	772.755	83.435	143.81
12	817.839	53.1578	129.003
13	846.397	33.0964	120.506
14	864.118	20.332	115.55
15	874.983	12.3924	112.624
16	881.598	7.51803	110.884
17	885.608	4.54821	109.844
18	888.033	2.74696	109.22
19	889.497	1.6574	108.845
20	890.381	0.99941	108.62
21	890.913	0.60242	108.484
22	891.234	0.36305	108.403
23	891.428	0.21876	108.354
24	891.544	0.13181	108.324



■ Figure 12.31

SIR model

1.3 Những vấn đề liên quan đến mô hình SIR

*Hệ số R_0 :

Một trong những đại lượng quan trọng nhất đối với một mô hình dịch bệnh là hệ số lây nhiễm cơ bản, hay thường gọi là “hệ số R_0 ”. Nếu $R_0 < 1$, dịch sẽ tắt trước khi kịp bùng phát, còn nếu $R_0 > 1$, dịch sẽ bùng phát. Thí dụ, trong mô hình SIR đơn giản ở trên, $R_0 = N/\gamma$, trong đó N là số người khỏe mạnh trung bình mà một người mắc bệnh có thể lây cho trong khoảng thời gian mắc bệnh $1/\gamma$. Một cách trực giác, điều này khá hợp lý: nếu trung bình một người mắc bệnh lây cho nhiều hơn một người thì số người mắc bệnh phải tăng (theo cấp số nhân), còn nếu trung bình một người mắc bệnh lây cho ít hơn một người khác thì số người mắc bệnh phải giảm dần.

Ước lượng này có vai trò rất quan trọng và được thể hiện trên hệ số $R_0 := \frac{\beta}{\gamma}$

1.4 Mở rộng của mô hình SIR - mô hình SIRD

Chúng ta xem xét một cộng đồng N người không đổi và không có sự dịch chuyển giữa bên trong và bên ngoài. Khi đó, mô hình SIRD đặt mỗi người trong cộng đồng vào một trong năm trạng thái sau:

$$S_t + I_t + R_t + D_t + C_t = N_t$$

với:

S_t = Susceptible

I_t = Infectious

R_t = Resolving

D_t = Dead

C_t = ReCovered

Lưu ý: R trong mô hình SIRD là Resolving khác với R trong mô hình SIR là Resolved. Ý nghĩa thực tế thể hiện số người được quan tâm chữa trị.

Sau khi thời kỳ lây nhiễm của người đó kết thúc, người đó ở trong trạng thái đã giải quyết vấn đề, R . Một hằng phân số θ của những người thoát khỏi trạng thái này mỗi kỳ (tức là sau thời gian chữa trị và không còn khả năng lây nhiễm) sẽ rơi vào một trong hai trường hợp (Tử vong – Bình phục):

Death: fraction δ

Recovery: fraction $1-\delta$

Trong các mô hình phiên bản trước đây của dự án này, chúng ta thấy rằng điều quan trọng là phải có một mô hình phân biệt giữa thời kỳ truyền nhiễm và thời gian phục hồi. Sự khác biệt này là chìa khóa để khớp dữ liệu với các giá trị tham số hợp lý về mặt sinh học khi chúng ta đặt các hạn chế trên đường thời gian của β

Ta có hệ mô hình SIRD được thể hiện đầy đủ dưới đây:

$$\Delta S_{t+1} = -\beta_t S_t I_t / N$$

$$\Delta I_{t+1} = \beta_t S_t I_t / N - \gamma I_t$$

$$\Delta R_{t+1} = \gamma I_t - \theta R_t$$

$$\Delta D_{t+1} = \delta \theta R_t$$

$$\Delta C_{t+1} = (1 - \delta) \theta R_t$$

2 Bài toán 2

2.1 Mô tả

Giả sử rằng có một loại cúm đang lây lan trong một cộng đồng dân cư. Giả sử rằng:

- Cộng đồng này đang bị cách ly, không ai được ra và cũng không ai được vào;
- Loại cúm này có thời gian từ khi phát bệnh cho đến khi hồi phục là 2 tuần không đổi theo thời gian;
- Một người khi mắc bệnh và hồi phục thì không còn mắc bệnh này lại lần thứ hai;
- Sau một thời gian điều tra, tỷ lệ mắc bệnh khi có tiếp xúc với người bệnh ở mức 0.2% sau một tuần tiếp xúc và giả sử tỷ lệ này cũng không đổi theo thời gian.

$$\frac{dS}{dt} = -0.002IS$$

$$\frac{dI}{dt} = 0.002IS - 0.5I$$

$$\frac{dR}{dt} = 0.5I$$

Giả sử tại thời điểm ban đầu, số người có khả năng nhiễm bệnh là 800 người, số người mắc bệnh là 5 người và số ca phục hồi khi ấy chưa có. Sử dụng phương pháp Euler, ta sẽ giải hệ SIR để tìm ra số người có khả năng bị lây nhiễm, số người mắc bệnh và số ca phục hồi sau hai tháng (8 tuần) tính từ thời điểm đầu ghi nhận số liệu.

2.2 Viết chương trình sử dụng thuật toán Euler tìm nghiệm của hệ SIR.

2.2.1 Ví dụ 1 mô hình SIR

Điều kiện đầu vào: $N=800$, $S(0)=793$, $I(0)=7$, $R(0)=0$

Tham số đầu vào: $t=12$ tuần, hệ số tiếp xúc $\beta=0.002$, hệ số bình phục $\gamma=0.5$

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 nweeks = 13.
5 dt = 1.
6 npts = int(nweeks/dt)
7 beta = .002
8 gamma = .5
9
10
11 S = np.zeros(npts)
12 I = np.zeros(npts)
13 R = np.zeros(npts)
14 t = np.arange(npts)*dt
15
16
17 I[0] = 7.
18 S[0] = 793.
```

```
19 R[0] = 0.  
20  
21 print(S[0], "\t", I[0], "\t", R[0])  
22 for i in range(npts-1):  
23     S[i+1] = S[i] - beta*S[i]*I[i]*dt  
24     I[i+1] = I[i] + (beta*(S[i]*I[i]) - gamma*I[i])*dt  
25     R[i+1] = R[i] + gamma*I[i]*dt  
26     print(S[i+1], "\t", I[i+1], "\t", R[i+1])  
27  
28  
29 fig = plt.figure(1); fig.clf()  
30  
31 plt.plot(t, S, 'b', lw=3, label='Susceptible')  
32 plt.plot(t, I, 'r', lw=3, label='Infected')  
33 plt.plot(t, R, 'g', lw=3, label='Removed')  
34 fig.legend(); plt.xlabel('Week'); plt.ylabel('People')  
35 print("Smax =", S.max())  
36 print("Imax =", I.max())  
37 print("Rmax =", R.max())
```

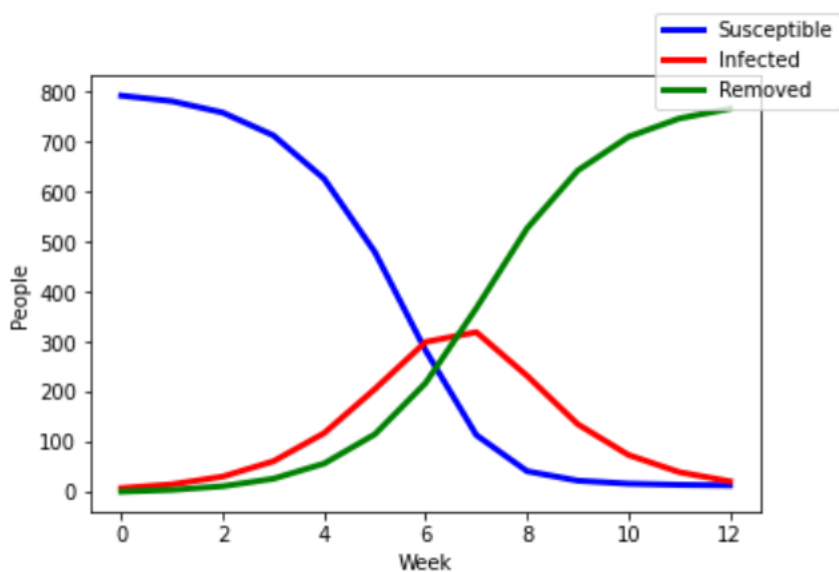
Listing 1: Python example

2.2.2 Kết quả ví dụ 1

Kết quả thu được cho như bảng sau:

Week 0	793.0	7.0	0.0	
Week 1	781.898	14.602	3.5	
Week 2	759.063450808		30.135549192000006	10.801
Week 3	713.3138628846525		60.817362519347526	25.868774596
Week 4	626.5501273063884		117.1724168379379	56.277455855673765
Week 5	479.721341933174		205.41499379218334	114.86366427464272
Week 6	282.6374289828124		299.7914098464533	217.5711611707344
Week 7	113.17288236254413		319.360251543495	367.466866093961
Week 8	40.887042004135196		231.96596613015643	527.1469918657085
Week 9	21.918237602748178		134.95178746646525	643.1299749307867
Week 10	16.00242691753706		73.39170441844374	710.6058686640193
Week 11	13.653536144917807		39.044742981841125	747.3017208732413
Week 12	12.58733852577462		20.58856911006375	766.8240923641619
Smax =	793.0			
I _{max} =		319.360251543495		
Rmax =			766.8240923641619	

Hình 3



Hình 4: Số ca mắc bệnh tăng trong khoảng 6 tuần đầu tiên và giảm dần ở các tuần tiếp theo

2.2.3 Nhận xét ví dụ 1

Quan sát đồ thị, ta thấy:

- Số ca mắc bệnh được biểu diễn theo sơ đồ hình sin thoải, tăng dần đến đỉnh điểm (tuần thứ 6) và giảm dần ở các tuần tiếp theo
- Số ca có nguy cơ mắc bệnh đến tuần thứ 8 thì gần như đồ thị chạy theo phương ngang, chứng tỏ dịch bệnh được kiểm soát tốt về vấn đề cách ly

- Số ca hồi phục luôn theo hướng đồ thị tăng dần thể hiện chiều hướng tốt và tích cực trong công tác chữa trị
- Theo kết quả số liệu, ta tìm ra được số ca mắc tối đa tại cùng một thời điểm lên tới $I_{\max}=319$ ca, dựa vào con số này các nhà chức trách có thể ước lượng khả năng quá tải để có giải pháp phù hợp.

2.2.4 Ví dụ 2 mô hình SIRD

Điều kiện đầu vào $N=800$, $S(0)=793$, $I(0)=7$, $R(0)=0$, $D(0)=0$, $C(0)=0$

Tham số đầu vào:

$t=12$ tuần

hệ số tiếp xúc $\beta = 0.002$

hệ số bình phục $\gamma = 0.5$

hệ số ra khỏi thời gian chữa trị $\theta = 0.6$

hệ số tử vong $\delta = 0.3$

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 nweeks = 13.
5 dt = 1.
6 npts = int(nweeks/dt)
7 beta = .002
8 gamma = .5
9 theta = .6
10 delta = .3
11
12
13 S = np.zeros(npts)
14 I = np.zeros(npts)
15 R = np.zeros(npts)
16 D = np.zeros(npts)
17 C = np.zeros(npts)
18 t = np.arange(npts)*dt
19
20
21 I[0] = 7.
22 S[0] = 793.
23 R[0] = 0.
24
25 print("Week 0", " ", S[0], "\t", I[0], "\t", R[0], "\t", D[0], "\t", C[0])
26 for i in range(npts-1):
27     S[i+1] = S[i] - beta*S[i]*I[i]*dt
28     I[i+1] = I[i] + (beta*(S[i]*I[i]) - gamma*I[i])*dt
29     R[i+1] = R[i] + gamma*I[i]*dt - theta*R[i]
30     D[i+1] = D[i] + delta*theta*R[i]
31     C[i+1] = C[i] + (1-delta)*theta*R[i]
32     print("Week", i+1, " ", S[i+1], "\t", I[i+1], "\t", R[i+1], "\t", D[i+1], "\t", C[i+1])
33
34
35 fig = plt.figure(1); fig.clf()
36
37 plt.plot(t,S,'b',lw=3, label='Susceptible')
38 plt.plot(t,I,'r',lw=3, label='Infected')
39 plt.plot(t,R,'g',lw=3, label='Resolving')
40 plt.plot(t,D,'black',lw=3, label='Died')
41 plt.plot(t,C,'yellow',lw=3, label='Recovered')
42 fig.legend(); plt.xlabel('Week'); plt.ylabel('People')
43 print("Smax =", S.max())
```

```
44 print("Imax =", I.max())
45 print("Rmax =", R.max())
46 print("Dmax =", D.max())
47 print("Cmax =", C.max())
```

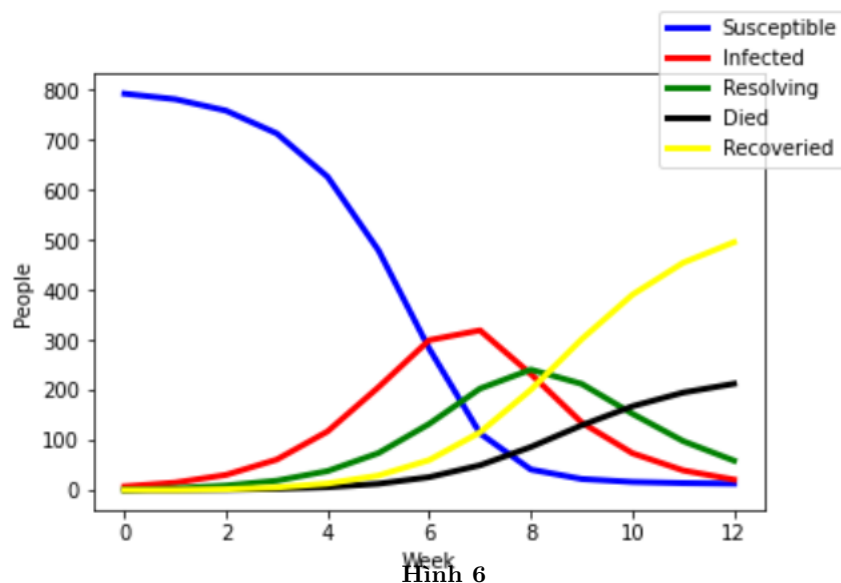
Listing 2: Python example

2.2.5 Kết quả ví dụ 2

Kết quả thu được cho như bảng sau:

Week 0	793.0	7.0	0.0	0.0	0.0					
Week 1	781.898	14.602		3.5	0.0	0.0				
Week 2	759.063450808		30.135549192000006			8.701	0.63	1.47		
Week 3	713.3138628846525		60.817362519347526			18.548174596000003			2.19618	5.12442
Week 4	626.5501273063884		117.1724168379379			37.82795109807377			5.5348514272800005	12.91465333032
Week 5	479.721341933174		205.41499379218334			73.71738885819846			12.34388262493328	28.802392791510982
Week 6	282.6374289828124		299.7914098464533			132.19445243937105			25.613012619409	59.76369611195433
Week 7	113.17288236254413		319.360251543495			202.77348589897503			49.408014058495795	115.28536613649017
Week 8	40.887042004135196		231.96596613015643			240.7895201313375			85.90724152031129	200.45023021405967
Week 9	21.918237602748178		134.95178746646525			212.2987911176132			129.24935514395204	301.5818286692214
Week 10	16.00242691753706		73.39170441844374			152.39541018027788			167.46313754512244	390.74732093861894
Week 11	13.653536144917807		39.044742981841125			97.65401628133303			194.89431137757245	454.75339321433563
Week 12	12.58733852577462		20.58856911006375			58.58397800345378			212.4720343082124	495.7680800524955
Smax =	793.0									
Imax =	319.360251543495									
Rmax =	240.7895201313375									
Dmax =	212.4720343082124									
Cmax =	495.7680800524955									

Hình 5



Hình 6

2.2.6 Nhận xét ví dụ 2

Quan sát đồ thị, ta thấy:

- Số ca mắc bệnh được biểu diễn theo sơ đồ hình sin thoải, tăng dần đến đỉnh điểm (tuần thứ 6) và giảm dần ở các tuần tiếp theo



- Số ca có nguy cơ mắc bệnh đến tuần thứ 8 thì gần như đồ thị chạy theo phương ngang, chứng tỏ dịch bệnh được kiểm soát tốt về vấn đề cách ly
- Số ca được quan tâm chữa trị tăng dần, sau khi số ca mắc bệnh đạt đỉnh ở tuần 6 thì số ca được quan tâm chữa trị theo ngay sau đạt đỉnh ở tuần 8. Sau đó, vì áp lực bệnh nhân mắc bệnh giảm dần nên số ca được chữa trị cũng giảm theo. Độ trễ so với số ca mắc khoảng 1 tuần.
- Tổng số ca tử vong và số ca bình phục hoàn toàn có xu hướng tăng dần và bằng số ca bình phục ở trường hợp mô hình SIR
- Theo kết quả số liệu, ta qua tâm tới 3 thông số sau:
 - + Số ca mắc tối đa $I_{\max}=319$
 - + Số ca được chữa trị tối đa $R_{\max}=241$
 - + Số ca tử vong đỉnh điểm $D_{\max}=212$

3 Bài toán 3

Viết chương trình theo ngôn ngữ tự chọn để lấy mẫu sử dụng thuật toán Metropolis–Hastings với tham số đầu vào là phân bố xác suất tiên nghiệm $\pi(\beta, \gamma)$ cho trước. Giá trị trả về là một mẫu gồm các cặp β và γ có phân bố xác suất $\pi(\beta, \gamma)$. Vẽ biểu đồ thể hiện quá trình chọn mẫu. Trình bày chi tiết kết quả trong báo cáo.

3.1 Mô tả

Trong bối cảnh hiện nay, phương án cách ly từng nhóm người đã từng tiếp xúc trực tiếp hoặc gián tiếp được các quốc gia trên thế giới xem như một cách hữu hiệu nhất để giảm thiểu số ca mắc bệnh. Như vậy các mô hình cách ly dạng SIR có thể được sử dụng trong trường hợp này. Tuy nhiên ta sẽ xét các hệ số β và γ có thể biến đổi theo thời gian do có sự điều chỉnh trong các lệnh cách ly theo thời gian. Từ đó, cơ hội tiếp xúc gần với người nhiễm virus tăng hoặc giảm dẫn đến xác suất lây nhiễm β thay đổi. Khi tình trạng các bệnh viện trở nên quá tải, sự tập trung của các y bác sĩ cho từng bệnh nhân có thay đổi và dẫn đến tỷ lệ phục hồi cũng khác nhau theo thời gian.

Việc ước lượng các hệ số β và γ phụ thuộc vào dữ liệu về COVID-19 đã được công bố. Cụ thể là số ca mắc bệnh và phục hồi tích lũy theo thời gian. Ở đây, ta sẽ sử dụng phương pháp suy luận Bayes.

Gọi:

- X = Biến ngẫu nhiên quan sát số ca mắc bệnh và số ca phục hồi tại từng thời điểm $t \geq t_0$
- $\pi(\beta, \gamma | X)$ = phân bố xác suất xuất hiện hậu nghiệm của β và γ khi có dữ liệu quan sát
- $\pi(X | \beta, \gamma)$ = phân bố xác suất của số ca mắc bệnh và số ca phục hồi khi β và γ cho trước
- $\pi(\beta, \gamma)$ = phân bố xác suất tiên nghiệm khi chưa có dữ liệu ghi nhận về số ca mắc và số ca phục hồi β và γ khi có dữ liệu quan sát

Định lý Bayes được phát biểu như sau

$$\pi(\beta, \gamma | X) \propto \pi(X | \beta, \gamma) \pi(\beta, \gamma)$$

Nghĩa là phân bố xác suất hậu nghiệm của β và γ có thể được tính bằng cách lấy phân bố xác suất của số ca mắc bệnh và số ca phục hồi khi β và γ cho trước nhân với phân bố xác suất tiên nghiệm của β và γ

3.2 Giải

1. Khởi tạo β_0 và γ_0 từ phân bố xác suất tiên nghiệm $\pi(\beta, \gamma)$.

```
1 beta0 = gamma.pdf(0.3, 1)
2 gamma0 = gamma.pdf(0.2, 1)
```

2. Khởi tạo β^* và γ^* ngẫu nhiên từ phân phối xác suất chuẩn $N(x, \sigma^2)$ để tạo β^* và γ^* , độ lệch chuẩn là 0.034.

```
1 betaS = np.random.normal(betaVar, 0.034)
2 gammaS = np.random.normal(gammaVar, 0.034)
```

3. Vì phân phối chuẩn đối xứng nên gán r bằng công thức

$$r := \min(1, \frac{\pi(\beta^*, \gamma^*)}{\pi(\beta, \gamma)})$$

4. Khởi tạo giá trị q ngẫu nhiên từ phân phối đều liên tục $U(0, 1)$

```
1 q = np.random.uniform(0,1)
```

5. Nếu $q < r$, tạo $\beta_{i+1} := \beta^*$ và $\gamma_{i+1} := \gamma^*$ với i là chỉ số phần tử trong mẫu

```
1 if (q < r):  
2     listBeta.append(betaS)  
3     listGamma.append(gammaS)
```

6. Ngược lại, tạo $\beta_{i+1} := \beta_i$ và $\gamma_{i+1} := \gamma_i$

```
1 else:  
2     listBeta.append(listBeta[-1])  
3     listGamma.append(listGamma[-1])
```

7. Lặp lại từ Bước 2 với $\beta := \beta_i$ và $\gamma := \gamma_i$ cho đến khi đủ kích cỡ mẫu.

3.3 Tổng hợp Code

```
1 import numpy as np  
2 from scipy.stats import gamma  
3 from scipy.stats import uniform  
4 import decimal  
5 import matplotlib.pyplot as plt  
6 #With lambda  
7 sizeOfSample = int(input("Type the sample number: "))  
8 listBeta = []  
9 listGamma = []  
10 beta0 = gamma.pdf(0.3, 1)  
11 gamma0 = gamma.pdf(0.2, 1)  
12 listBeta.append(beta0)  
13 listGamma.append(gamma0)  
14 print("Base beta is", listBeta[0])  
15 print("Base gamma is", listGamma[0])  
16  
17 while ((len(listBeta) < sizeOfSample) and (len(listGamma) < sizeOfSample)):  
18     betaVar = listBeta[-1]  
19     gammaVar = listGamma[-1]  
20     # Normal distribution with sigma = (x* 0.34)^2  
21     betaS = np.random.normal(betaVar, 0.034) # Use decimal when overflow  
22     gammaS = np.random.normal(gammaVar, 0.034)  
23     # print("Beta star is", betaS)  
24     # print("Gamma star is", gammaS)  
25     betaS=gamma.pdf(betaS,1)  
26     gammaS=gamma.pdf(gammaS,1)  
27     r = min(1, (betaS * gammaS) / (betaVar * gammaVar))  
28     q = np.random.uniform(0,1) # Uniform distribution (continuous) U(0,1)  
29     # print("q is", q)  
30  
31     if (q < r):  
32         listBeta.append(betaS)  
33         listGamma.append(gammaS)  
34     else:
```



```

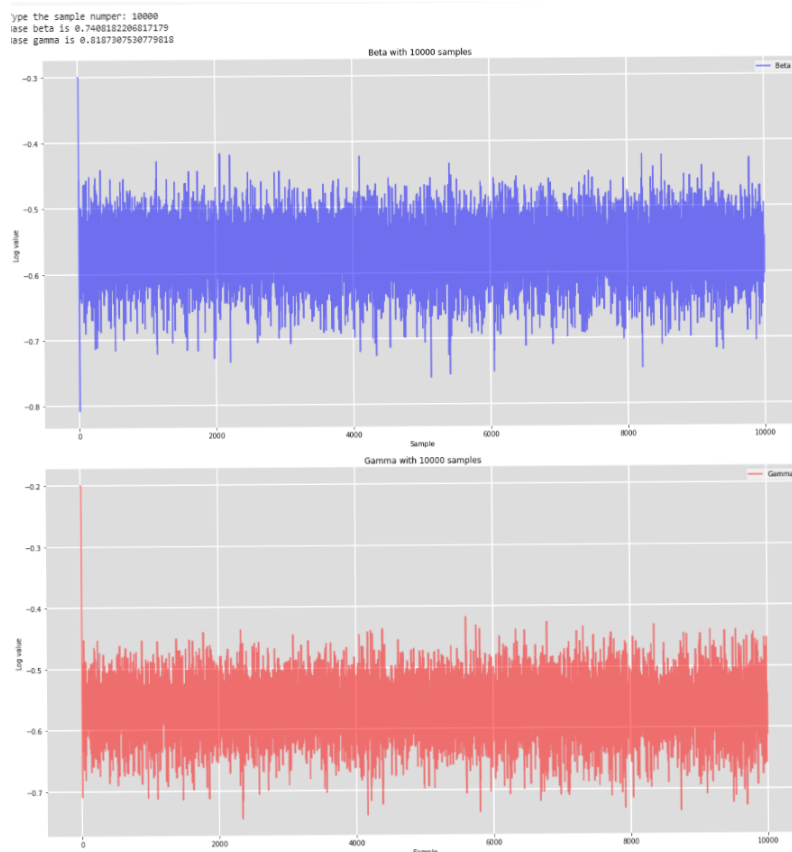
35     listBeta.append(listBeta[-1])
36     listGamma.append(listGamma[-1])
37
38 n = np.linspace(0, sizeOfSample, sizeOfSample)
39 betaRange = [np.log(x) for x in listBeta] # To plot
40 gammaRange = [np.log(x) for x in listGamma]
41 fig = plt.figure(facecolor='w', figsize=(20,10))
42 ax = fig.add_subplot(111, facecolor='#dddddd', axisbelow=True)
43 ax.plot(n, betaRange, 'b', alpha=0.5, lw=2, label='Beta')
44 ax.plot(n, gammaRange, 'r', alpha=0.5, lw=2, label='Gamma')
45 ax.title.set_text('Beta and Gamma with ' + str(sizeOfSample) + ' samples')
46 ax.set_xlabel('Sample')
47 ax.set_ylabel('Value')
48 ax.grid(b=True, which='major', c='w', lw=2, ls='--')
49 legend = ax.legend()
50 legend.get_frame().set_alpha(0.5)
51 for spine in ('top', 'right', 'bottom', 'left'):
52     ax.spines[spine].set_visible(False)
53 plt.show()

```

Listing 3: Python example

3.4 Kết quả

Hình 7



4 Bài toán 4

4.1 Mô tả

Nhóm quyết định chọn Nhật Bản để ước lượng giá trị trung bình của hệ số R_0

Cơ sở lý thuyết

Giá trị trung bình của hệ số này đúng bằng:

$$E(R_0) = \int \pi(\beta, \gamma|X) R_0(\beta, \gamma) d(\beta, \gamma) \quad (20)$$

Trong đó X là dữ liệu về số ca mắc bệnh và phục hồi quan sát được. Giá trị trung bình này có thể ước lượng được vì phân bố xác suất $\pi(\beta, \gamma|X)$ có thể tính được nhờ vào công thức Bayes

Tuy nhiên, tích phân (20) không thể được tính toán một cách trực tiếp. Thay vào đó chúng ta sẽ sử dụng công thức xấp xỉ

$$E(R_0) = \sum_{i=1}^m \pi(\beta, \gamma|X) \frac{\beta_i}{\gamma_i} \quad (21)$$

trong đó (β_i, γ_i) được lấy ra dựa trên phân bố xác suất tiên nghiệm $\pi(\beta, \gamma|X)$ và m là kích cỡ mẫu. Việc chọn mẫu cũng như cặp hệ số (β_i, γ_i) ta đã thực hiện ở bài toán 3.

Mặt khác, ta cũng thế vào công thức trên tiên nghiệm $\pi(\beta, \gamma|X)$ theo phân phối Gamma được tính bằng:

$$\pi(\beta, \gamma|X) = \prod_{i=1}^n \frac{\gamma^\beta}{\Gamma^\beta} X(t_i)^{\beta-1} \exp(-\gamma X(t_i)) \quad (15)$$

4.2 Hiện thực code

```
1 import numpy as np
2 import pandas as pd
3 from scipy.special import gamma
4 from scipy.stats import gamma as sGamma
5 import matplotlib.pyplot as plt
6 import decimal
7 import requests
8 import io
9 import math
10 START_DATE = '1/22/20'
11 url = 'https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/
      csse_covid_19_data/csse_covid_19_time_series/
      time_series_covid19_confirmed_global.csv'
12 contentFile = requests.get(url).content
13 df = pd.read_csv(io.StringIO(contentFile.decode('utf-8')))
14 dailyTime = df.columns[4:]
15
16 def loadData(type, country):
17     url = 'https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/
      csse_covid_19_data/csse_covid_19_time_series/time_series_covid19_'
18     contentFile = requests.get(url + str(type) + '_global.csv').content
19     df = pd.read_csv(io.StringIO(contentFile.decode('utf-8')))
20     country_df = df[df['Country/Region'] == country]
21     return country_df.loc[START_DATE:]
22
23 def dailyData(data, time):
```

```
24     sumValue = 0
25     for x in data[time]:
26         sumValue += x
27     return sumValue
28
29 def processData(country, b, g):
30     i = 0
31     piX = 1
32     dataI = loadData('confirmed', country)
33     dataR = loadData('recovered', country)
34     while (i < len(dailyTime)):
35         dailyI = dailyData(dataI, dailyTime[i])
36         dailyR = dailyData(dataR, dailyTime[i])
37         X = dailyI + dailyR
38         piX *= (g**b * (X)**(b - 1) * math.exp((-1) * g * X)) / gamma(b)
39         i += 1
40     return piX
41
42 def main():
43     country = 'Japan'
44     sizeOfSample = int(input("Type the sample number: "))
45     listBeta = []
46     listGamma = []
47     beta0 = sGamma.pdf(0.5, 1)
48     gamma0 = sGamma.pdf(0.2, 1)
49     listBeta.append(beta0)
50     listGamma.append(gamma0)
51     print("Base beta is", listBeta[0])
52     print("Base gamma is", listGamma[0])
53
54     while ((len(listBeta) < sizeOfSample) and (len(listGamma) < sizeOfSample)):
55         betaVar = listBeta[-1]
56         gammaVar = listGamma[-1]
57         # Normal distribution with sigma = (x* 0.34)^2
58         betaS = np.random.normal(betaVar, decimal.Decimal(betaVar * 0.34) ** 2)
59         # Use decimal when overflow
60         gammaS = np.random.normal(gammaVar, decimal.Decimal(gammaVar * 0.34) ** 2)
61         # print("Beta star is", betaS)
62         # print("Gamma star is", gammaS)
63         betaS = sGamma.pdf(betaS, 1)
64         gammaS = sGamma.pdf(gammaS, 1)
65         r = min(1, (betaS * gammaS) / (betaVar * gammaVar))
66         q = np.random.uniform(0,1) # Uniform distribution (continuous) U(0,1)
67         # print("q is", q)
68
69         if (q < r):
70             listBeta.append(betaS)
71             listGamma.append(gammaS)
72         else:
73             listBeta.append(listBeta[-1])
74             listGamma.append(listGamma[-1])
75
76     i = 0
77     ERo = 0
78     while (i < sizeOfSample):
79         processData(country, listBeta[i], listGamma[i]) * (listBeta[i] / listGamma
80         [i])
81         i += 1
82     print(ERo)
83
84 if __name__ == "__main__":
```

83 `main()`

Listing 4: Python example

4.3 Kết quả và đánh giá

```
1 Ngày 22/01/2020: R0 ~ 0 #xap xi 0
2 Ngày 22/02/2020: R0 = 0
3 Ngày 22/03/2020: R0 = 0
4 Ngày 22/04/2020: R0 = 0
5 Ngày 22/06/2020: R0 = 0
6 Ngày 22/07/2020: R0 = 0
```

Listing 5: Result

Nhận xét: Lúc đầu chính phủ Nhật Bản chưa ban bố tình trạng dịch bệnh cũng như chưa thực hiện các chính sách cách ly và hạn chế đi lại nên hệ số R_0 chỉ gần bằng 0. Sau đó, vào các tháng tiếp theo chính phủ đã thực hiện tốt chính sách kiểm soát dịch bệnh của mình do đó hệ số R_0 đã được kéo về mức bằng 0.



Tài liệu

- [Dal] Dalgaard, P. *Introductory Statistics with R*. Springer 2008.
- [K-Z] Kenett, R. S. and Zacks, S. *Modern Industrial Statistics: with applications in R, MINITAB and JMP*, 2nd ed., John Wiley and Sons, 2014.
- [Ker] Kerns, G. J. *Introduction to Probability and Statistics Using R*, 2nd ed., CRC 2015.
- [Joseph] Joseph T Wu*, Kathy Leung*, Gabriel M Leung *Nowcasting and forecasting the potential domestic and international spread of the 2019-nCoV outbreak originating in Wuhan, China: a modelling study*
- [Tiến] Lê Quang Tiến *MCMC Thuật toán Metropolis–Hastings và lấy mẫu Gibbs*
- [Thạch] Nguyễn Hoàng Thạch, Phan Thị Hà Dương *Tìm hiểu về một Mô hình dự báo dịch Covid-19 từ Vũ Hán*