

Model Epidemiologiczny

P. Szewerniak, K. Oratowski, P. Chajduła, M. Caputa

June 2020



1 Adaptacja kodu ZombieApocalypse

Naszym zadaniem było sprawdzenie jak manipulacja parametrami równań różniczkowych wpływa na kształt krzywej zachorowań. Wzorowaliśmy się na modelu Apokalipsy Zombie, dostarczonym przez prowadzącego.

Taki układ pokazuje jak rozwiązywać układ równań różniczkowych pierwszego rzędu z pomocą bibliotek SciPy. W naszym przypadku jest to układ 3 równań powiązanych zależnościami takimi jak: S - osoby, które mogą się zarazić, Z - liczba zarażonych, R - ilość śmiertelności, P - ilość urodzeń na dzień, d - odsetek naturalnych śmiertelności, B - szansa na zarażenie, A - śmiertelność wirusa

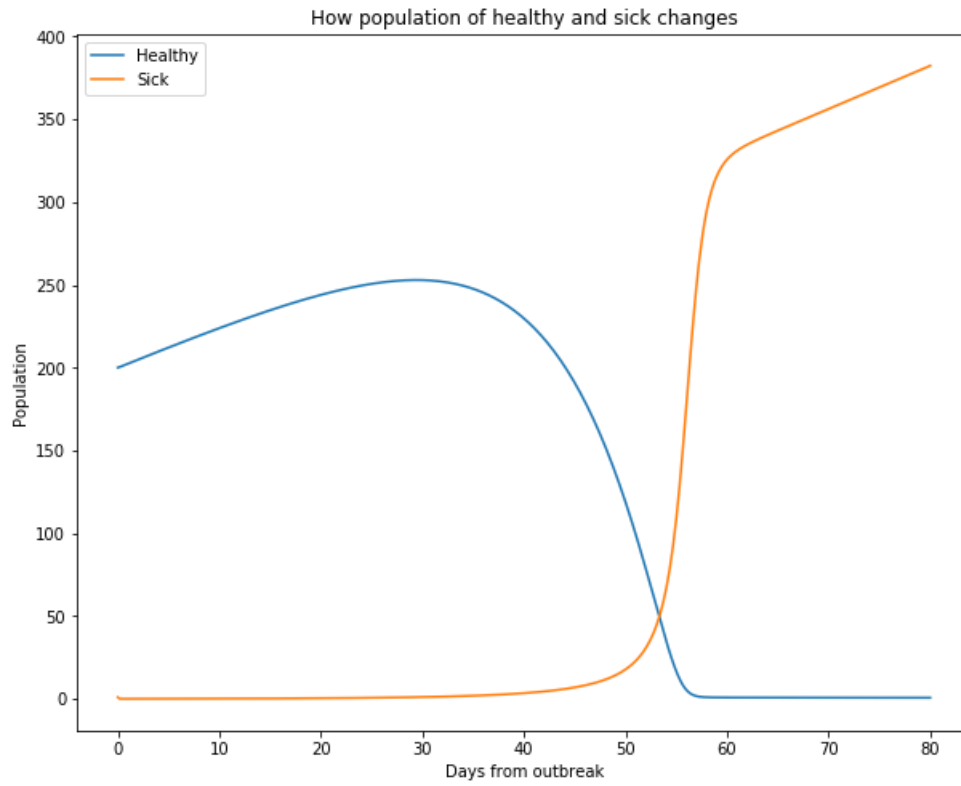
$$\begin{aligned}f_0 &= P - B * S_i * Z_i - d * S_i \\f_1 &= B * S_i * Z_i + R_i - A * S_i * Z_i \\f_2 &= d * S_i + A * S_i * Z_i - R_i\end{aligned}$$

Wykluczaliśmy czynnik G, który odpowiadał za osoby, które po śmierci stają się zakażonymi co w naszej sytuacji nie występuje.

2 Kod w Python

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from scipy.integrate import odeint
4 plt.ion()
5 plt.rcParams['figure.figsize'] = 10, 8
6
7 #initial conditions
8 S0 = 200      #initial population
9 Z0 = 1        #initial sick population
10 R0 = 0        #initial death population
11 y0 = [S0, Z0, R0] #initial condition vector
12 t = np.linspace(0, 80, 500)
13
14 P = 0.013 * S0 #birth rate
15 d = 0.000002 #natural death rate
16 B = 0.01 #transmission percent
17 A = 0.1 #lethality
18
19 def f(y, t):
20     Si = y[0]
21     Zi = y[1]
22     Ri = y[2]
23     f0 = P - B * Si * Zi - d * Si
24     f1 = B * Si * Zi + Ri - A * Si * Zi
25     f2 = d * Si + A * Si * Zi - Ri
26     return [f0, f1, f2]
27
28
29
30 soln = odeint(f, y0, t)
31 S = soln[:, 0]
32 Z = soln[:, 1]
33 R = soln[:, 2]
34 plt.figure()
35 plt.plot(t, S, label = 'Healthy')
36 plt.plot(t, Z, label = 'Sick')
37 plt.xlabel('Days from outbreak')
38 plt.ylabel('Population')
39 plt.title('How population of healthy and sick changes')
40 plt.legend(loc=0)
```

3 Wyniki i rysunki



- Czynniki P - odpowiadający za ilość urodzeń na dzień, określiliśmy biorąc dane z 2015 roku dotyczące liczby urodzeń po czym przeskalowaliśmy ją w stosunku do rozpatrywanej populacji.
- Czynniki d - określiliśmy go biorąc dane z 2015 roku dotyczące liczby zgonów po czym przeskalowaliśmy ją w stosunku do rozpatrywanej populacji.
- Czynniki A - odpowiadający za śmiertelność wirusa. Określiliśmy go na podstawie źródła: <https://wiadomosci.gazeta.pl/wiadomosci/7,114881,25974848,koronawirus-w-eszwecji-smiertelnosc-na-poziomie-10-proc-z-powodu.html>

Współczynnik B który określa wartość prawdopodobieństwa zakażenia się jest problematyczny do wyznaczenia. Nawet anglojęzyczna strona Wikipedii traktująca o COVID-19 podaje „People can transmit the virus without showing symptoms, but according to the WHO and ECDC, it is unknown how often this happens”. Ustawiliśmy ją na wartość jednego procenta tj. osoba zakażona po kontakcie z liczbą stu osób zakazi tylko jedną. Argumentujemy to wprowadzonymi restrykcjami.

Dopiero po około 40 dniach pojawia się pierwszy zakażony co wiąże się z małą wartością współczynnika B. Wystarczy zaledwie parę dni aby przy naszych przyjętych warunkach doszło do bardzo szybkiego skoku zarażonych (przejście fazowe), a następnie do dążenia do maksymalnej wartości tj. liczby wszystkich dostępnych ludzi.

Aby epidemia na naszej stosunkowo małej próbie ustała szansa na zarażenie musiałaby być bardzo mała co nie ma odzwierciedlenia w rzeczywistości.

4 Model SIR

Naszym celem było zbadanie zachowania się krzywych: osób, które mogą się zarazić, zarażonych oraz osób, które się uodporniły po przejściu choroby.

Model SIR jest jednym z najprostszych modeli epidemiologicznych, który powstał już w 1927r. Opiera się on na 3 równaniach różniczkowych pierwszego rzędu powiązanych ze sobą zmianami liczby osób należących do poszczególnych grup oraz dwoma współczynnikami β i γ , gdzie β oznacza średnią liczbę kontaktów powodujących infekcje a γ jest współczynnikiem wyzdrowień (czyli przejścia osobników zainfekowanych po okresie choroby do grupy osobników odpornych na daną infekcję).

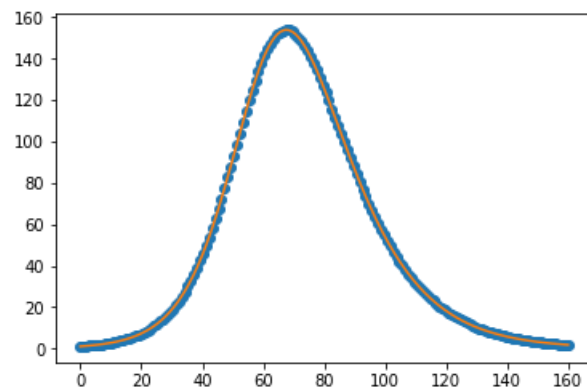
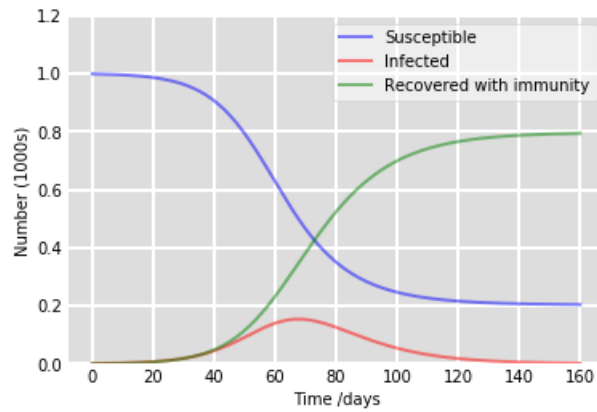
Równania różniczkowe opisujące ten model:

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

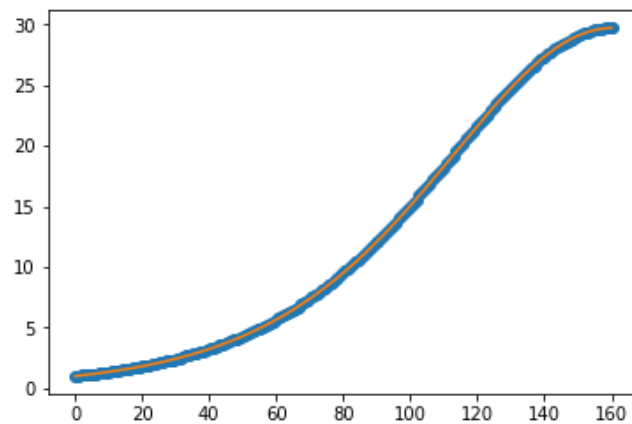
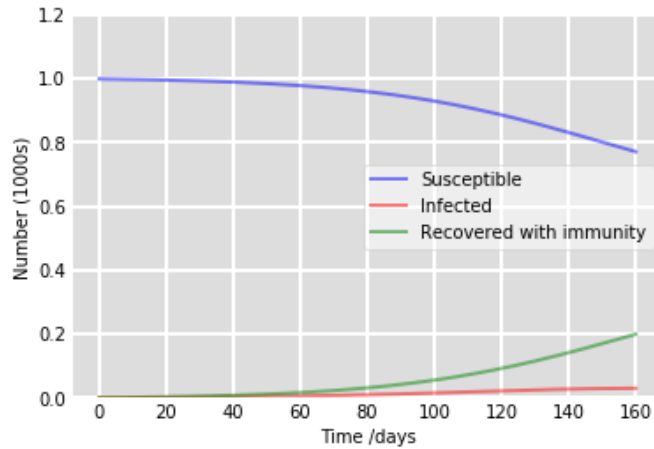
5 Kod w Pythonie

```
1 import numpy as np
2 from scipy.integrate import odeint
3 import matplotlib.pyplot as plt
4 from scipy import interpolate
5
6 # Total population, N.
7 N = 1000
8 # Initial number of infected and recovered individuals, I0 and R0.
9 I0, R0 = 1, 0
10 # Everyone else, S0, is susceptible to infection initially.
11 S0 = N - I0 - R0
12 # Contact rate, beta, and mean recovery rate, gamma, (in 1/days).
13 beta, gamma = 0.2, 1./10
14 # A grid of time points (in days)
15 t = np.linspace(0, 160, 160)
16
17 # The SIR model differential equations.
18 def deriv(y, t, N, beta, gamma):
19     S, I, R = y
20     dSdt = -beta * S * I / N
21     dIdt = beta * S * I / N - gamma * I
22     dRdt = gamma * I
23     return dSdt, dIdt, dRdt
24
25 # Initial conditions vector
26 y0 = S0, I0, R0
27 # Integrate the SIR equations over the time grid, t.
28 ret = odeint(deriv, y0, t, args=(N, beta, gamma))
29 S, I, R = ret.T
30
31 # Plot the data on three separate curves for S(t), I(t) and R(t)
32 fig = plt.figure(facecolor='w')
33 ax = fig.add_subplot(111, facecolor='#dddddd', axisbelow=True)
34 ax.plot(t, S/1000, 'b', alpha=0.5, lw=2, label='Susceptible')
35 ax.plot(t, I/1000, 'r', alpha=0.5, lw=2, label='Infected')
36 ax.plot(t, R/1000, 'g', alpha=0.5, lw=2, label='Recovered with immunity')
37 ax.set_xlabel('Time /days')
38 ax.set_ylabel('Number (1000s)')
39 ax.set_ylim(0,1.2)
40 ax.yaxis.set_tick_params(length=0)
41 ax.xaxis.set_tick_params(length=0)
42 ax.grid(b=True, which='major', c='w', lw=2, ls='-')
43 legend = ax.legend()
44 legend.get_frame().set_alpha(0.5)
45 for spine in ('top', 'right', 'bottom', 'left'):
46     ax.spines[spine].set_visible(False)
47 plt.show()
48
49
50 f = interpolate.interp1d(t, I)
51 xnew = np.linspace(0, 160, num=640)
52 plt.plot(t, I, 'o', xnew, f(xnew), '-')
53 plt.show()
54
55 m = max(f(xnew))
56 print('Po interpolacji wartosc maksymalna wynosi:',m)
```

6 Wyniki i rysunki



('Po interpolacji wartosc maksymalna wynosi:', 153.9004584409455)



('Po interpolacji wartosc maksymalna wynosi:', 29.718188294046485)

Dla naszego modelu olbrzymie znaczenie maja parametry β i γ . Poczatkowo przyjeliśmy $\beta = 0.2$ i $\gamma = 1/10$. Juz nawet zmiana pierwszego parametru o 0.07 całkowicie wypłaszcza nam krzywa zachorowań zanim w ogóle wirus bedzie miał szanse sie rozprzestrzenić. W pierwszym przypadku wartość zachorowań osiągnęła liczbę 153 osób a w drugim ledwie 29 co jest olbrzymia różnica zważając na niewielka modyfikacje parametrów.

7 Bibliografia

- https://en.wikipedia.org/wiki/Compartmental_models_in_epidemiologyThe_SIR_model
- <https://scipython.com/book/chapter-8-scipy/additional-examples/the-sir-epidemic-model/>
- https://www.fuw.edu.pl/~jarekz/MODELOWANIE/modele_epidemii.pdf
- https://pg.edu.pl/documents/70523227/96262108/Sesja%20%20207%20Prezentacja_eTee_2019_%2816x9%29_AB_MW.pdf