

```
In[1]:= SetOptions[EvaluationNotebook[], CellContext -> Notebook]
```

Modelowanie dynamiki epidemii

Marcin Choiński

Uniwersytet Warszawski

Wydział Matematyki, Informatyki i Mechaniki

kierunek matematyka

mc380398@mimuw.edu.pl

Oświadczam, że projekt został wykonany przeze mnie.

Wyrażam zgodę na umieszczenie projektu na stronie internetowej kursu w portalu Moodle.

Model SI

Najprostszy model epidemiologiczny to model typu SI. W modelu tym opisujemy infekcję, na którą osobniki rozważanej populacji nie uodparniają się. Wyróżniamy tylko dwie grupy - osoby zdrowe i osoby chore. Niech $S(t)$ oznacza część populacji, która jest zdrowa, natomiast niech $I(t)$ oznacza część populacji, która jest chora. Budując model, postulujemy, że

- wszystkie nowo narodzone osobniki są zdrowe, infekcja rozprzestrzenia się poziomo w populacji, nie ma transmisji pionowej;
 - infekcja rozprzestrzenia się przez kontakt między osobnikami podatnymi i chorymi;
 - kontakty między osobnikami są losowe, ich liczba jest proporcjonalna do liczebności poszczególnych klas osobników;
 - po przebyciu infekcji osobniki zdrowieją i wracają do grupy podatnych.

Przy tych założeniach, skalując wielkość populacji do 1, otrzymujemy układ dwóch równań:

$$(1) S' = \mu - \beta I S + \gamma I - \mu S,$$

$$(2) I' = \beta I S - \gamma I - \mu I,$$

gdzie:

- S, I - proporcje osób podatnych i chorych w populacji, $S + I = 1$,
- μ - współczynnik rozrodczości/śmiertelności,
- β - współczynnik kontaktów/zakaźności, który odzwierciedla część kontaktów prowadzących do zachorowania,
- γ - współczynnik wyzdrowień, $1/\gamma$ odzwierciedla średni czas trwania infekcji.

In[2]:=

Wygasanie epidemii

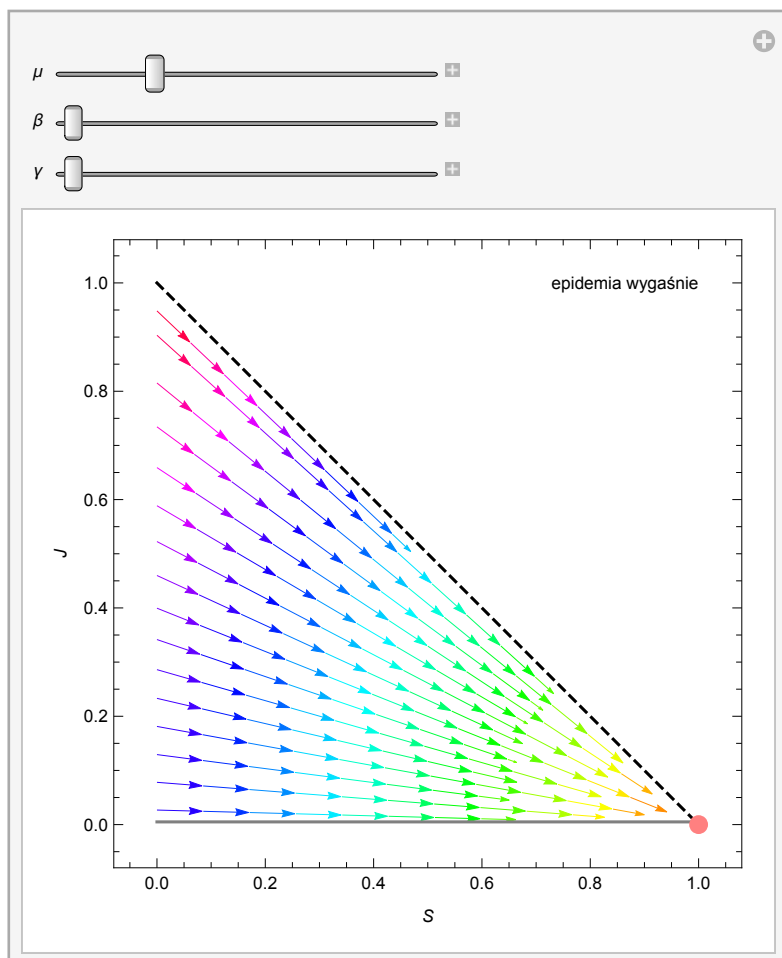
Oznaczmy $R_0 = \frac{\beta}{\gamma + \mu}$. Współczynnik R_0 nazywamy podstawowym progiem odnowienia dla danej infekcji. Jeżeli $R_0 \leq 1$, to $(S, I) \rightarrow (1, 0)$ dla $t \rightarrow \infty$, czyli infekcja wygasa i populacja wraca do normalnego stanu. Jeżeli $R_0 > 1$, to $(S, I) \rightarrow \left(\frac{1}{R_0}, 1 - \frac{1}{R_0}\right)$ dla $t \rightarrow \infty$, czyli infekcja nie wygasa - ustala się stan epidemii. Dynamikę epidemii modeluje się za pomocą portretu fazowego, który przedstawia zależność rozwiązań układu równań opisującego modelu. W kodzie zagęszczenie (w kolejnych częściach projektu - liczebność) populacji osób zainfekowanych oznacza się przez J zamiast I .

```

In[3]:= Manipulate[
  Show[ (* wektory pola, pokazuje kierunek rozwiązań,
        kolor strzałek obrazuje normę wektorów pola*)
    StreamPlot[{ $\mu - \beta * S * J + \gamma * J - \mu * S$ ,  $\beta * S * J - \gamma * J - \mu * J$ }, {S, 0, 1},
      {J, 0, 1}, RegionFunction -> Function[{S, J},  $S + J \leq 1$ ],
      StreamColorFunction -> Hue, FrameLabel -> {S, J}],
    (* izokliny zerowe*) Plot[{ $(\mu * S - \mu) / (-\beta * S + \gamma)$ , 0.005}, {S, 0, 1},
      PlotRange -> Automatic, PlotStyle -> {Brown, Gray}],
    (* krzywa dająca warunek  $S+I \leq 1$  *) Plot[1 - S, {S, 0, 1},
      PlotRange -> Automatic, PlotStyle -> {Black, Dashed}],
    ParametricPlot[{ $(\gamma + \mu) / \beta$ , J}, {J, 0, 1}, PlotStyle -> {Gray, Dashed}, Frame -> True],
    (* sprawdzanie, czy epidemia wygaśnie *)
    If[ $\beta / (\gamma + \mu) \leq 1$ , Graphics[Text["epidemia wygaśnie", {1, 1}, {1, 0}]],
      Graphics[Text["epidemia nie wygaśnie", {1, 1}, {1, 0}]]],
    (* punkt równowagi *) Graphics[
      {Pink, PointSize[0.03], Point[
        {S, J} /. NSolve[ $\mu - \beta * S * J + \gamma * J - \mu * S == 0 \ \&\& \ \beta * S * J - \gamma * J - \mu * J == 0$ , {S, J}]
      ] }
    ]
  ],
  { $\mu$ , 0.0001, 1}, { $\beta$ , 0.0001, 1}, { $\gamma$ , 0.0001, 1}, SaveDefinitions -> True]

```

Out[3]=



Symulacja modelu

Kiedy znane są wartości wszystkich parametrów opisujących model epidemiologiczny oraz wartości początkowe funkcji, czyli liczba osobników chorych i zainfekowanych, można przeprowadzić symulację modelu za pomocą modelu dyskretnego, przy użyciu na przykład dyskretyzacji metodą Eulera. W symulacji użyto zmodyfikowanego modelu opisanego równaniami różniczkowymi:

$$(1) S' = \lambda - \beta I S + \gamma I - \delta S, \quad (2) I' = \beta I S - \gamma I - \delta I - \alpha I,$$

gdzie λ - stały napływ osobników do populacji, δ - współczynnik śmiertelności naturalnej, α - współczynnik śmiertelności związanej z chorobą. Dzięki zmianom opis dynamiki populacji jest bardziej realistyczny. W symulacji uwzględniono górny i dolny próg liczebności populacji, dzięki czemu można sprawdzać, w którym momencie populacja może wyginąć powodu epidemii lub kiedy nastąpi przeludnienie - jeżeli założymy brak infekcji.

In[4]:=

```

Manipulate[ (* pure function na obliczenie liczby osób zdrowych i chorych, podwójna
  NestWhile List z ograniczeniami na maksymalną i minimalną liczbę populacji *)
  bezinfekcji = NestWhileList[ {Max[0, Round[ naplyw + (1 - as) * #[[1]] -
    beta * #[[1]] * #[[2]] / (#[[1]] + #[[2]]) + gamma * #[[2]] ]}, Max[0, Round[
    beta * #[[1]] * #[[2]] / (#[[1]] + #[[2]]) + (1 - (ac + as + gamma)) * #[[2]]]}] &,
    {S0, I0}, up > (#[[1]] + #[[2]]) > down &, 1, okres] ;
  (* tworzenie tabeli w celu wykonania wykresu ListPlot *)
  Table[bezinfekcji[[i]][[1]], {i, 1, Length[bezinfekcji]}];
  (* dwa wykresy, jeden dotyczący osób zdrowych, drugi dotyczący osób chorych *)
  wykres = ListPlot[
    {Labeled[Table[bezinfekcji[[i]][[1]], {i, 1, Length[bezinfekcji]}], "zdrowi"],

     Labeled[ Table[bezinfekcji[[i]][[2]], {i, 1, Length[bezinfekcji]}], "chorzy"]},
    Joined → True, PlotRange → All],

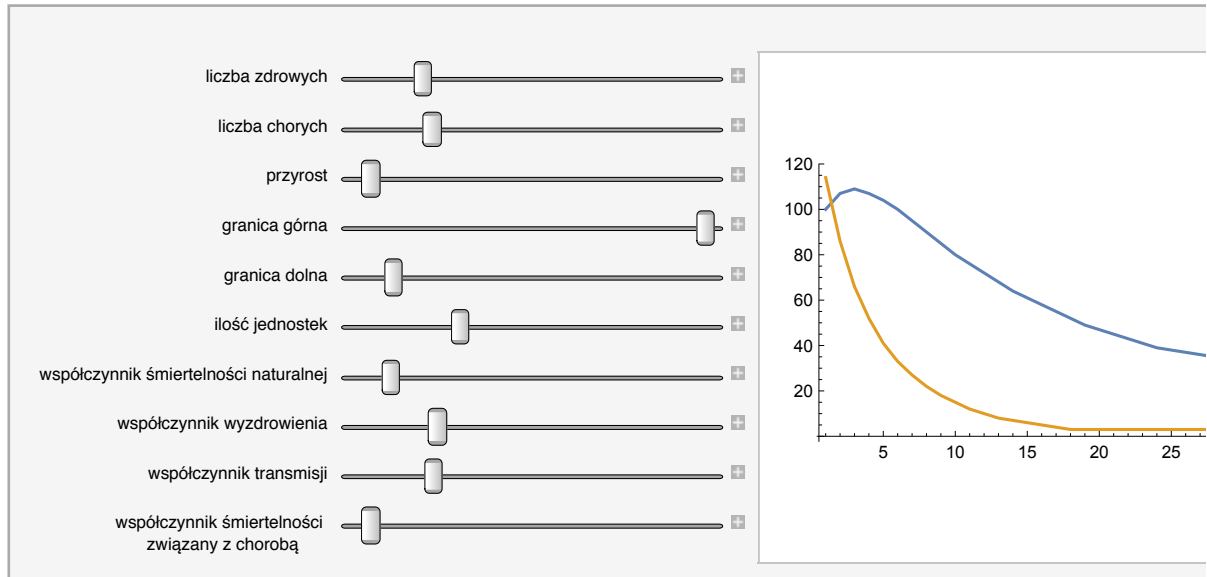
  (* przypisywanie etykietom nazwy zmiennych z wartościami początkowymi *)

  {{S0, 100, "liczba zdrowych"}, 10, 500, 1},
  {{I0, 100, "liczba chorych"}, 10, 500, 1}, {{naplyw, 1, "przyrost"}, 1, 30, 1},

  {{up, 1000, "granica górna"}, 500, 1000, 1},
  {{down, 5, "granica dolna"}, 1, 100, 1}, {{okres, 30, "ilość jednostek"}, 1, 100, 1},
  {{as, 0.001, "współczynnik śmiertelności naturalnej"}, 0.001, 0.95},
  {{gamma, 0.001, "współczynnik wyzdrowienia"}, 0.001, 0.999},
  {{beta, 0.001, "współczynnik transmisji"}, 0.001, 0.999},
  {{ac, 0.001, "współczynnik śmiertelności związany z chorobą"}, 0.001, 0.999},
  SaveDefinitions → True]

```

Out[4]=



Szacowanie współczynnika transmisji choroby

Współczynnik transmisji, oznaczany przez β , jest definiowany jako stopień, z jakim następuje przeniesienie czynnika chorobotwórczego z osób zainfekowanych na osoby zdrowe, podatne na choroby. Współczynnik ten zależy między innymi od liczebności osób zdrowych i zainfekowanych w populacji, dlatego nie można określić stałej wartości współczynnika dla określonej choroby. Współczynnik należy oszacować dla poszczególnego zestawu danych. W programie estymowana jest wartość współczynnika transmisji dla określonych liczb osób zdrowych i chorych. Wielkość populacji jest stała, liczebności grupy osób zdrowych w danym czasie są wygenerowane losowo. Dla poszczególnych wartości β z przedziału $[0,1]$ przeprowadzana jest symulacja modelu dla początkowych wielkości subpopulacji osób chorych i zdrowych, następnie obliczany jest kwadrat różnicy liczby osób zdrowych wyznaczonej symulacją i liczby osób zdrowych wygenerowanej losowo w czasie t , wszystkie różnice są zsumowane. Nie wyznacza się kwadratów różnic dla subpopulacji osób chorych, gdyż ze stałości liczebności populacji wynika, że liczba osób chorych jest zależna od liczby osób zdrowych. Na końcu wybiera się tę wartość β , dla której suma kwadratów różnic jest najmniejsza.

```
In[5]:= SR = RandomInteger[100, 10];
JR = Table[100 - SR[[j]], {j, 1, 10}];

n = 100;
T = Length[SR]; Manipulate[ Jt = Table[0, {j, 1, n}, {i, 1, T}];
St = Table[0, {j, 1, n}, {i, 1, T}]; betat = Table[j / n, {j, 1, n}];
For[j = 1, j < n + 1, j++, St[[j]][[1]] = SR[[1]]; Jt[[j]][[1]] = JR[[1]] ];
total = St[[1]][[1]] + Jt[[1]][[1]];
For[k = 1, k < n + 1, k++, For[i = 2, i < T + 1, i++,
  St[[k]][[i]] = Round[St[[k]][[i - 1]] + (total) * ni - betat[[k]] * St[[k]][[i - 1]] *
  Jt[[k]][[i - 1]] / total + gamma * Jt[[k]][[i - 1]] - ni * St[[k]][[i - 1]]];
  Jt[[k]][[i]] = Round[total - St[[k]][[i]]];
  If[Jt[[k]][[i]] < 0, Jt[[k]][[i]] = 0]; ];
Sroznice = Table[0, {j, 1, n}, {i, 1, T}];
For[k = 1, k < n + 1, k++,
  For[i = 1, i < T + 1, i++, Sroznice[[k]][[i]] = Power[SR[[i]] - St[[k]][[i]], 2];];
Ssum = Table[0, {j, 1, n}];
For[k = 1, k < n + 1, k++,
  For[i = 1, i < T + 1, i++, Ssum[[k]] = Ssum[[k]] + Sroznice[[k]][[i]]; ];
Flatten[Position[Ssum, Min[Ssum]]][[1]];

betaopt = N[betat[[Flatten[Position[Ssum, Min[Ssum]]][[1]]]];
Grid[{{"lista zdrowych", SR}, {"lista chorych", JR}, {"beta", betaopt}}, Frame -> All],
{gamma, 0.001, 1}, {ni, 0.001, 1}, SynchronousUpdating -> False, SaveDefinitions -> True]
```

Out[8]=

gamma

ni

+

Null ⁹	lista zdrowych	{72, 57, 94, 47, 20, 41, 43, 29, 56, 42}
	lista chorych	{28, 43, 6, 53, 80, 59, 57, 71, 44, 58}
	beta	0.2