

specjalność: Analityk danych

Aleksandra Burczyk

Monika Gmaj

Kinga Jankowska

Kamil Łangowski

PROJEKT 2

raport

Modelowanie matematyczne i symulacje komputerowe,
semestr zimowy 2022/2023

Spis treści

Zadanie 1.	5
Zadanie 2. Wprowadzenie	6
Zadanie 2.1.	6
Zadanie 2.2.	7
a) $(S_0, I_0, \beta, \gamma, \mu) = (N - 1, 1, 0.5, 0.8, 0.4)$	8
b) $(S_0, I_0, \beta, \gamma, \mu) = (N - 1, 1, 0.2, 0.04, 0.01)$	14
c) $(S_0, I_0, \beta, \gamma, \mu) = (N - 1, 1, 0.8, 0.4, 0.01)$	20
d) $(S_0, I_0, \beta, \gamma, \mu) = (N - 1, 1, 0.8, 0.7, 0.01)$	26
Zadanie 3.	32
b)	33
c)	35
d)	39
Zadanie 4.	42
a)	42
b)	43
c)	44
d)	45
b) dla różnych liczebności populacji	46

Zadanie 5.	50
Zadanie 6.	51
Zadanie 7.	53
Kod	55

Deterministyczne modele kompartmentowe - model SIR z demografią

Rozważamy matematyczny model dynamiki populacji otwartej, w której pod uwagę brać będziemy zarówno urodzenia, jak i zgony osobników danej populacji.

Zakładamy, że w populacji urodzenia oraz zgony osobników równoważą się, czyli populacja ma ustaloną wielkość N .

Podczas, gdy społeczeństwo dotyka infekcja, populację podzielić możemy na trzy różne grupy:

- grupę podatnych (ang. **S**usceptible)
- grupę zainfekowanych (ang. **I**nfected)
- grupę ozdrowiałych (ang. **R**ecovered).

W danym momencie t mamy, że:

- $S(t)$ odpowiada liczbie osób jeszcze nie zainfekowanych, ale podatnych na infekcję
- $I(t)$ odpowiada liczbie osób zainfekowanych
- $R(t)$ odpowiada liczbie osób, które przeszły infekcję, ale już wyzdrowiały.

Zakładamy także, że jeśli osobnik miał infekcję i wyzdrowiał, to nie może zarazić się ponownie.

Dla każdego t mamy następującą zależność

$$S(t) + I(t) + R(t) = N, \quad (1)$$

która implikuje równość

$$\frac{S(t)}{dt} + \frac{I(t)}{dt} + \frac{R(t)}{dt} = 0. \quad (2)$$

Modelując przyjmujemy poniższe założenia.

- Osobnicy podatni (grupa S) mogą zostać zainfekowani w każdym momencie t . Ich liczebność dla danego t zależy od liczby osób, które zmarły, urodziły się, przeszły do grupy zainfekowanych I .
- Osobnicy zainfekowani (grupa I) mogą z czasem wyzdrowieć (prześć do grupy R) lub umrzeć.
- Osobnicy wyzdrowiali (grupa R) nie mogą wrócić do grupy S , mogą pozostać w grupie R lub umrzeć.

Powyższe założenia wraz z (1) dają nam kompartmentowy model opisany układem równań różniczkowych

$$\begin{cases} \frac{d}{dt}S(t) = \mu N - \frac{\beta}{N}S(t)I(t) - \mu S(t), \\ \frac{d}{dt}I(t) = \frac{\beta}{N}S(t)I(t) - \gamma I(t) - \mu I(t), \\ \frac{d}{dt}R(t) = \gamma I(t) - \mu R(t), \end{cases} \quad (3)$$

gdzie:

- $\beta > 0$ to współczynnik przejścia z grupy podatnych do grupy zainfekowanych,
- $\gamma > 0$ to współczynnik zdrowienia,
- $\mu > 0$ to współczynnik narodzin i jednocześnie śmiertelności (tyle ile osobników się narodzi, tyle umiera w każdym momencie t).

Zakładamy także, że naturalna długość życia osobnika populacji wynosi $\frac{1}{\mu}$ roku.

Korzystając z (1) powyższy model rozpatrzyć możemy w następującej postaci:

$$\begin{cases} \frac{d}{dt}S(t) = \mu N - \frac{\beta}{N}S(t)I(t) - \mu S(t), \\ \frac{d}{dt}I(t) = \frac{\beta}{N}S(t)I(t) - \gamma I(t) - \mu I(t), \\ R(t) = N - S(t) - I(t). \end{cases} \quad (4)$$

W poniższych zadaniach przyjmujemy, że $t \in [0, T]$, gdzie dla $t = 0$ mamy warunek początkowy

$$S(0) = S_0, \quad I(0) = I_0.$$

W naszym projekcie analizie poddamy rozwiązanie numeryczne powyższego problemu różniczkowego uzyskane niestandardową metodą Mickens'a.

Analiza zachowania długoterminowego - symulacje rozwiązań

Dla modeli epidemiologicznych definiujemy tzw. podstawowy współczynnik reprodukcji R_0 (ang. basic reproduction number), który można traktować jako oczekiwaną liczbę przypadków zakażeń bezpośrednio generowanych przez jednego chorego osobnika w populacji, w której wszystkie osobniki są podatne na zakażenie. Definicja współczynnika R_0 odnosi się do sytuacji kiedy żadne inne osoby w populacji nie są zarażone ani zaszczepione. Współczynnik R_0 dla modelu SIR z demografią definiuje się wzorem:

$$R_0 = \frac{\beta}{\mu + \gamma}. \quad (5)$$

Według wartości R_0 możemy określić, czy dla konkretnych parametrów epidemia wybuchnie ($R_0 > 1$), czy też szybko wygaśnie nie rozprzestrzeniając się ($R_0 < 1$). Ponadto, model posiada dwa punkty równowagi:

- punkt równowagi **wolny od choroby** - stabilny, gdy $R_0 < 1$,
- punkt **endemiczny** - stabilny, gdy $R_0 > 1$.

Zadanie 1.

Korzystając z układu równań (3) wyznaczmy punkty równowagi modelu, będącego głównym tematem naszego projektu, przyrównując równania do zera i rozwiązując ten układ równań.

$$\begin{cases} \mu N - \frac{\beta}{N} S(t) I(t) - \mu S(t) &= 0, \\ \frac{\beta}{N} S(t) I(t) - \gamma I(t) - \mu I(t) &= 0, \\ \gamma I(t) - \mu R(t) &= 0. \end{cases}$$

Poniżej rozwiążemy ten układ równań krok po kroku. Zaczniemy od drugiego równania.

$$I(t) \left(\frac{\beta}{N} S(t) - \gamma - \mu \right) = 0$$

Rozważamy dwa przypadki.

$$I(t) = 0$$

Podstawiając powyższą równość do pierwszego równania otrzymujemy:

$$\mu N - \frac{\beta}{N} S(t) \cdot 0 - \mu S(t) = 0$$

$$\mu N - \mu S(t) = 0$$

$$\mu S(t) = \mu N$$

$$S(t) = N$$

Podstawiając $I(t) = 0$ do trzeciego równania otrzymujemy:

$$\gamma \cdot 0 - \mu R(t) = 0.$$

$$R(t) = 0.$$

Otrzymujemy pierwszy punkt równowagi.

$$\begin{cases} \mathbf{S}^*(t) &= N \\ \mathbf{I}^*(t) &= 0 \\ \mathbf{R}^*(t) &= 0 \end{cases}$$

$$\frac{\beta}{N} S(t) - \gamma - \mu = 0$$

Wyznaczamy wartość $S(t)$.

$$\frac{\beta}{N} S(t) = \mu + \gamma$$

$$S(t) = \frac{N}{\beta} (\mu + \gamma)$$

Podstawiamy $S(t)$ do pierwszego równania, by wyznaczyć $I(t)$.

$$-\frac{\beta}{N} S(t) I(t) = \mu (S(t) - N)$$

$$-\frac{\beta}{N} \frac{N}{\beta} (\mu + \gamma) I(t) = \mu \left(\frac{N}{\beta} (\mu + \gamma) - N \right)$$

$$-(\mu + \gamma) I(t) = \frac{\mu(N\mu + N\gamma - N\beta)}{\beta}$$

$$I(t) = \frac{\mu(N\beta - N\gamma - N\mu)}{\beta(\mu + \gamma)}$$

Teraz podstawiając $I(t)$ do trzeciego równania otrzymamy $R(t)$.

$$R(t) = \frac{\gamma}{\mu} I(t)$$

$$R(t) = \frac{\gamma}{\mu} \frac{\mu(N\beta - N\gamma - N\mu)}{\beta(\mu + \gamma)}$$

$$R(t) = \frac{\gamma(N\beta - N\gamma - N\mu)}{\beta(\mu + \gamma)}$$

Otrzymujemy drugi punkt równowagi.

$$\begin{cases} \mathbf{S}^*(t) &= \frac{N}{\beta} (\mu + \gamma) \\ \mathbf{I}^*(t) &= \frac{\mu(N\beta - N\gamma - N\mu)}{\beta(\mu + \gamma)} \\ \mathbf{R}^*(t) &= \frac{\gamma(N\beta - N\gamma - N\mu)}{\beta(\mu + \gamma)} \end{cases}$$

Wykorzystamy definicję R_0 (5) by w uproszczonej postaci zapisać punkty równowagi.

$$\begin{cases} \mathbf{S}^*(t) &= N \\ \mathbf{I}^*(t) &= 0 \\ \mathbf{R}^*(t) &= 0 \end{cases} \quad \begin{cases} \mathbf{S}^*(t) &= \frac{N}{R_0} \\ \mathbf{I}^*(t) &= \frac{N\mu(R_0-1)}{\beta} \\ \mathbf{R}^*(t) &= \frac{N\gamma(R_0-1)}{\beta} \end{cases}$$

W zadaniach 2. oraz 3. punktu $(S^*, I^*, R^*) = (N, 0, 0)$ nie będziemy obliczać, przyjmujemy liczebność $N = 10000$, więc punkt ten ma następujące współrzędne $(S^*, I^*, R^*) = (10000, 0, 0)$ i jest punktem wolnym od choroby.

Zadanie 2.

Przyjmujemy N , czyli liczebność populacji równą 10 000 oraz krok czasowy $T = 1000$ dni (około 3 lata). Przyjmujemy także krok obliczeniowy $h = 0.1$, czyli ilość wykonanych iteracji metody to $n = \frac{T}{h} = \frac{1000}{0.1} = 10000$.

2.1.

Wprowadzimy metodę niestandardową typu Mickensa dla modelu (3).

$$\begin{cases} \frac{S_{n+1}-S_n}{h} &= \mu N - \frac{\beta}{N} I_n S_{n+1} - \mu S_{n+1} \\ \frac{I_{n+1}-I_n}{h} &= -\mu I_{n+1} - \gamma I_{n+1} + \frac{\beta}{N} S_{n+1} I_n \\ \frac{R_{n+1}-R_n}{h} &= -\mu R_{n+1} + \gamma I_{n+1} \end{cases}$$

$$\begin{cases} S_{n+1} &= \mu h N - \frac{\beta h}{N} I_n S_{n+1} - \mu h S_{n+1} + S_n \\ I_{n+1} &= -\mu h I_{n+1} - \gamma h I_{n+1} + \frac{\beta h}{N} S_{n+1} I_n + I_n \\ R_{n+1} &= -\mu h R_{n+1} + \gamma h I_{n+1} + R_n \end{cases}$$

$$\begin{cases} S_{n+1} + \frac{\beta h}{N} I_n S_{n+1} + \mu h S_{n+1} &= \mu h N + S_n \\ I_{n+1} + \mu h I_{n+1} + \gamma h I_{n+1} &= \frac{\beta h}{N} S_{n+1} I_n + I_n \\ R_{n+1} + \mu h R_{n+1} &= \gamma h I_{n+1} + R_n \end{cases}$$

$$\begin{cases} S_{n+1} \left(1 + \frac{\beta h}{N} I_n + \mu h\right) &= \mu h N + S_n \\ I_{n+1} (1 + \mu h + \gamma h) &= \frac{\beta h}{N} S_{n+1} I_n + I_n \\ R_{n+1} (1 + \mu h) &= \gamma h I_{n+1} + R_n \end{cases}$$

$$\begin{cases} S_{n+1} &= \frac{\mu h N + S_n}{1 + \frac{\beta h}{N} I_n + \mu h} \\ I_{n+1} &= \frac{\frac{\beta h}{N} S_{n+1} I_n + I_n}{1 + \mu h + \gamma h} \\ R_{n+1} &= \frac{\gamma h I_{n+1} + R_n}{1 + \mu h} \end{cases}$$

Implementujemy ją w środowisku R w poniższy sposób.

Zaznaczamy także, że dla ułatwienia, w kodzie zamiast liter greckich użyte zostały pierwszych liter ich nazw, tj. w częściach, gdzie występuje kod przyjmujemy, że $\beta = b$, $\gamma = g$, $\mu = m$. A także $T = TT$, gdyż samo T występuje w R jako funkcja wbudowana, co zaburza działanie programu.

```

S_n1 <- function(h, m, N, Sn, b, In) return((h*m*N+Sn)/(1+h*m+(b/N)*h*In))

I_n1 <- function(b, N, h, S_n1, In, g, m) return(((b/N)*h*S_n1*In + In)/(1+g*h+m*h))

R_n1 <- function(g, h, I_n1, Rn, m) return((g*h*I_n1+Rn)/(1+m*h))

simulation_mickens <- function(S_0, I_0, R_0, h, b, g, m, TT, N) {
  n = TT / h

  S={}
  S[1] <- S_0

  I={}
  I[1] <- I_0

  R={}
  R[1] <- R_0

  for(i in 2:n) {
    S_n1 = S_n1(h, m, N, S[i-1], b, I[i-1])
    S[i] = S_n1

    I_n1 = I_n1(b, N, h, S[i], I[i-1], g, m)
    I[i] = I_n1

    R_n1 = R_n1(g, h, I[i], R[i-1], m)
    R[i] = R_n1
  }
  return(list(S,I,R))
}

```

2.2.

Po zaimplementowaniu poniższej metody, możemy przejść do wykonywania symulacji na konkretnych parametrach. Krok ten dzielimy na cztery etapy (dla czterech zestawów parametrów) i do każdego z nich zapisujemy wnioski:

- wyznaczmy punkt równowagi oraz bazowy współczynnik reprodukcji R_0 , czyli teoretyczną liczbę osób, którą zaraża każdy chory, przy założeniu, że wszystkie jego kontakty następują z osobami podatnymi na zakażenie
- przedstawimy na wykresach rozwiązanie numeryczne modelu **S I R**
- wyznaczmy portrety fazowe dla par $I \sim S$, $S \sim R$, $I \sim R$ wraz z zaznaczonymi punktami początkowymi oraz punktami równowagi
- wyznaczmy trójwymiarowy portret fazowy $S \sim I \sim R$.

Zanim przejdziemy do przykładów, to zaimplementujemy jeszcze funkcje wyznaczające punkty równowagi oraz wartości R_0 zgodnie ze wyznaczonymi wcześniej wzorami.

```

R_0 <- function(b, g, m) return( b / (g + m))

pkt_S <- function(N, R_0) return (N/R_0)

pkt_I <- function(N, m, b, R_0) return ((N * m * (R_0 - 1)) / (b))

pkt_R <- function(N, g, b, R_0) return ((N * g * (R_0 - 1)) / (b))

```

Możemy już przejść do przykładów.

2.2.a)

Rozważany zestaw parametrów:

a) $(S_0, I_0, \beta, \gamma, \mu) = (N - 1, 1, 0.5, 0.8, 0.4)$.

- Na początku wyznaczamy wartość współczynnika R_0 korzystając ze wcześniej zaimplementowanej funkcji.

```
R0_a = R_0(b = 0.5, g = 0.8, m = 0.4)
```

Otrzymaliśmy wynik:

$$R_{0_a} \approx 0.4166667 < 1$$

Następnie wyznaczamy punkt równowagi dla parametrów z tego przykładu.

```
rownowagaSIR_a <- c(pkt_S(N = 10000, R0_a),  
                    pkt_I(N = 10000, b = 0.5, m = 0.4, R0_a),  
                    pkt_R(N = 10000, b = 0.5, g = 0.8, R0_a))
```

Punkt równowagi ma następujące współrzędne.

$$\begin{cases} S_a &= 24000.000 \\ I_a &= -4666.667 \\ R_a &= -9333.333 \end{cases}$$

W N I O S E K: R_{0_a} ma wartość mniejszą od jedynki, co sugeruje, że epidemia szybko wygaśnie nie rozprzestrzeniając się. W dalszej części sprawdzimy, czy punkt równowagi określony powyższymi współrzędnymi, jest stabilny, czy też nie. Podejrzewamy już jednak, że punkt ten nie dopasuje się ładnie do naszego modelu, bowiem współrzędne I oraz R są ujemne, co oznaczałoby, że jest ujemna liczba zarówno osób zainfekowanych, jak i ozdrowiałych, co nie ma sensu. Współrzędna punktu równowagi dla grupy S zaś przekracza maksymalną pojemność populacji, która wynosi 10 000.

- Przeprowadzamy symulację.

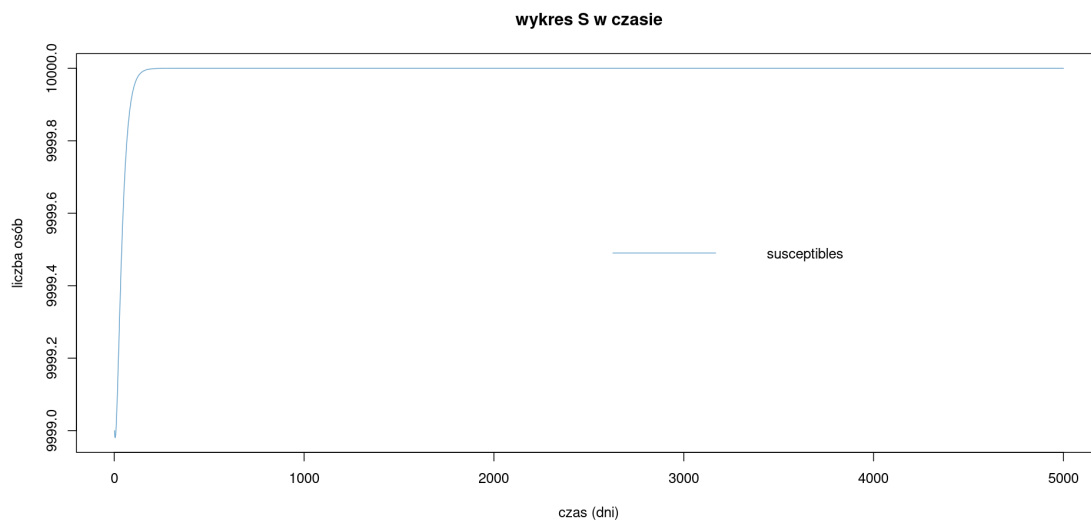
```
x_a = simulation_mickens(9999, 1, 0, h = 0.1, b = 0.5, g = 0.8, m = 0.4, TT = 1000, N = 10000)  
S_a = x_a[[1]]  
I_a = x_a[[2]]  
R_a = x_a[[3]]  
df_a = data.frame(S_a, I_a, R_a)
```

Przedstawimy ilość osób należącą do poszczególnych grup S, I, R dla pierwszych dziesięciu iteracji.

	S_a	I_a	R_a
1	9999.000	1.0000000000	0.00000000
2	9998.990	0.9374954928	0.07211504
3	9998.984	0.8788977729	0.13694890
4	9998.981	0.8239626638	0.19506338
5	9998.981	0.7724612474	0.24698104
6	9998.983	0.7241789111	0.29318784
7	9998.987	0.6789144540	0.33413557
8	9998.993	0.6364792494	0.37024415
9	9999.001	0.5966964588	0.40190371
10	9999.011	0.5594002960	0.42947667

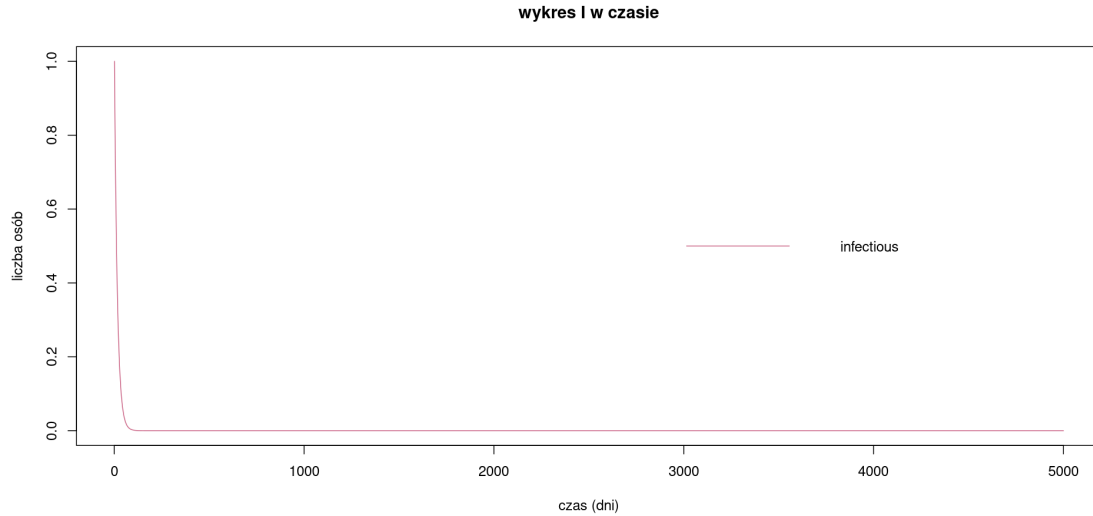
Następnie wyniki te przedstawimy na wykresach (trzech różnych, ze względu na duże różnice w skali osi y).

```
plot(c(1:length(S_a)), S_a, main = "wykres_S_w_czasie",
type = "l", col = "skyblue3",
xlab = "czas_", ylab = "liczba_os b")
legend("right", c("susceptibles"), col = c("skyblue3"), lty = 1, bty = "n")
```



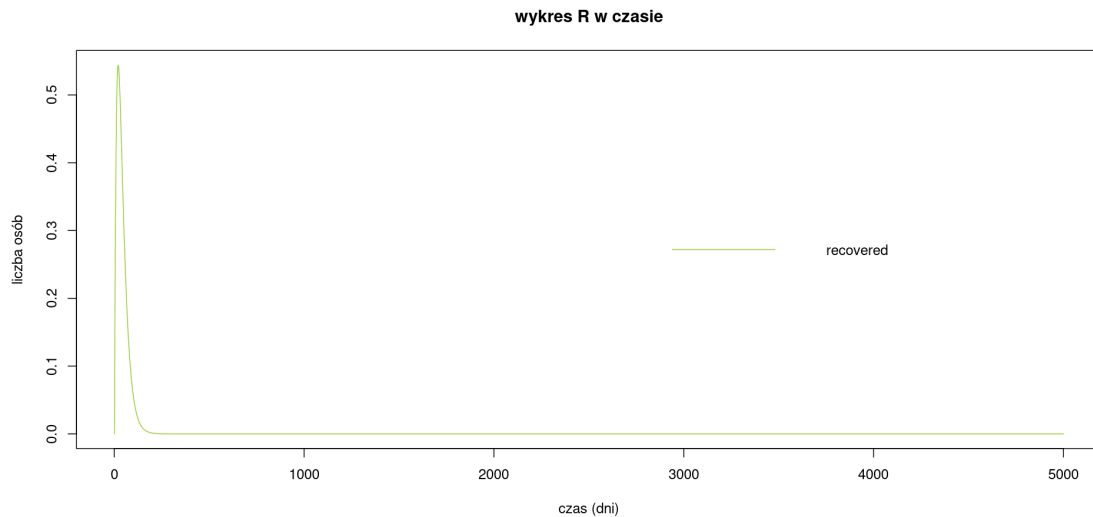
Rysunek 1: Wykres **S** w czasie, przy parametrach a) $(S_0, I_0, \beta, \gamma, \mu) = (N - 1, 1, 0.5, 0.8, 0.4)$

```
plot(c(1:length(I_a)), I_a, main = "wykres_I_w_czasie", type = "l", col = "palevioletred3",
xlab = "czas_", ylab = "liczba_os b")
legend("right", c("infectious"), col = c("palevioletred3"), lty = 1, bty = "n")
```



Rysunek 2: Wykres **I** w czasie, przy parametrach a) $(S_0, I_0, \beta, \gamma, \mu) = (N - 1, 1, 0.5, 0.8, 0.4)$

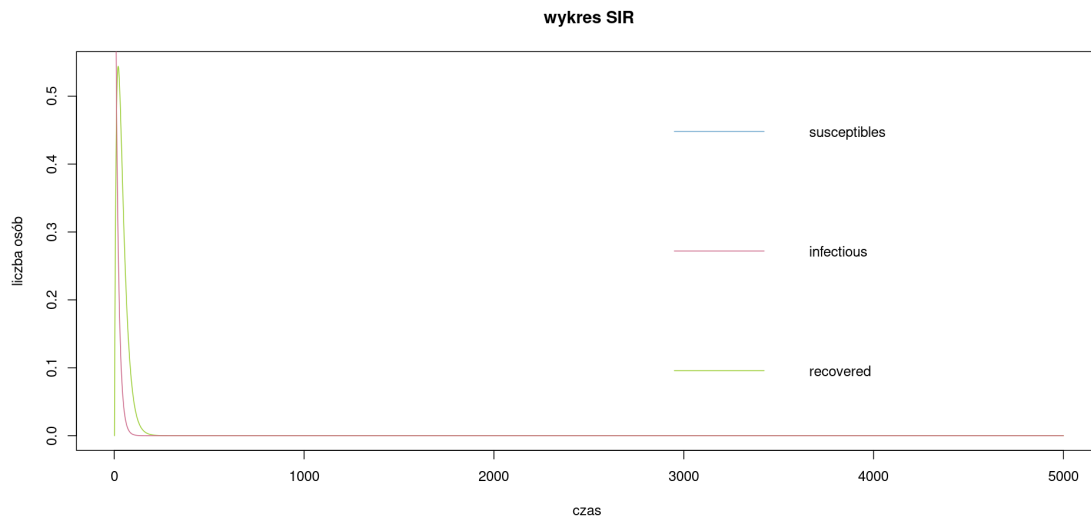
```
plot(c(1:length(R_a)), R_a, main = "wykres_R_w_czasie", type = "l", col = "yellowgreen",
     xlab = "czas_", ylab = "liczba_osob")
legend("right", c("recovered"), col = c("yellowgreen"), lty = 1, bty = "n")
```



Rysunek 3: Wykres **R** w czasie, przy parametrach a) $(S_0, I_0, \beta, \gamma, \mu) = (N - 1, 1, 0.5, 0.8, 0.4)$

Wykreślając na wspólnym wykresie S, I, R otrzymujemy następujący wykres (linia reprezentująca grupę S jest poza skalą, jej zmiana mocno zaburza widoczność, dlatego też nie zmieniamy jej).

```
plot(c(1:length(R_a)), R_a, main = "wykres_SIR", type = "l", col = "yellowgreen",
     xlab = "czas_", ylab = "liczba_osob")
lines(c(1:length(I_a)), I_a, type = "l", col = "palevioletred3")
lines(c(1:length(S_a)), S_a, type = "l", col = "skyblue3")
legend("right", c("susceptibles", "infectious", "recovered"), col = c("skyblue3",
                              "palevioletred3", "yellowgreen"), lty = 1, bty = "n")
```

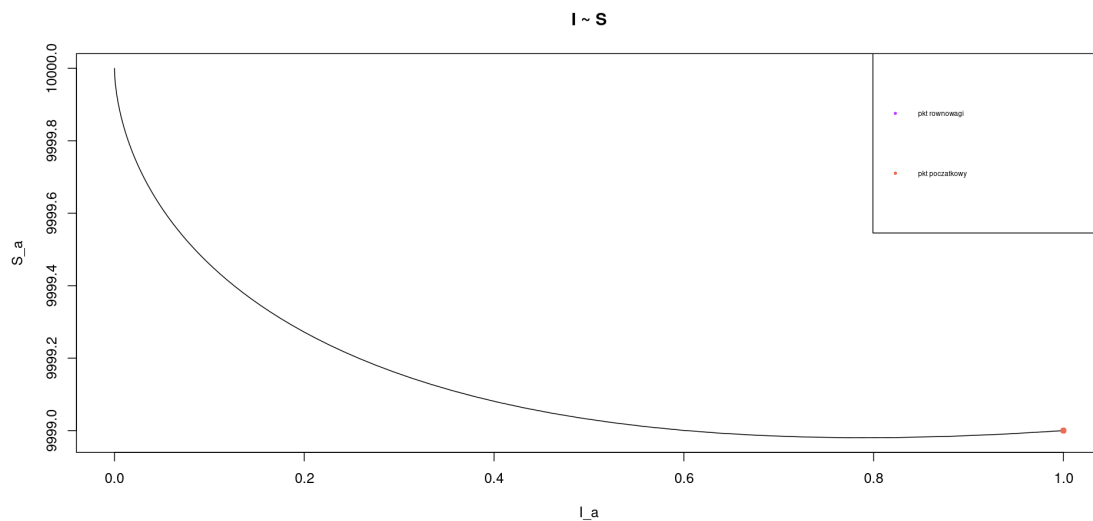


Rysunek 4: Wykres S , I , R w czasie, przy parametrach a) $(S_0, I_0, \beta, \gamma, \mu) = (N - 1, 1, 0.5, 0.8, 0.4)$

W N I O S E K: Wykres S bardzo szybko osiąga maksymalną pojemność populacji, jaką jest 10 000, dzieje się to już w 223 iteracji, czyli po nieco ponad 22 dniach. Podobnie z grupą I oraz R , one równie szybko osiągają wartość 0 i utrzymują się na tym poziomie, co oznacza, że nie pojawiają się żadne nowe przypadki osobników zainfekowanych, ani tych, którzy ozdrowieli. Czyli tak jak sugerowała nam wartość współczynnika R_0 - epidemia szybko wygasła, nie rozprzestrzeniając się przy tym.

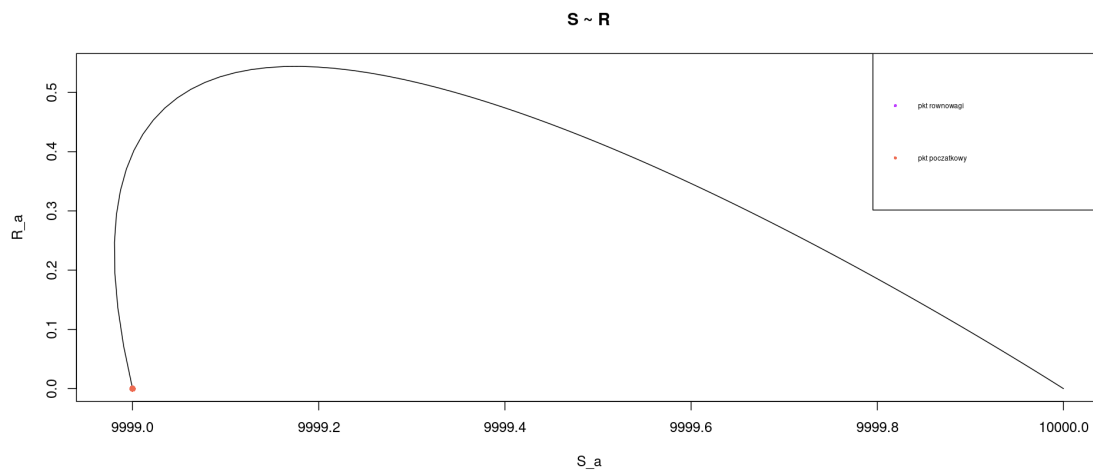
- Przechodzimy do wyznaczenia portretów fazowych. Wyznaczymy portrety fazowe dla par $I \sim S$, $S \sim R$, $I \sim R$ wraz z zaznaczonymi punktami początkowymi oraz punktami równowagi.

```
plot(I_a, S_a, main = "I~S", type = "l")
points(rownowagaSIR_a[2], rownowagaSIR_a[1], pch = 16, col = "darkorchid1")
points(I_a[1], y = S_a[1], pch = 16, col = "coral2")
legend("topright", c("pkt_rownowagi", "pkt_poczatkowy"),
      col = c("darkorchid1", "coral2"), cex=0.5, pch=c(16,16))
```



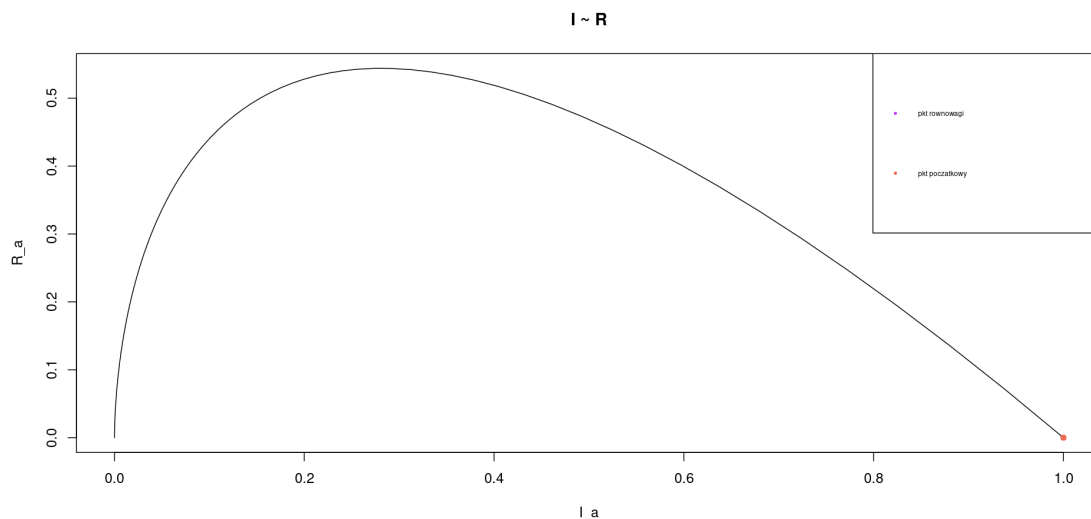
Rysunek 5: Portret fazowy $I \sim S$ przy parametrach a) $(S_0, I_0, \beta, \gamma, \mu) = (N - 1, 1, 0.5, 0.8, 0.4)$

```
plot(S_a, R_a, main = "_S_~_R_", type = "l")
points(rownowagaSIR_a[1], rownowagaSIR_a[3], pch = 16, col = "darkorchid1")
points(S_a[1], y = R_a[1], pch = 16, col = "coral2")
legend("topright", c("pkt_rownowagi", "pkt_poczatkowy"),
      col = c("darkorchid1", "coral2"), cex=0.5, pch=c(16,16))
```



Rysunek 6: Portret fazowy $S \sim R$ przy parametrach a) $(S_0, I_0, \beta, \gamma, \mu) = (N - 1, 1, 0.5, 0.8, 0.4)$

```
plot(I_a, R_a, main = "_I_~_R_", type = "l")
points(rownowagaSIR_a[2], rownowagaSIR_a[3], pch = 16, col = "darkorchid1")
points(I_a[1], y = R_a[1], pch = 16, col = "coral2")
legend("topright", c("pkt_rownowagi", "pkt_poczatkowy"),
      col = c("darkorchid1", "coral2"), cex=0.5, pch=c(16,16))
```



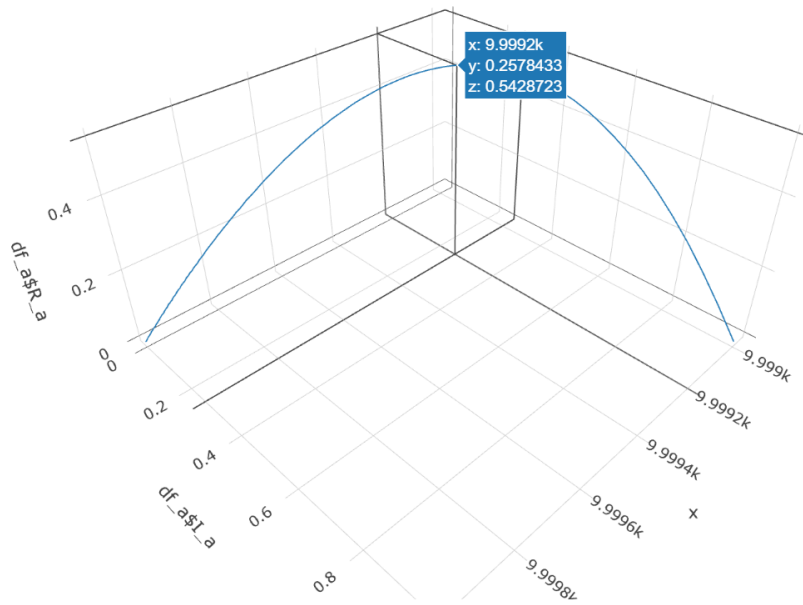
Rysunek 7: Portret fazowy $I \sim R$ przy parametrach a) $(S_0, I_0, \beta, \gamma, \mu) = (N - 1, 1, 0.5, 0.8, 0.4)$

W N I O S E K: Dla parametrów z tego podpunktu $(\beta, \gamma, \mu) = (0.5, 0.8, 0.4)$, gdzie $N = 10000$ na portretach fazowych obserwujemy, że kończą się one punktem $(S^*, I^*, R^*) = (10000, 0, 0)$, czyli punktem wygaśnięcia epidemii. Drugi z punktów równowagi (który obliczyliśmy wcześniej) nie został osiągnięty, nie wyświetla się na wykresach, gdyż znajduje się poza skalą.

Podsumowując, punkt równowagi $(S^*, I^*, R^*) = (N, 0, 0) = (10000, 0, 0)$ jest stabilnym stanem wolnym od choroby.

- Wyznamy trójwymiarowy portret fazowy $S \sim I \sim R$ korzystając z funkcji `plot_ly` pochodzącej z pakietu `plotly`.

```
library("plotly")
plot_ly(df_a, x=df_a$S_a, y=df_a$I_a, z=df_a$R_a,
main = "S_a I_a R_a", mode = 'lines', opacity = 1)
```



Rysunek 8: Portret fazowy $S \sim I \sim R$ przy parametrach a) $(S_0, I_0, \beta, \gamma, \mu) = (N - 1, 1, 0.5, 0.8, 0.4)$

W N I O S E K: Już na podstawie dwuwymiarowych portretów fazowych domyślaliśmy się jak wyglądać będzie trójwymiarowy portret fazowy. Widzimy, że wykres rozpoczyna swój bieg od momentu, gdy zainfekowany jest jeden osobnik, reszta populacji jest podatna na zarażenie, a kończy na momencie wolnym od epidemii dla $S = N = 10000$. Współrzędne I oraz R nigdy nie osiągają wartości większej od 1.

2.2.b)

Rozważany zestaw parametrów:

b) $(S_0, I_0, \beta, \gamma, \mu) = (N - 1, 1, 0.2, 0.04, 0.01)$.

- Na początku wyznaczamy wartość współczynnika R_0 korzystając ze wcześniej zaimplementowanej funkcji.

$R0_b = R_0(b = 0.2, g = 0.04, m = 0.01)$

Otrzymaliśmy wynik:

$$R0_b = 4 > 1$$

Następnie wyznaczyliśmy punkt równowagi dla parametrów z tego przykładu.

```
rownowagaSIR_b <- c(pkt_S(N = 10000, R0_b),
                    pkt_I(N = 10000, b = 0.2, m = 0.01, R0_b),
                    pkt_R(N = 10000, b = 0.2, g = 0.04, R0_b))
```

Punkt równowagi ma następujące współrzędne.

$$\begin{cases} S_b &= 2500 \\ I_b &= 1500 \\ R_b &= 6000 \end{cases}$$

W N I O S E K: $R_0 \cdot b$ ma wartość większą od jedynki, co sugeruje endemiczność. Choroba rozprzestrzenia się w populacji na normalnym lub oczekiwanym poziomie. Pandemia zaczyna zmieniać się w endemiczną, gdy choroba staje się bardziej stabilna i łatwiejsza do opanowania. Czyli dla zestawu parametrów z tego przykładu epidemia wybuchnie i będzie się rozprzestrzeniała aż do osiągnięcia punktu równowagi. Współrzędne obliczonego punktu równowagi, w odróżnieniu od podpunktu a), nie budzą naszych podejrzeń. W dalszej części sprawdzimy, jak model zachowuje się w obszarze sąsiadującym z punktem $(S^*, I^*, R^*) = (2500, 1500, 6000)$ - określimy jego stabilność.

- Przeprowadzamy symulację.

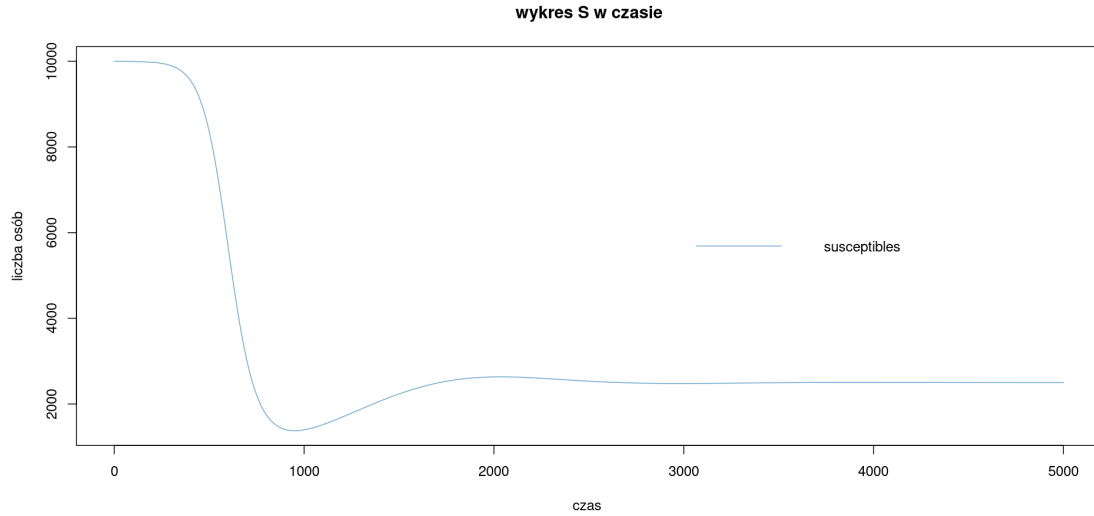
```
x_b = simulation_mickens(9999, 1, 0, h = 0.1, b = 0.2, g = 0.04, m = 0.01, TT = 1000, N = 10000)
S_b = x_b[[1]]
I_b = x_b[[2]]
R_b = x_b[[3]]
df_b = data.frame(S_b, I_b, R_b)
```

Przedstawimy ilość osób należącą do poszczególnych grup S, I, R dla pierwszych dziesięciu iteracji.

	S_b	I_b	R_b
1	9999.000	1.000000	0.000000000
2	9998.981	1.014923	0.004055638
3	9998.962	1.030069	0.008167747
4	9998.942	1.045441	0.012337176
5	9998.922	1.061043	0.016564782
6	9998.902	1.076877	0.020851438
7	9998.882	1.092947	0.025198029
8	9998.861	1.109257	0.029605453
9	9998.840	1.125811	0.034074622
10	9998.819	1.142611	0.038606461

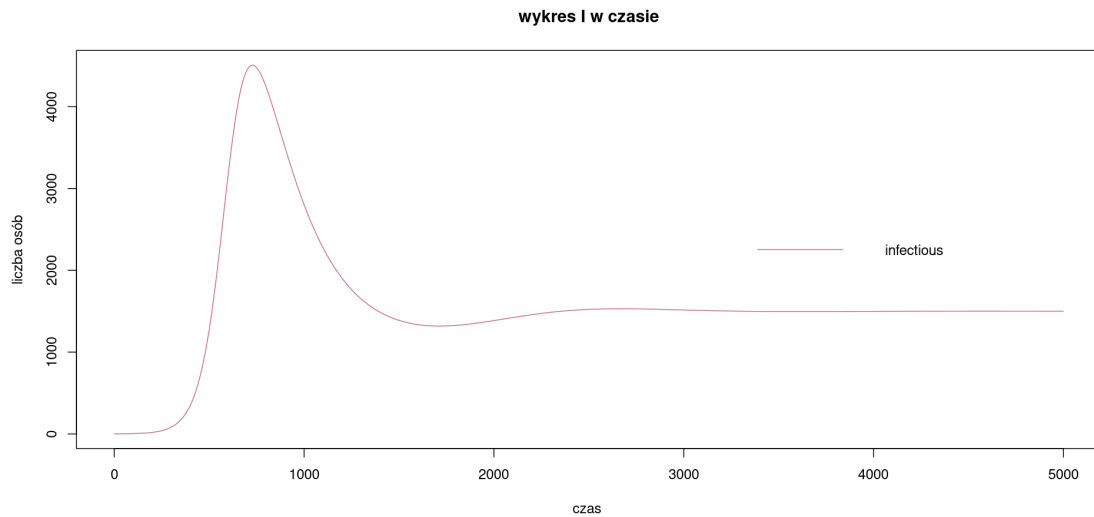
Następnie wyniki te przedstawimy na wykresach (trzech różnych, ze względu na duże różnice w skali).

```
plot(c(1:length(S_b)), S_b, main = "wykres_S_w_czasie", type = "l",
col = "skyblue3", xlab = "czas", ylab = "liczba_os_b")
legend("right", c("susceptibles"), col = c("skyblue3"), lty = 1, bty = "n")
```



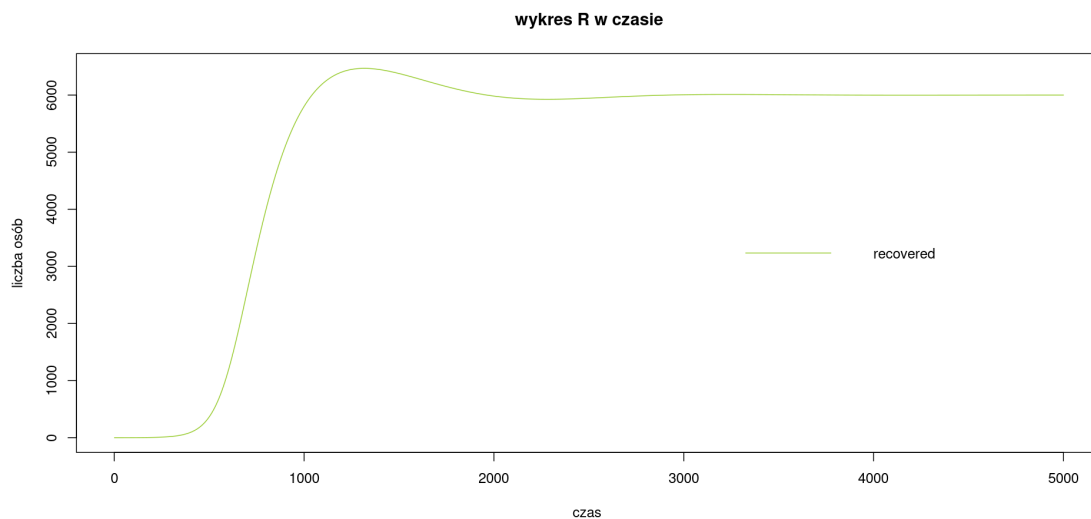
Rysunek 9: Wykres **S** w czasie, przy parametrach b) $(S_0, I_0, \beta, \gamma, \mu) = (N - 1, 1, 0.2, 0.04, 0.01)$

```
plot(c(1:length(I_b)), I_b, main = "wykres_I_w_czasie", type = "l",
col = "palevioletred3", xlab = "czas", ylab = "liczba_osob")
legend("right", c("infectious"), col = c("palevioletred3"), lty = 1, bty = "n")
```



Rysunek 10: Wykres **I** w czasie, przy parametrach b) $(S_0, I_0, \beta, \gamma, \mu) = (N - 1, 1, 0.2, 0.04, 0.01)$

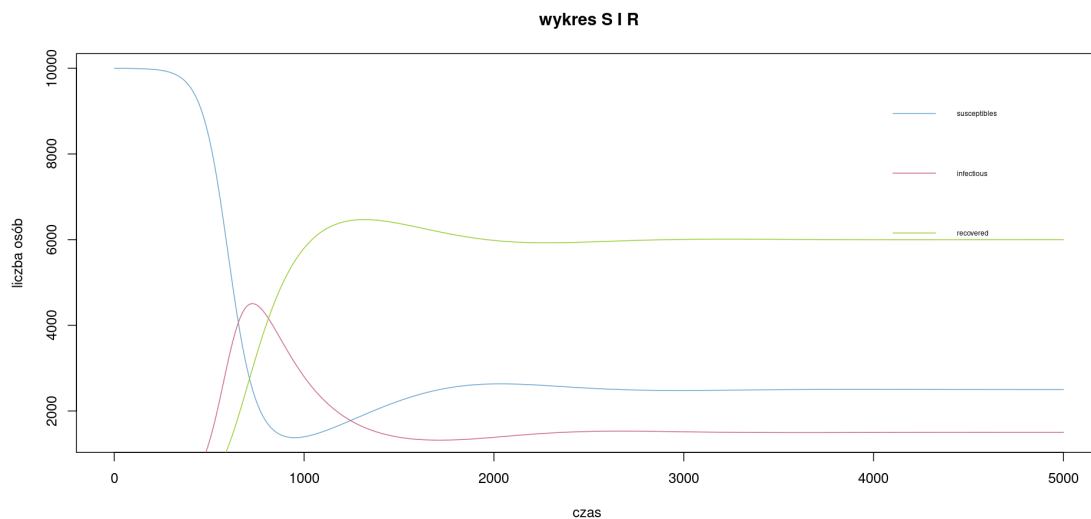
```
plot(c(1:length(R_b)), R_b, main = "wykres_R_w_czasie", type = "l",
col = "yellowgreen", xlab = "czas", ylab = "liczba_osob")
legend("right", c("recovered"), col = c("yellowgreen"), lty = 1, bty = "n")
```

Rysunek 11: Wykres **R** w czasie, przy parametrach b) $(S_0, I_0, \beta, \gamma, \mu) = (N - 1, 1, 0.2, 0.04, 0.01)$

Następnie wyniki te przedstawimy na wspólnym wykresie *SIR*.

```
plot(c(1:length(S_b)), S_b, main = "wykres S-I-R",
     type = "l", col = "skyblue3",
     xlab = "czas", ylab = "liczba_osob")
lines(c(1:length(I_b)), I_b, type = "l", col = "palevioletred3")
lines(c(1:length(R_b)), R_b, type = "l", col = "yellowgreen")
legend("topright", c("susceptibles", "infectious", "recovered"), col = c("skyblue3",
"palevioletred3", "yellowgreen"), lty = 1, bty = "n", cex=0.5)
```



Rysunek 12: Wykres **S, I, R** w czasie, przy parametrach b) $(S_0, I_0, \beta, \gamma, \mu) = (N - 1, 1, 0.2, 0.04, 0.01)$

WNIOSK: Wykres *S* już w okolicach 1000 iteracji, czyli 100 dniu, zmienia swoją wartość z 10 000 osób na poniżej 2 000 osób, czyli około 80% naszej populacji zmieniło grupę z grupy podatnej na zachorowanie, na jedną z grup: zainfekowanych lub ozdrowiałych. Po 2000 iteracji, czyli po 200 dniu sytuacja w grupie *S* się stabilizuje i

pozostaje na poziomie około 2 000 osób.

Krzywa na wykresie I szybko rośnie, od 0 do ponad 4 000 osobników na przestrzeni mniej niż 1000 iteracji, czyli mniej niż 100 dni. Jednakże podobnie jak grupa S - stabilizuje się już w okolicach dnia 200, na poziomie około 1 500 osób.

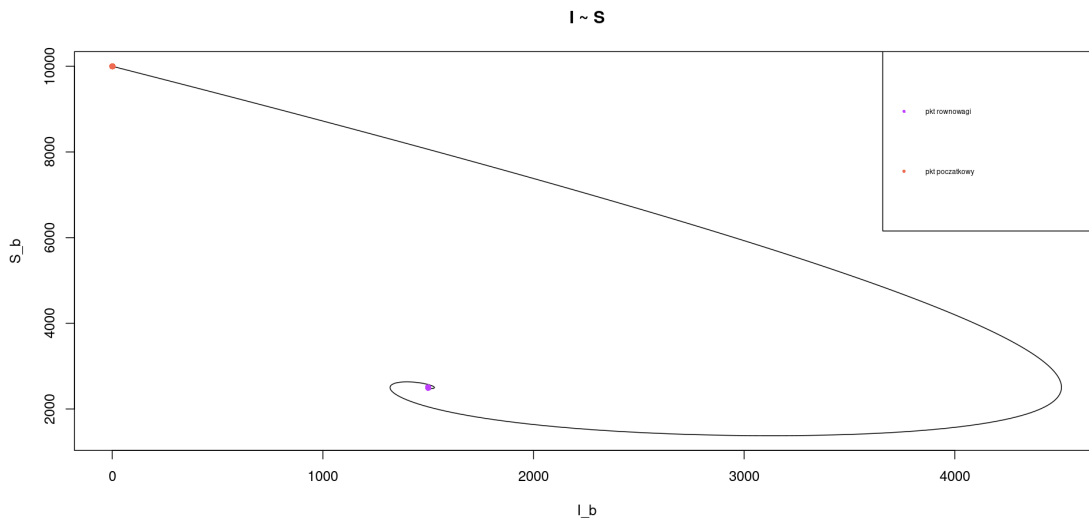
Jeśli chodzi o wykres R , to on podobnie jak I , szybko rośnie i osiąga swoje maksimum w 1000 iteracji, następnie w okolicach 2000 stabilizuje się na poziomie około 6 000 osób.

Czyli tak jak sugerowała nam wartość współczynnika R_0 - epidemia szybko się rozprzestrzeniła, stabilizując się na równych poziomach już w okolicach 2 000 iteracji.

Oznacza to, że taki stan epidemii się utrzymuje.

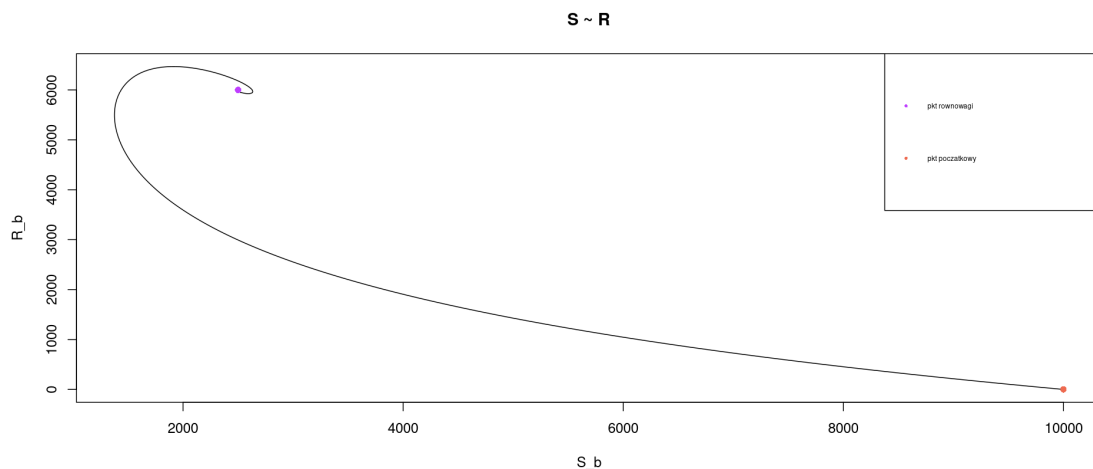
- Przechodzimy do wyznaczenia portretów fazowych. Wyznamy portrety fazowe dla par $I \sim S$, $S \sim R$, $I \sim R$ wraz z zaznaczonymi punktami początkowymi oraz punktami równowagi.

```
plot(I_b, S_b, main = "I~S", type = "l")
points(rownowagaSIR_b[2], rownowagaSIR_b[1], pch = 16, col = "darkorchid1")
points(I_b[1], y = S_b[1], pch = 16, col = "coral2")
legend("topright", c("pkt_rownowagi", "pkt_poczatkowy"),
      col = c("darkorchid1", "coral2"), cex=0.5, pch=c(16,16))
```



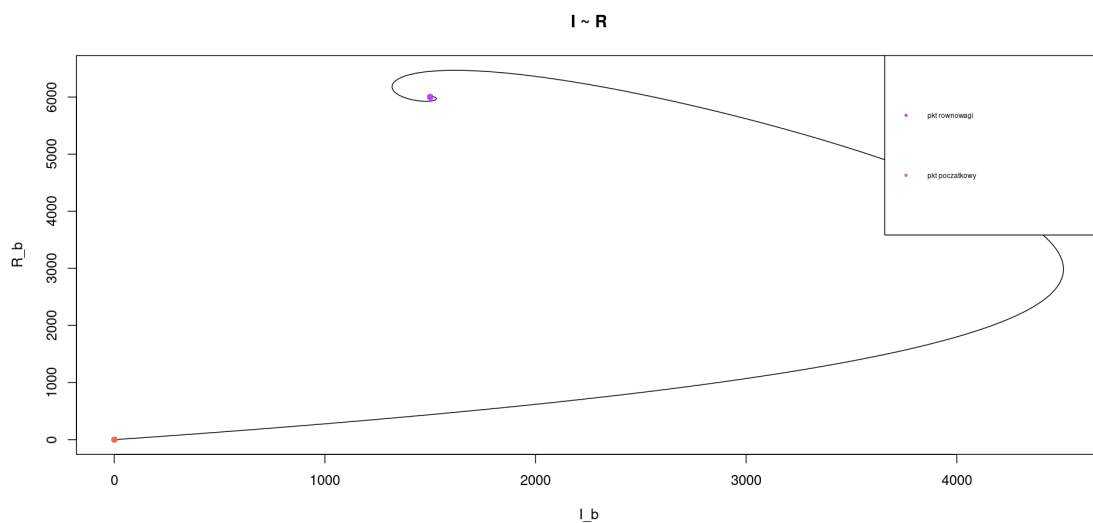
Rysunek 13: Portret fazowy $I \sim S$ przy parametrach b) $(S_0, I_0, \beta, \gamma, \mu) = (N - 1, 1, 0.2, 0.04, 0.01)$

```
plot(S_b, R_b, main = "S~R", type = "l")
points(rownowagaSIR_b[1], rownowagaSIR_b[3], pch = 16, col = "darkorchid1")
points(S_b[1], y = R_b[1], pch = 16, col = "coral2")
legend("topright", c("pkt_rownowagi", "pkt_poczatkowy"),
      col = c("darkorchid1", "coral2"), cex=0.5, pch=c(16,16))
```



Rysunek 14: Portret fazowy $S \sim R$ przy parametrach b) $(S_0, I_0, \beta, \gamma, \mu) = (N - 1, 1, 0.2, 0.04, 0.01)$

```
plot(I_b, R_b, main = "I ~ R", type = "l")
points(rownowagaSIR_b[2], rownowagaSIR_b[3], pch = 16, col = "darkorchid1")
points(I_b[1], y = R_b[1], pch = 16, col = "coral2")
legend("topright", c("pkt_rownowagi", "pkt_początkowy"),
      col = c("darkorchid1", "coral2"), cex=0.5, pch=c(16,16))
```

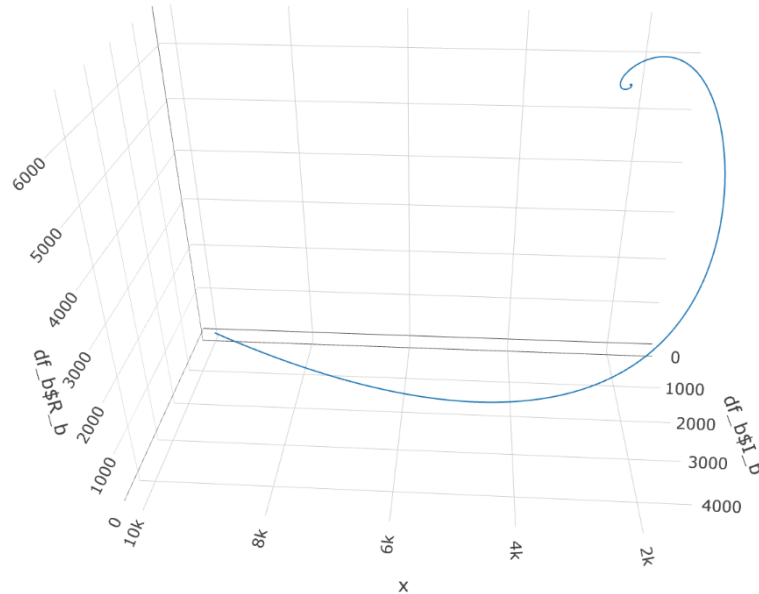


Rysunek 15: Portret fazowy $I \sim R$ przy parametrach b) $(S_0, I_0, \beta, \gamma, \mu) = (N - 1, 1, 0.2, 0.04, 0.01)$

W N I O S E K: Dla parametrów z tego podpunktu $(\beta, \gamma, \mu) = (0.2, 0.04, 0.01)$, gdzie $N = 10000$ na portretach fazowych obserwujemy zbieganie do punktu równowagi $(S^*, I^*, R^*) = (2500, 1500, 6000)$ - punkt ten jest więc stabilnym punktem endemicznym, gdyż wartość współczynnika R_0 jest większa od 1. Po osiągnięciu punktu równowagi, nasze wykresy S, I, R z poprzedniej części tego podpunktu, stabilizują się na równych poziomach.

- Wyznamy trójwymiarowy portret fazowy $S \sim I \sim R$ korzystając z funkcji `plot_ly` pochodzącej z pakietu `plotly`.

```
library("plotly")
plot_ly(df_b, x=df_b$S_b, y=df_b$I_b, z=df_b$R_b,
main = "S I R", mode = 'lines', opacity = 1)
```



Rysunek 16: Portret fazowy $S \sim I \sim R$ przy parametrach b) $(S_0, I_0, \beta, \gamma, \mu) = (N - 1, 1, 0.2, 0.04, 0.01)$

W N I O S E K: Już na podstawie dwuwymiarowych portretów fazowych domyślaliśmy się jak wyglądać będzie trójwymiarowy portret fazowy. Obserwujemy wyraźne zbieganie do punktu równowagi $(S^*, I^*, R^*) = (2500, 1500, 6000)$.

2.2.c)

Rozważany zestaw parametrów:

c) $(S_0, I_0, \beta, \gamma, \mu) = (N - 1, 1, 0.8, 0.4, 0.01)$.

- Na początku wyznaczamy wartość współczynnika R_0 korzystając ze wcześniej zaimplementowanej funkcji.

`R0_c = R_0(b = 0.8, g = 0.4, m = 0.01)`

Otrzymaliśmy wynik:

$$R_{0_c} \approx 1.95122 > 1$$

Następnie wyznaczaliśmy punkt równowagi dla parametrów z tego przykładu.

```
rownowagaSIR_c <- c(pkt_S(N = 10000, R0_c),
                    pkt_I(N = 10000, b = 0.8, m = 0.01, R0_c),
                    pkt_R(N = 10000, b = 0.8, g = 0.4, R0_c))
```

Punkt równowagi ma następujące współrzędne.

$$\begin{cases} S_c &= 5125.0000 \\ I_c &= 118.9024 \\ R_c &= 4756.0976 \end{cases}$$

W N I O S E K: R_{0_c} ma wartość większą od jedynki, co sugeruje endemiczność. Choroba rozprzestrzenia się w populacji na normalnym lub oczekiwanym poziomie. Pandemia zaczyna zmieniać się w endemiczną, gdy choroba staje się bardziej stabilna i łatwiejsza do opanowania. Czyli dla zestawu parametrów z tego przykładu epidemia wybuchnie i będzie się rozprzestrzeniała aż do osiągnięcia punktu równowagi. W dalszej części sprawdzimy, jak model zachowuje się w obszarze sąsiadującym z punktem $(S^*, I^*, R^*) = (5125, 118.9024, 4756.0976)$.

- Przeprowadzamy symulację.

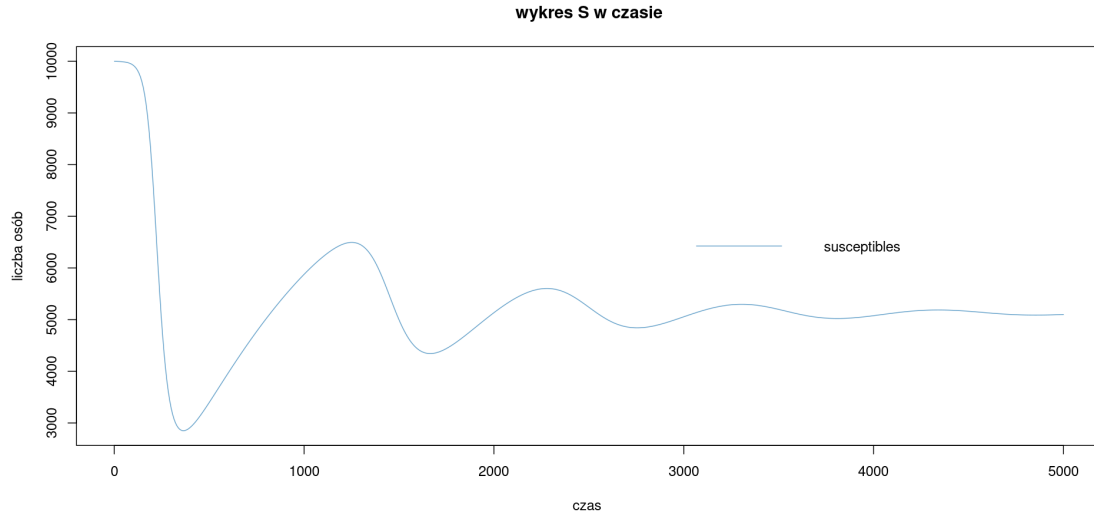
```
x_c = simulation_mickens(9999, 1, 0, h = 0.1, b = 0.8, g = 0.4, m = 0.01, TT = 1000, N = 10000)
S_c = x_c[[1]]
I_c = x_c[[2]]
R_c = x_c[[3]]
df_c = data.frame(S_c, I_c, R_c)
```

Przedstawimy ilość osób należącą do poszczególnych grup S, I, R dla pierwszych dziesięciu iteracji.

	S_c	I_c	R_c
1	9999.000	1.000000	0.0000000
2	9998.921	1.037456	0.04145677
3	9998.839	1.076314	0.08442489
4	9998.754	1.116626	0.12896098
5	9998.666	1.158448	0.17512379
6	9998.575	1.201836	0.22297423
7	9998.481	1.246847	0.27257554
8	9998.382	1.293543	0.32399328
9	9998.281	1.341988	0.37729549
10	9998.175	1.392245	0.43255274

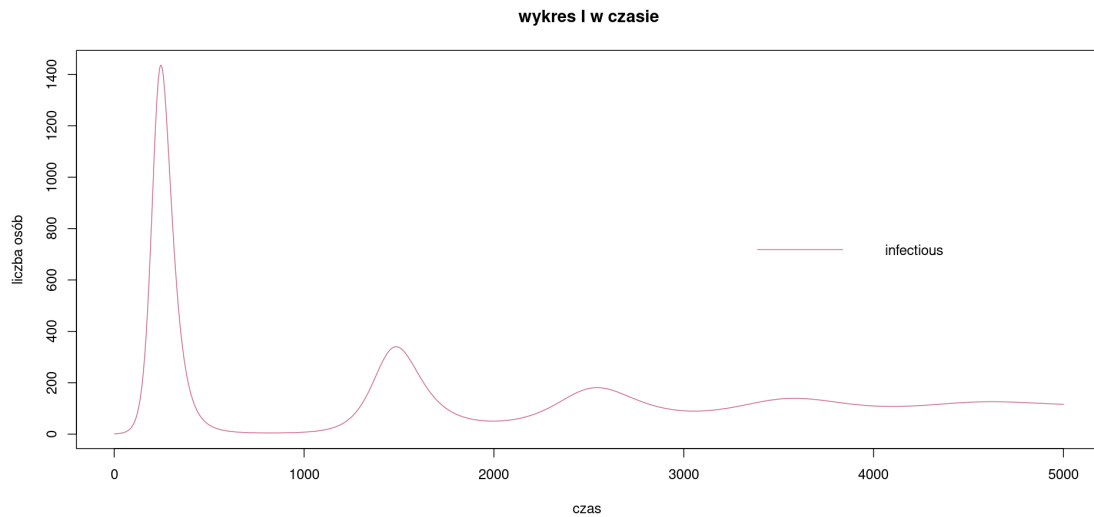
Następnie wyniki te przedstawimy na wykresach - trzech oddzielnych oraz jednym wspólnym.

```
plot(c(1:length(S_c)), S_c, main = "wykres_S_w_czasie", type = "l", col = "skyblue3",
     xlab = "czas", ylab = "liczba_osob")
legend("right", c("susceptibles"), col = c("skyblue3"), lty = 1, bty = "n")
```



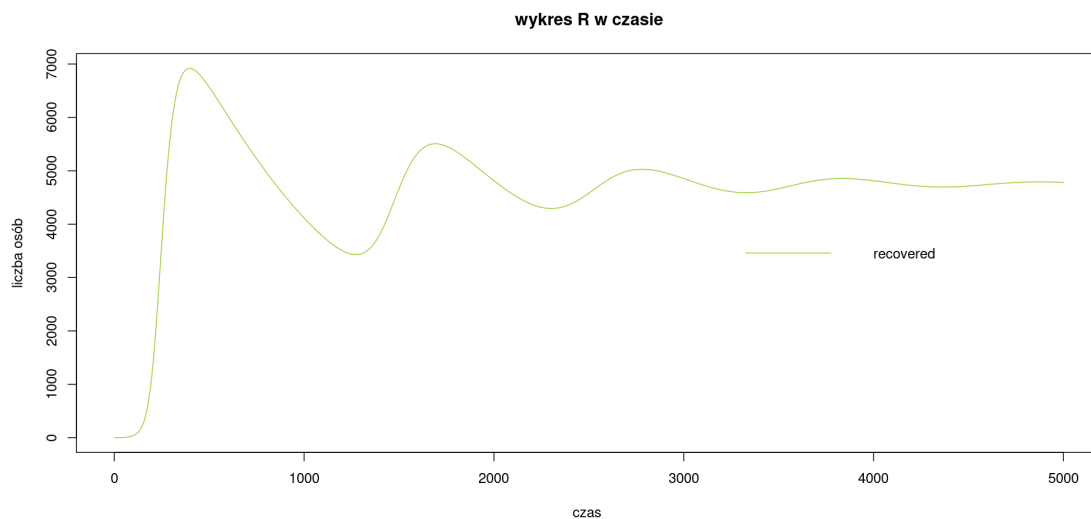
Rysunek 17: Wykres **S** w czasie, przy parametrach $c) (S_0, I_0, \beta, \gamma, \mu) = (N - 1, 1, 0.8, 0.4, 0.01)$

```
plot(c(1:length(I_c)), I_c, main = "wykres_I_w_czasie", type = "l", col = "palevioletred3",
     xlab = "czas", ylab = "liczba_osob")
legend("right", c("infectious"), col = c("palevioletred3"), lty = 1, bty = "n")
```



Rysunek 18: Wykres **I** w czasie, przy parametrach $c) (S_0, I_0, \beta, \gamma, \mu) = (N - 1, 1, 0.8, 0.4, 0.01)$

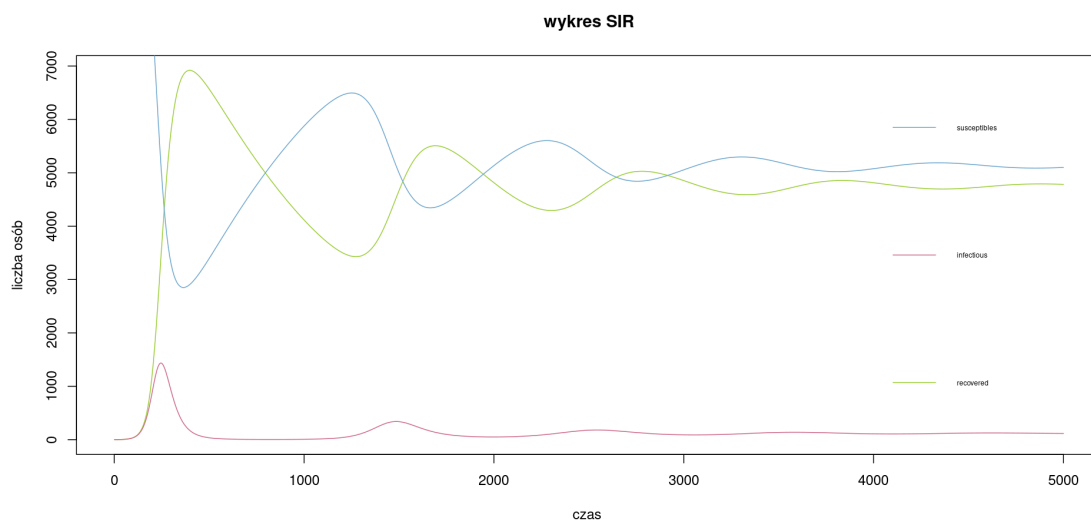
```
plot(c(1:length(R_c)), R_c, main = "wykres_R_w_czasie", type = "l", col = "yellowgreen",
     xlab = "czas", ylab = "liczba_osob")
legend("right", c("recovered"), col = c("yellowgreen"), lty = 1, bty = "n")
```



Rysunek 19: Wykres R w czasie, przy parametrach $c) (S_0, I_0, \beta, \gamma, \mu) = (N - 1, 1, 0.8, 0.4, 0.01)$

Następnie wyniki te przedstawimy na wspólnym wykresie.

```
plot(c(1:length(R_c)), R_c, main = "wykres_SIR", type = "l", col = "yellowgreen",
     xlab = "czas", ylab = "liczba_osob")
lines(c(1:length(I_c)), I_c, type = "l", col = "palevioletred3")
lines(c(1:length(S_c)), S_c, type = "l", col = "skyblue3")
legend("right", c("susceptibles", "infectious", "recovered"), col = c("skyblue3", "palevioletred3",
"yellowgreen"), lty = 1, bty = "n", cex=0.5)
```



Rysunek 20: Wykres S, I, R w czasie, przy parametrach $c) (S_0, I_0, \beta, \gamma, \mu) = (N - 1, 1, 0.8, 0.4, 0.01)$

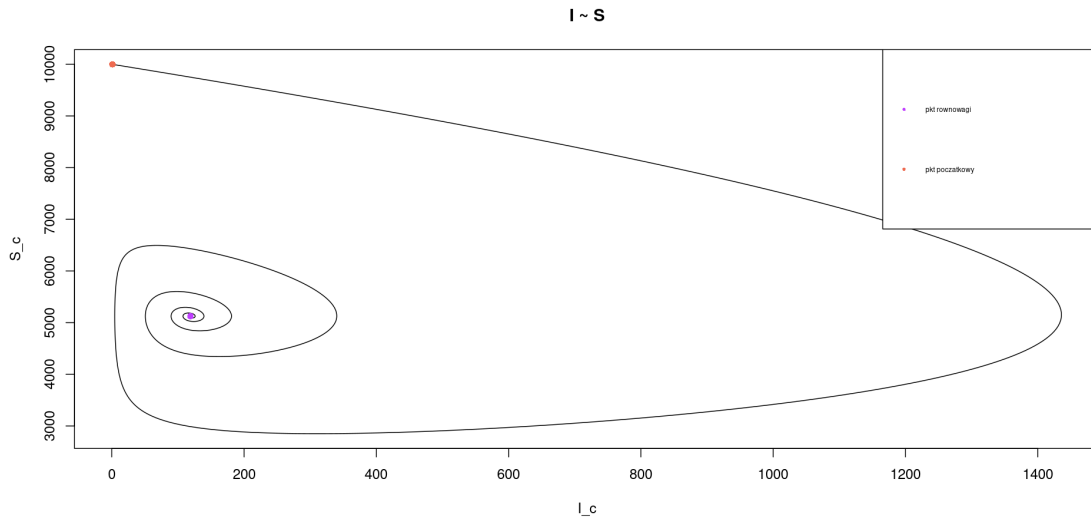
WNIOSK: Wykres S szybko zmienia swoją dynamikę, na początku już w okolicach 40 dnia osiąga swoje minimum - jakim jest wartość około 3 000 osób. Następnie rośnie i znowu maleje. Powoli stabilizuje się dopiero na poziomie około 5 000 osób po 400 dniu. Dynamika wykresów I oraz R jest bardzo podobna do dynamiki S . Krzywe po osiągnięciu swojego maksimum, naprzemiennie rosną oraz maleją, stabilizując się odpowiednio na

poziomach około 100 oraz 4 500 osób w ostatnich iteracjach.

Czyli tak jak sugerowała nam wartość współczynnika R_{0-c} - epidemia rozprzestrzeniła się, stabilizując się na równych poziomach w ostatniej fazie iteracji. Oznacza to, że epidemia na przestrzeni rozważanego przedziału czasowego jest bardzo zmienna, stabilizuje się dopiero w ostatnich dniach.

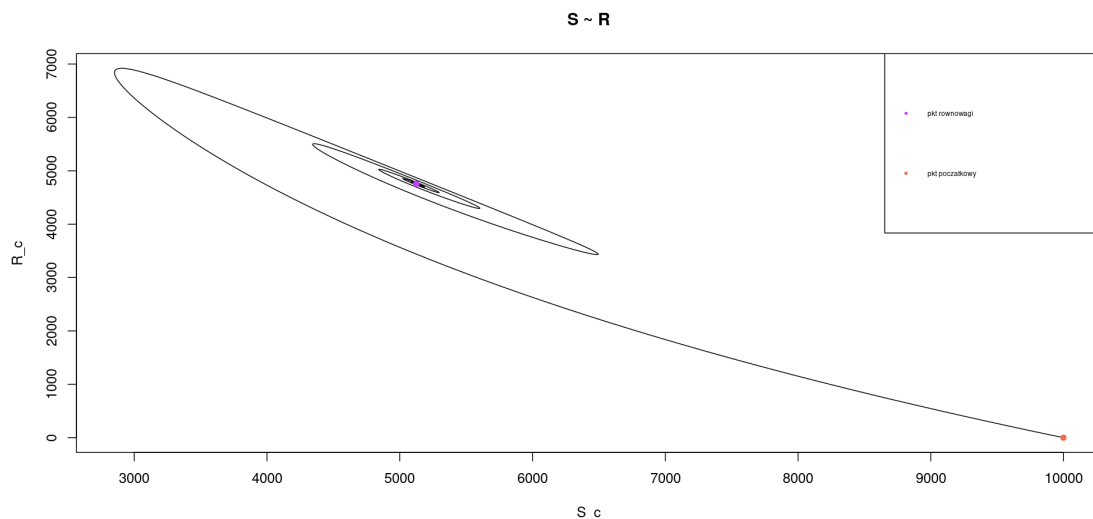
- Przechodzimy do wyznaczenia portretów fazowych. Wyznamy portrety fazowe dla par $I \sim S$, $S \sim R$, $I \sim R$ wraz z zaznaczonymi punktami początkowymi oraz punktami równowagi.

```
plot(I_c, S_c, main = "I~S", type = "l")
points(rownowagaSIR_c[2], rownowagaSIR_c[1], pch = 16, col = "darkorchid1")
points(I_c[1], y = S_c[1], pch = 16, col = "coral2")
legend("topright", c("pkt_rownowagi", "pkt_poczatkowy"),
      col = c("darkorchid1", "coral2"), cex=0.5, pch=c(16,16))
```



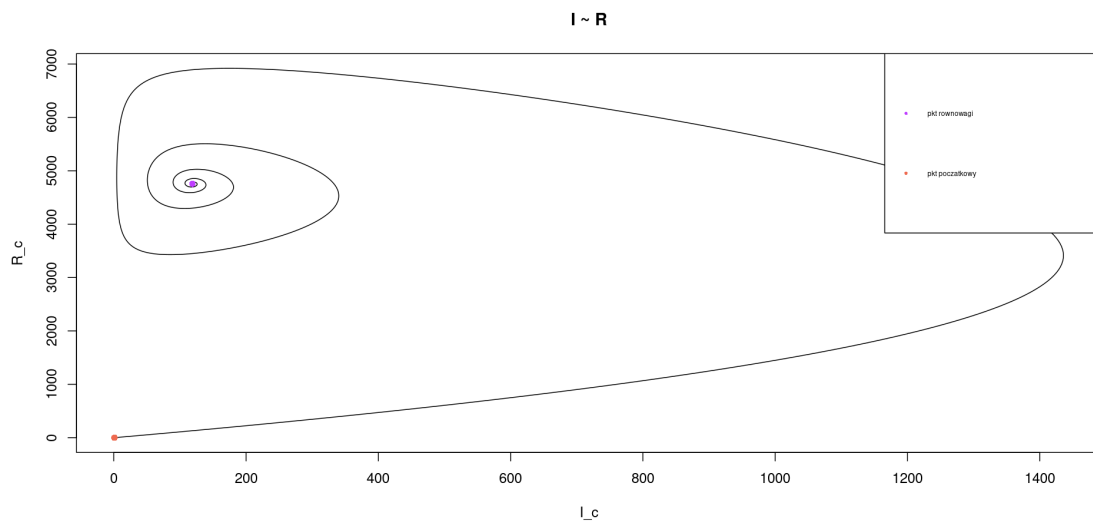
Rysunek 21: Portret fazowy $I \sim S$ przy parametrach c $(S_0, I_0, \beta, \gamma, \mu) = (N - 1, 1, 0.8, 0.4, 0.01)$

```
plot(S_c, R_c, main = "S~R", type = "l")
points(rownowagaSIR_c[1], rownowagaSIR_c[3], pch = 16, col = "darkorchid1")
points(S_c[1], y = R_c[1], pch = 16, col = "coral2")
legend("topright", c("pkt_rownowagi", "pkt_poczatkowy"),
      col = c("darkorchid1", "coral2"), cex=0.5, pch=c(16,16))
```

Rysunek 22: Portret fazowy $S \sim R$ przy parametrach c) $(S_0, I_0, \beta, \gamma, \mu) = (N - 1, 1, 0.8, 0.4, 0.01)$

```
plot(I_c, R_c, main = "I ~ R", type = "l")
points(rownowagaSIR_c[2], rownowagaSIR_c[3], pch = 16, col = "darkorchid1")
points(I_c[1], y = R_c[1], pch = 16, col = "coral2")
legend("topright", c("pkt rownowagi", "pkt początkowy"),
      col = c("darkorchid1", "coral2"), cex=0.5, pch=c(16,16))
```



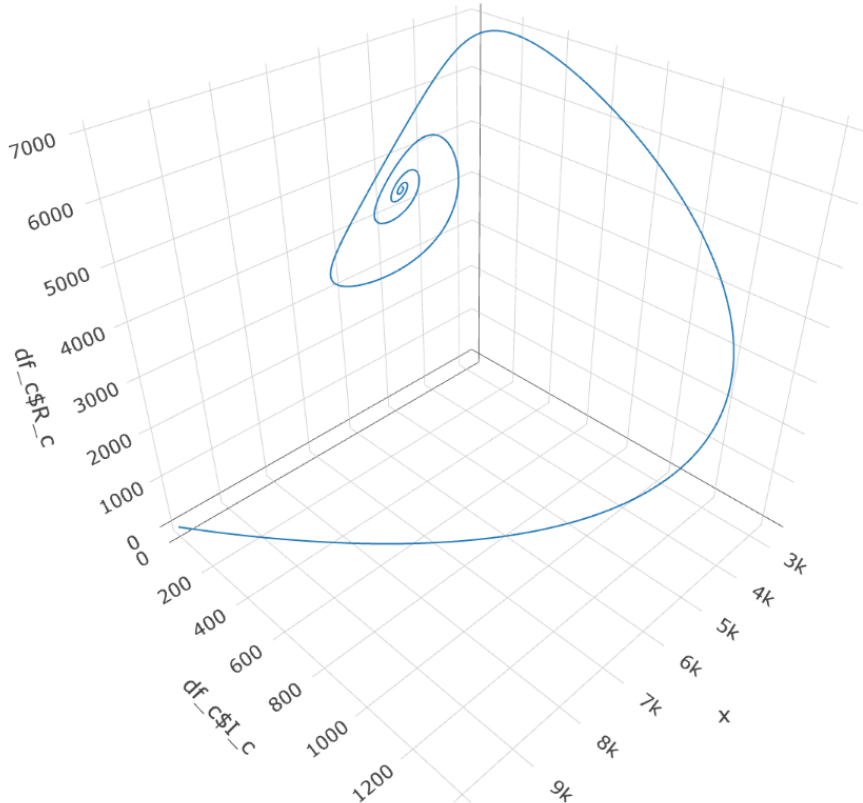
Rysunek 23: Portret fazowy $I \sim R$ przy parametrach c) $(S_0, I_0, \beta, \gamma, \mu) = (N - 1, 1, 0.8, 0.4, 0.01)$

W N I O S E K: Dla parametrów z tego podpunktu $(\beta, \gamma, \mu) = (0.8, 0.4, 0.01)$, gdzie $N = 10000$ na portretach fazowych obserwujemy zbieganie do punktu równowagi $(S^*, I^*, R^*) = (5125, 118.9024, 4756.0976)$ - punkt ten jest więc stabilnym punktem endemicznym, gdyż wartość współczynnika R_0 jest większa od 1. Po osiągnięciu punktu równowagi, nasze wykresy S, I, R z poprzedniej części tego podpunktu, stabilizują się na równych odpowiednim współrzędnym poziomach.

- Wyznamy trójwymiarowy portret fazowy $S \sim I \sim R$ korzystając z funkcji `plot_1y`

pochodzącej z pakietu `plotly`.

```
library("plotly")
plot_ly(df_c, x=df_c$S_c, y=df_c$I_c, z=df_c$R_c,
main = "S I R", mode = 'lines', opacity = 1)
```



Rysunek 24: Portret fazowy $S \sim I \sim R$ przy parametrach $c) (S_0, I_0, \beta, \gamma, \mu) = (N - 1, 1, 0.8, 0.4, 0.01)$

W N I O S E K: Już na podstawie dwuwymiarowych portretów fazowych domyślaliśmy się jak wyglądać będzie trójwymiarowy portret fazowy. Obserwujemy wyraźne zbieganie do punktu równowagi $(S^*, I^*, R^*) = (5125, 118.9024, 4756.0976)$.

2.2.d)

Rozważany zestaw parametrów:

d) $(S_0, I_0, \beta, \gamma, \mu) = (N - 1, 1, 0.8, 0.7, 0.01)$.

- Na początku wyznaczamy wartość współczynnika R_0 korzystając ze wcześniej zaimplementowanej funkcji.

```
R0_d = R_0(b = 0.8, g = 0.7, m = 0.01)
```

Otrzymaliśmy wynik:

$$R_{0_d} \approx 1.126761 > 1$$

Następnie wyznaczaliśmy punkt równowagi dla parametrów z tego przykładu.

```
rownowagaSIR_d <- c(pkt_S(N = 10000, R0_d),
```

```
pkt_I(N = 10000, b = 0.8, m = 0.01, R0_d),
pkt_R(N = 10000, b = 0.8, g = 0.7, R0_d))
```

Punkt równowagi ma następujące współrzędne.

$$\begin{cases} S_d &= 8875.00000 \\ I_d &= 15.84507 \\ R_d &= 1109.15493 \end{cases}$$

W N I O S E K: R_0_d ma wartość większą od jedynki, co sugeruje endemiczność. Choroba rozprzestrzenia się w populacji na normalnym lub oczekiwanym poziomie. Pandemia zaczyna zmieniać się w endemiczną, gdy choroba staje się bardziej stabilna i łatwiejsza do opanowania. Czyli dla zestawu parametrów z tego przykładu epidemia wybuchnie i będzie się rozprzestrzeniała aż do osiągnięcia punktu równowagi. W dalszej części sprawdzimy, jak model zachowuje się w obszarze sąsiadującym z punktem $(S^*, I^*, R^*) = (8875, 15.84507, 1109.15493)$.

- Przeprowadzamy symulację.

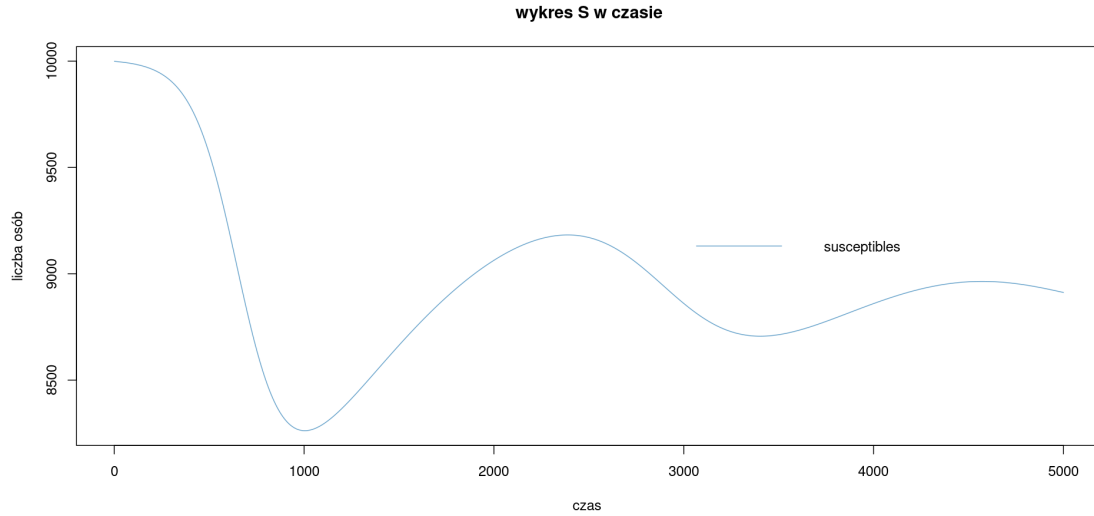
```
x_d = simulation_mickens(9999, 1, 0, h = 0.1, b = 0.8, g = 0.7, m = 0.01, TT = 1000, N = 10000)
S_d = x_d[[1]]
I_d = x_d[[2]]
R_d = x_d[[3]]
df_d = data.frame(S_d, I_d, R_d)
```

Przedstawimy ilość osób należącą do poszczególnych grup S, I, R dla pierwszych dziesięciu iteracji.

	S_d	I_d	R_d
1	9999.000	1.000000	0.00000000
2	9998.921	1.008395	0.07051715
3	9998.842	1.016860	0.14155583
4	9998.761	1.025396	0.21312044
5	9998.681	1.034003	0.28521542
6	9998.599	1.042681	0.35784525
7	9998.518	1.051432	0.43101445
8	9998.435	1.060255	0.50472756
9	9998.352	1.069151	0.57898917
10	9998.268	1.078122	0.65380391

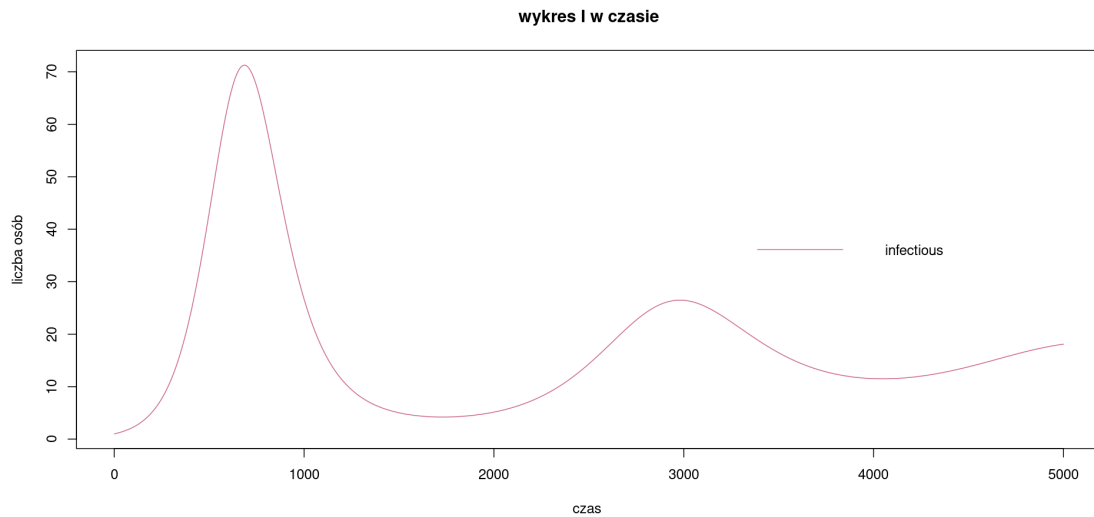
Następnie wyniki te przedstawimy na wykresach.

```
plot(c(1:length(S_d)), S_d, main = "wykres_S_w_czasie", type = "l", col = "skyblue3",
xlab = "czas", ylab = "liczba_os b")
legend("right", c("susceptibles"), col = c("skyblue3"), lty = 1, bty = "n")
```



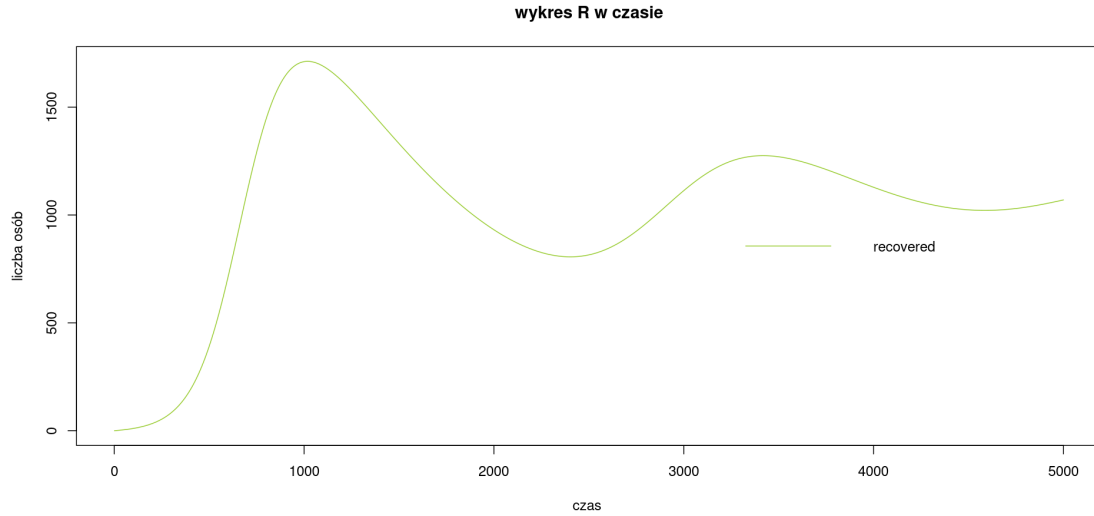
Rysunek 25: Wykres **S** w czasie, przy parametrach d) $(S_0, I_0, \beta, \gamma, \mu) = (N - 1, 1, 0.8, 0.7, 0.01)$

```
plot(c(1:length(I_d)), I_d, main = "wykres_I_w_czasie", type = "l", col = "palevioletred3",
     xlab = "czas", ylab = "liczba_osob")
legend("right", c("infectious"), col = c("palevioletred3"), lty = 1, bty = "n")
```



Rysunek 26: Wykres **I** w czasie, przy parametrach d) $(S_0, I_0, \beta, \gamma, \mu) = (N - 1, 1, 0.8, 0.7, 0.01)$

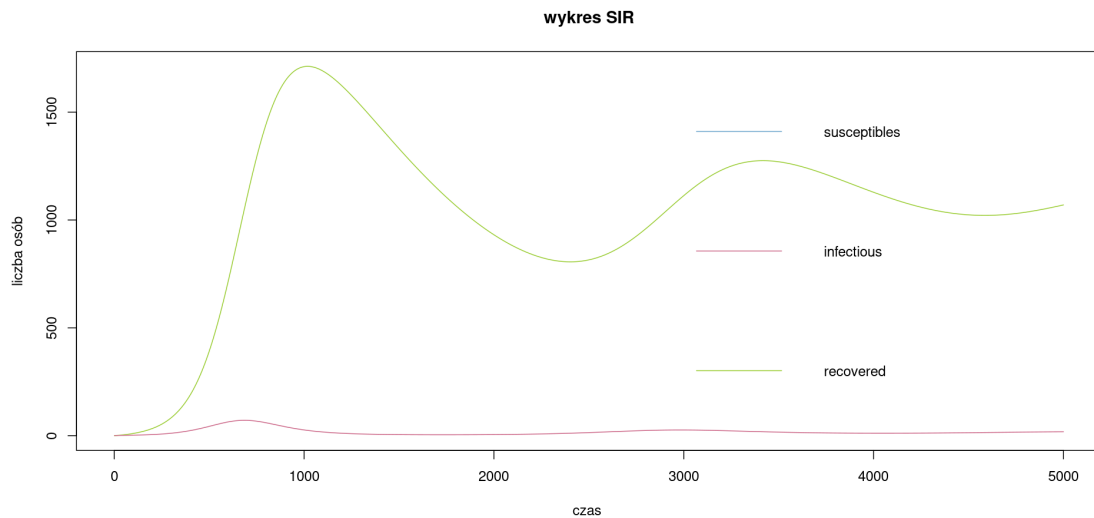
```
plot(c(1:length(R_d)), R_d, main = "wykres_R_w_czasie", type = "l", col = "yellowgreen",
     xlab = "czas", ylab = "liczba_osob")
legend("right", c("recovered"), col = c("yellowgreen"), lty = 1, bty = "n")
```



Rysunek 27: Wykres \mathbf{R} w czasie, przy parametrach d) $(S_0, I_0, \beta, \gamma, \mu) = (N - 1, 1, 0.8, 0.7, 0.01)$

Następnie wyniki te przedstawimy na wspólnym wykresie SIR .

```
plot(c(1:length(R_d)), R_d, main = "wykres_SIR", type = "l", col = "yellowgreen",
     xlab = "czas", ylab = "liczba_osob")
lines(c(1:length(I_d)), I_d, type = "l", col = "palevioletred3")
lines(c(1:length(S_d)), S_d, type = "l", col = "skyblue3")
legend("right", c("susceptibles", "infectious", "recovered"), col = c("skyblue3", "palevioletred3",
"yellowgreen"), lty = 1, bty = "n")
```



Rysunek 28: Wykres $\mathbf{S, I, R}$ w czasie, przy parametrach d) $(S_0, I_0, \beta, \gamma, \mu) = (N - 1, 1, 0.8, 0.7, 0.01)$

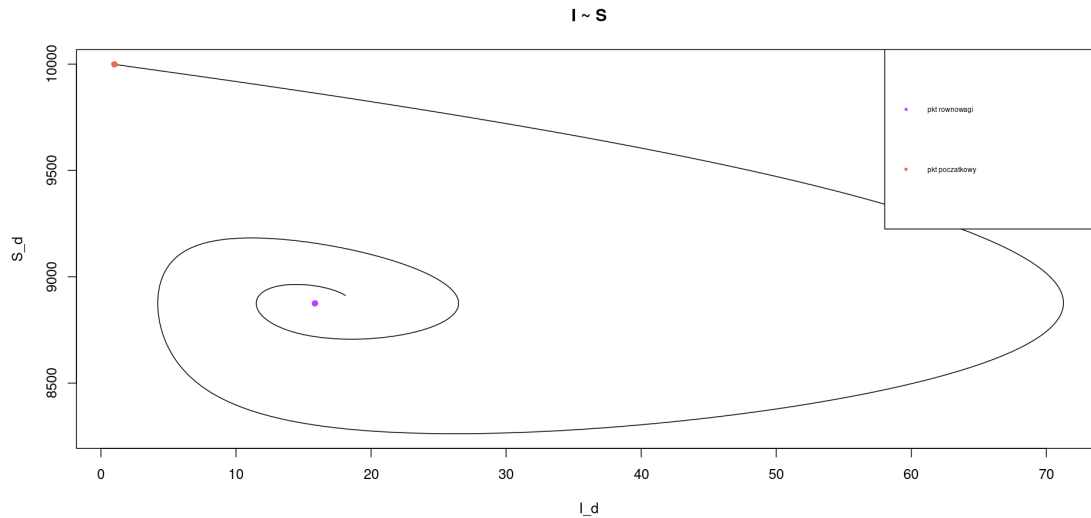
WNIOSK: Wykres S ma zmienną dynamikę, na początku maleje, by po osiągnięciu swojego minimum globalnego na rozważanym zbiorze argumentów. By następnie znowu zacząć rosnąć i maleć, na przemian. Nie obserwujemy wyraźnej stabilizacji, krzywa cały czas ma widoczne zakrzywienia. Dynamika wykresów I oraz R jest bardzo podobna do dynamiki S . Krzywe po osiągnięciu

swojego maksimum, naprzemiennie rosną oraz maleją. Ponownie, tak jak w przypadku wykresu S nie obserwujemy wyraźnej stabilizacji, krzywe cały czas mają widoczne zakrzywienia.

Czyli tak jak sugerowała nam wartość współczynnika R_0 - epidemia rozprzestrzeniła się, nie zaobserwowaliśmy jednak wyraźnej stabilizacji na przestrzeni wszystkich iteracji.

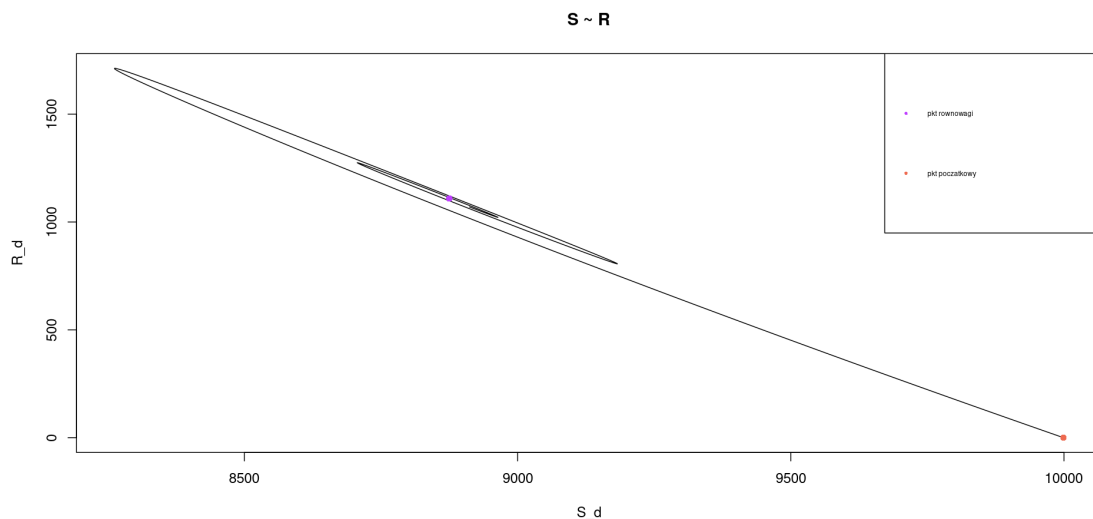
- Przechodzimy do wyznaczenia portretów fazowych. Wyznamy portrety fazowe dla par $I \sim S$, $S \sim R$, $I \sim R$ wraz z zaznaczonymi punktami początkowymi oraz punktami równowagi.

```
plot(I_d, S_d, main = "I~S", type = "l")
points(rownowagaSIR_d[2], rownowagaSIR_d[1], pch = 16, col = "darkorchid1")
points(I_d[1], y = S_d[1], pch = 16, col = "coral2")
legend("topright", c("pkt_rownowagi", "pkt_poczatkowy"),
      col = c("darkorchid1", "coral2"), cex=0.5, pch=c(16,16))
```



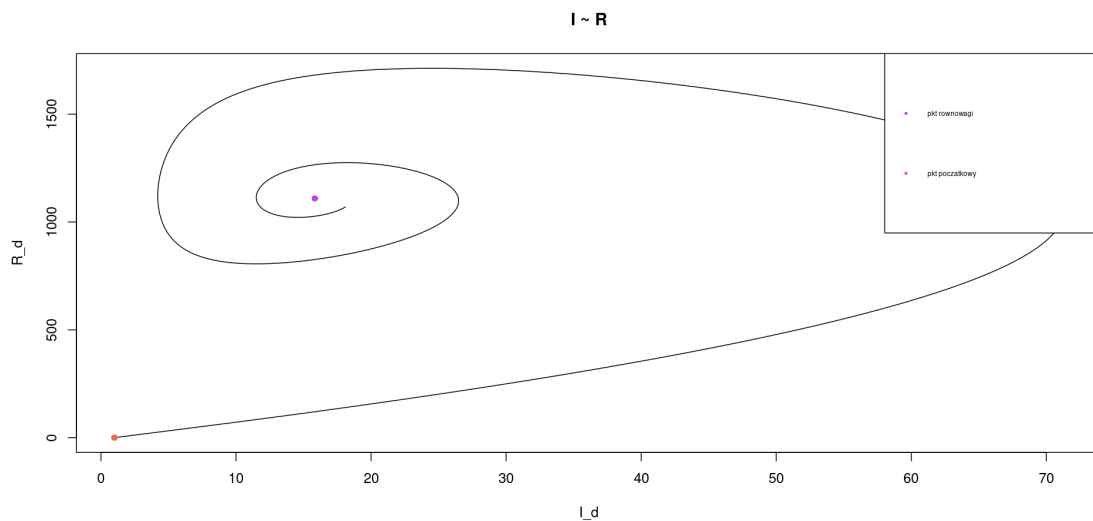
Rysunek 29: Portret fazowy $I \sim S$ przy parametrach d $(S_0, I_0, \beta, \gamma, \mu) = (N - 1, 1, 0.8, 0.7, 0.01)$

```
plot(S_d, R_d, main = "S~R", type = "l")
points(rownowagaSIR_d[1], rownowagaSIR_d[3], pch = 16, col = "darkorchid1")
points(S_d[1], y = R_d[1], pch = 16, col = "coral2")
legend("topright", c("pkt_rownowagi", "pkt_poczatkowy"),
      col = c("darkorchid1", "coral2"), cex=0.5, pch=c(16,16))
```



Rysunek 30: Portret fazowy $S \sim R$ przy parametrach d) $(S_0, I_0, \beta, \gamma, \mu) = (N - 1, 1, 0.8, 0.7, 0.01)$

```
plot(I_d, R_d, main = "I~R", type = "l")
points(rownowagaSIR_d[2], rownowagaSIR_d[3], pch = 16, col = "darkorchid1")
points(I_d[1], y = R_d[1], pch = 16, col = "coral2")
legend("topright", c("pkt rownowagi", "pkt początkowy"),
      col = c("darkorchid1", "coral2"), cex=0.5, pch=c(16,16))
```



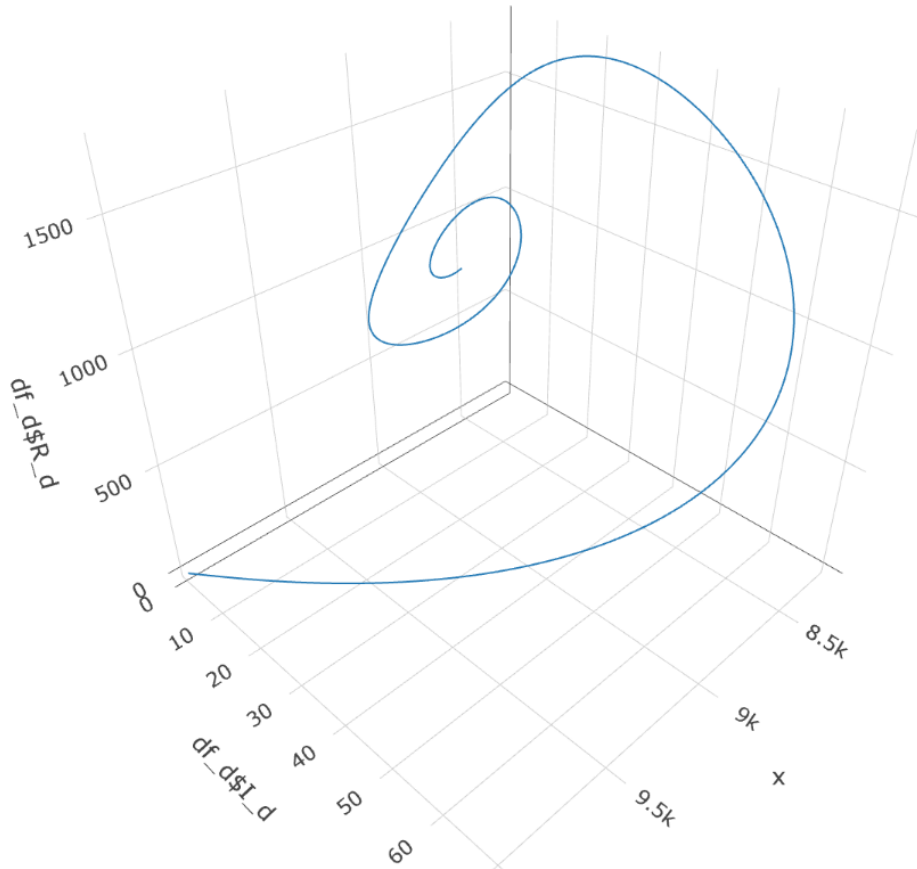
Rysunek 31: Portret fazowy $I \sim R$ przy parametrach d) $(S_0, I_0, \beta, \gamma, \mu) = (N - 1, 1, 0.8, 0.7, 0.01)$

WNIOSK: Dla parametrów z tego podpunktu $(\beta, \gamma, \mu) = (0.8, 0.7, 0.01)$, gdzie $N = 10000$ na portretach fazowych obserwujemy lekkie zbieganie, jednakże trzeba zaznaczyć, że krzywe nigdy nie dosięgają punktu równowagi $(S^*, I^*, R^*) = (8875, 15.84507, 1109.15493)$. Punkt ten jest zatem stabilnym punktem endemicznym. Tak jak zauważyliśmy w poprzedniej części tego podpunktu, nasze wykresy S, I, R nie stabilizują się wyraźnie na poziomach odpowiednio dla S i I oraz R : 8 875, 15.84507, 1 109.15493.

- Wyznamy trójwymiarowy portret fazowy $S \sim I \sim R$ korzystając z funkcji `plot_1y`

pochodzącej z pakietu `plotly`.

```
library("plotly")
plot_ly(df_d, x=df_d$S_d, y=df_d$I_d, z=df_d$R_d,
main = "S I R", mode = 'lines', opacity = 1)
```



Rysunek 32: Portret fazowy $S \sim I \sim R$ przy parametrach d) $(S_0, I_0, \beta, \gamma, \mu) = (N - 1, 1, 0.8, 0.7, 0.01)$

W N I O S E K: Już na podstawie dwuwymiarowych portretów fazowych domyślaliśmy się jak wyglądać będzie trójwymiarowy portret fazowy. Nie obserwujemy wyraźnego zbiegania do punktu równowagi $(S^*, I^*, R^*) = (8875, 15.84507, 1109.15493)$, więc punkt ten nie jest stabilny.

Zadanie 3.

Kolejno dla zestawów parametrów z Zadania 2.2.b), Zadania 2.2.c) oraz Zadania 2.2.d) wykonamy symulacje grupy osobników zakażonych I dla różnych warunków początkowych. Wyniki przedstawimy na wspólnym wykresie.

Zakładamy, że na początku epidemii nie mamy żadnych ozdrowieńców, więc manipulujemy jedynie warunkami początkowymi dla grupy osobników podatnych (S) oraz zainfekowanych (I), tak by zawsze sumowały się one do N - liczebności populacji.

Przechodzimy już do przykładów.

3.b)

Rozważane parametry:

b) $(\beta, \gamma, \mu) = (0.2, 0.04, 0.01)$

Różne warunki początkowe, dla których przeprowadzamy symulacje:

```
# S_0 = 5000
# I_0 = 5000
x_b3v1 = simulation_mickens(5000, 5000, 0, h = 0.1,
b = 0.2, g = 0.04, m = 0.01, TT = 1000, N = 10000)
S_b3v1 = x_b3v1[[1]]
I_b3v1 = x_b3v1[[2]]
R_b3v1 = x_b3v1[[3]]

# S_0 = 7000
# I_0 = 3000
x_b3v2 = simulation_mickens(7000, 3000, 0, h = 0.1,
b = 0.2, g = 0.04, m = 0.01, TT = 1000, N = 10000)
S_b3v2 = x_b3v2[[1]]
I_b3v2 = x_b3v2[[2]]
R_b3v1 = x_b3v2[[3]]

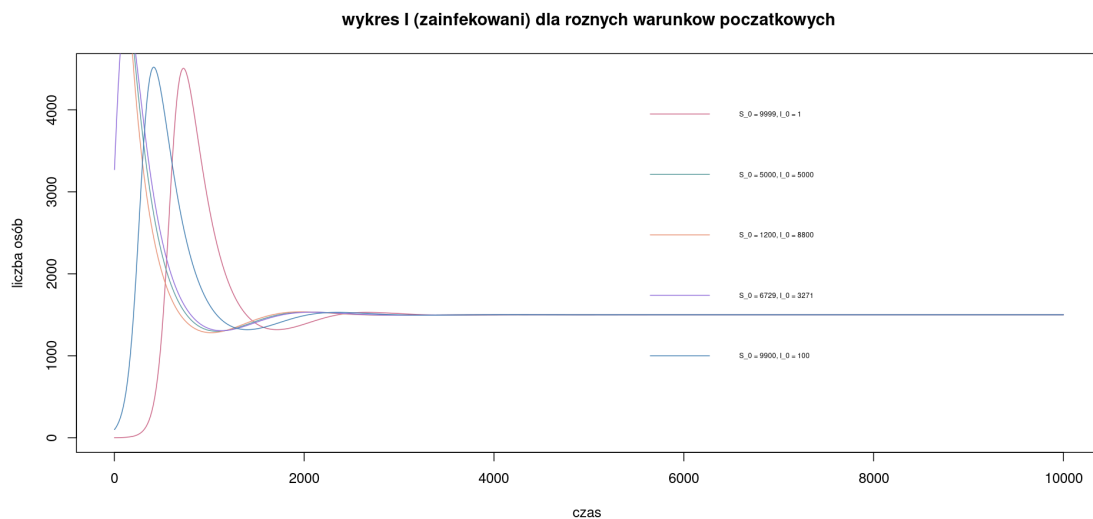
# S_0 = 1200
# I_0 = 8800
x_b3v3 = simulation_mickens(1200, 8800, 0, h = 0.1,
b = 0.2, g = 0.04, m = 0.01, TT = 1000, N = 10000)
S_b3v3 = x_b3v3[[1]]
I_b3v3 = x_b3v3[[2]]
R_b3v3 = x_b3v3[[3]]

# S_0 = 6729
# I_0 = 3271
x_b3v4 = simulation_mickens(6729, 3271, 0, h = 0.1,
b = 0.2, g = 0.04, m = 0.01, TT = 1000, N = 10000)
S_b3v4 = x_b3v4[[1]]
I_b3v4 = x_b3v4[[2]]
R_b3v4 = x_b3v4[[3]]

# S_0 = 9900
# I_0 = 100
x_b3v5 = simulation_mickens(9900, 100, 0, h = 0.1,
b = 0.2, g = 0.04, m = 0.01, TT = 1000, N = 10000)
S_b3v5 = x_b3v5[[1]]
I_b3v5 = x_b3v5[[2]]
R_b3v5 = x_b3v5[[3]]
```

Wyniki tych symulacji przedstawiamy na wspólnym wykresie.

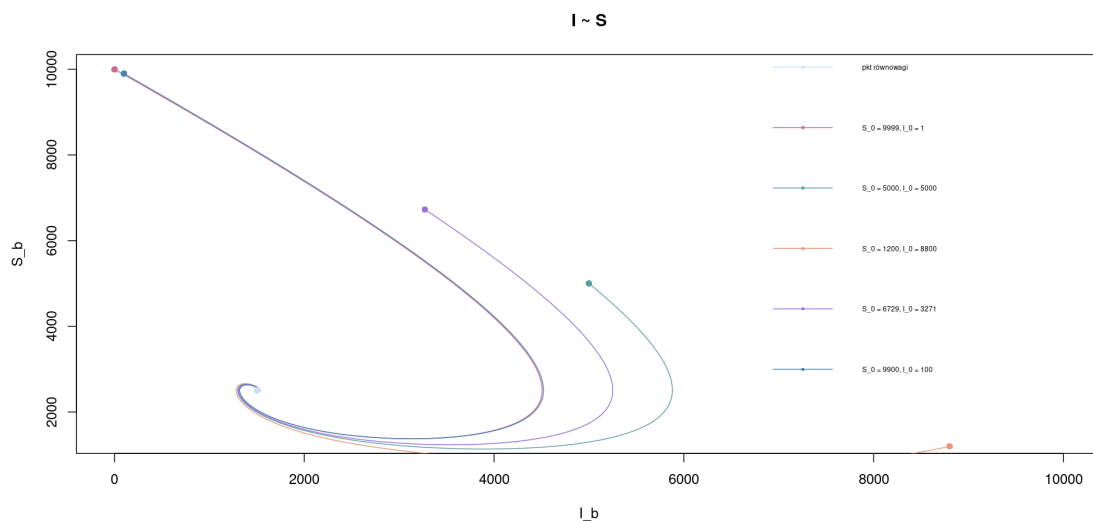
```
plot(c(1:length(I_b)), I_b, main = "wykres_I_(zainfekowani)_dla_roznych_warunkow_poczatkowych",
type = "l", col = "palevioletred3", xlab = "czas", ylab = "liczba_osob")
lines(c(1:length(I_b)), I_b3v1, col="cadetblue")
lines(c(1:length(I_b)), I_b3v3, col="darksalmon")
lines(c(1:length(I_b)), I_b3v4, col="mediumpurple")
lines(c(1:length(I_b)), I_b3v5, col="steelblue")
legend("topright", c("S_0=9999, I_0=1", "S_0=5000, I_0=5000", "S_0=1200, I_0=8800",
"S_0=6729, I_0=3271", "S_0=9900, I_0=100"), col = c("palevioletred3", "cadetblue",
"darksalmon", "mediumpurple", "steelblue" ), lty = 1, bty = "n", cex=0.5)
```



Rysunek 33: Wykres zainfekowanych przy parametrach b) $(\beta, \gamma, \mu) = (0.2, 0.04, 0.01)$ i dla różnych warunków początkowych

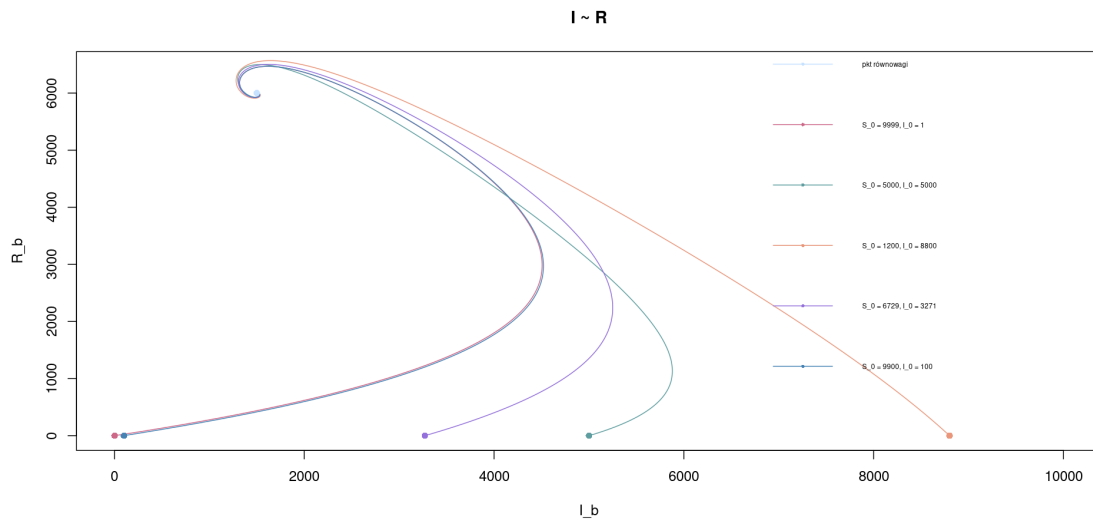
Wykreślmy także portrety fazowe $I \sim S$ oraz $I \sim R$. Rozpoczynając od $I \sim S$.

```
plot(I_b, S_b, main = "I ~ S", type = "l", col = "palevioletred3", xlim = c(0, 10000))
lines(I_b3v1, S_b3v1, col = "cadetblue")
lines(I_b3v3, S_b3v3, col = "darksalmon")
lines(I_b3v4, S_b3v4, col = "mediumpurple")
lines(I_b3v5, S_b3v5, col = "steelblue")
points(rownowagaSIR_b[2], rownowagaSIR_b[1], pch = 16, col = "slategray1")
points(I_b[1], y = S_b[1], pch = 16, col = "palevioletred3")
points(I_b3v1[1], y = S_b3v1[1], pch = 16, col = "cadetblue")
points(I_b3v3[1], y = S_b3v3[1], pch = 16, col = "darksalmon")
points(I_b3v4[1], y = S_b3v4[1], pch = 16, col = "mediumpurple")
points(I_b3v5[1], y = S_b3v5[1], pch = 16, col = "steelblue")
legend("topright", c("pkt rownowagi", "S_0 = 9999, I_0 = 1", "S_0 = 5000, I_0 = 5000",
"S_0 = 1200, I_0 = 8800", "S_0 = 6729, I_0 = 3271", "S_0 = 9900, I_0 = 100"),
col = c("slategray1", "palevioletred3", "cadetblue", "darksalmon", "mediumpurple", "steelblue"),
lty = 1, bty = "n", cex = 0.5, pch = c(16, 16), inset = -0.12)
```



Rysunek 34: Portret fazowy $I \sim S$ przy parametrach b $(\beta, \gamma, \mu) = (0.2, 0.04, 0.01)$ i dla różnych warunków początkowych

```
plot(I_b, R_b, main = "I ~ R", type = "l", col = "palevioletred3", xlim = c(0, 10000))
lines(I_b3v1, R_b3v1, col = "cadetblue")
lines(I_b3v3, R_b3v3, col = "darksalmon")
lines(I_b3v4, R_b3v4, col = "mediumpurple")
lines(I_b3v5, R_b3v5, col = "steelblue")
points(rownowagaSIR_b[2], rownowagaSIR_b[3], pch = 16, col = "slategray1")
points(I_b[1], y = R_b[1], pch = 16, col = "palevioletred3")
points(I_b3v1[1], y = R_b3v1[1], pch = 16, col = "cadetblue")
points(I_b3v3[1], y = R_b3v3[1], pch = 16, col = "darksalmon")
points(I_b3v4[1], y = R_b3v4[1], pch = 16, col = "mediumpurple")
points(I_b3v5[1], y = R_b3v5[1], pch = 16, col = "steelblue")
legend("topright", c("pkt rownowagi", "S_0 = 9999, I_0 = 1", "S_0 = 5000, I_0 = 5000",
"S_0 = 1200, I_0 = 8800", "S_0 = 6729, I_0 = 3271", "S_0 = 9900, I_0 = 100"),
col = c("slategray1", "palevioletred3", "cadetblue", "darksalmon", "mediumpurple", "steelblue"),
lty = 1, bty = "n", cex = 0.5, pch = c(16, 16), inset = -0.12)
```



Rysunek 35: Portret fazowy $I \sim R$ przy parametrach b $(\beta, \gamma, \mu) = (0.2, 0.04, 0.01)$ i dla różnych warunków początkowych

W N I O S E K: Różne warunki początkowe nie wpływają na interpretację modelu przy zadanych parametrach $(\beta, \gamma, \mu) = (0.2, 0.04, 0.01)$. Interpretujemy każdą z tych symulacji analogicznie do tej przeprowadzonej w Zadaniu 2.2.b).

3.c)

Rozważane parametry:

c) $(\beta, \gamma, \mu) = (0.8, 0.4, 0.01)$

Różne warunki początkowe, dla których przeprowadzamy symulacje:

```
# S_0 = 5000
# I_0 = 5000
x_c3v1 = simulation_mickens(5000, 5000, 0, h = 0.1,
b = 0.8, g = 0.4, m = 0.01, TT = 1000, N = 10000)
S_c3v1 = x_c3v1[[1]]
```

```

I_c3v1 = x_c3v1[[2]]
R_c3v1 = x_c3v1[[3]]

# S_0 = 7000
# I_0 = 3000
x_c3v2 = simulation_mickens(7000, 3000, 0, h = 0.1,
b = 0.8, g = 0.4, m = 0.01, TT = 1000, N = 10000)
S_c3v2 = x_c3v2[[1]]
I_c3v2 = x_c3v2[[2]]
R_c3v1 = x_c3v2[[3]]

```

```

# S_0 = 1200
# I_0 = 8800
x_c3v3 = simulation_mickens(1200,8800, 0, h = 0.1,
b = 0.8, g = 0.4, m = 0.01, TT = 1000, N = 10000)
S_c3v3 = x_c3v3[[1]]
I_c3v3 = x_c3v3[[2]]
R_c3v3 = x_c3v3[[3]]

```

```

# S_0 = 6729
# I_0 = 3271
x_c3v4 = simulation_mickens(6729, 3271, 0, h = 0.1,
b = 0.8, g = 0.4, m = 0.01, TT = 1000, N = 10000)
S_c3v4 = x_c3v4[[1]]
I_c3v4 = x_c3v4[[2]]
R_c3v4 = x_c3v4[[3]]

```

```

# S_0 = 9900
# I_0 = 100
x_c3v5 = simulation_mickens(9900, 100, 0, h = 0.1,
b = 0.8, g = 0.4, m = 0.01, TT = 1000, N = 10000)
S_c3v5 = x_c3v5[[1]]
I_c3v5 = x_c3v5[[2]]
R_c3v5 = x_c3v5[[3]]

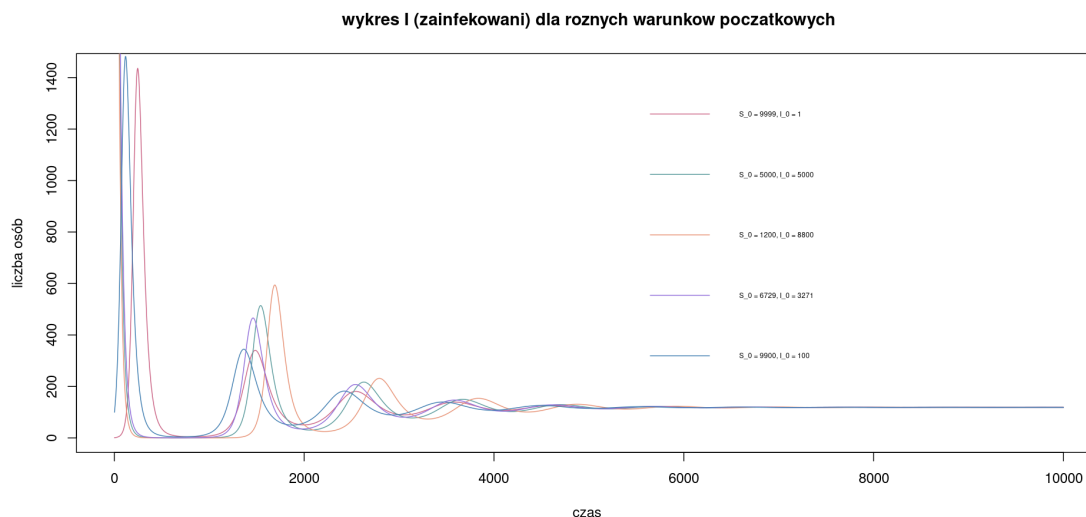
```

Wyniki tych symulacji przedstawiamy na wspólnym wykresie.

```

plot(c(1:length(I_c)), I_c, main = "wykres_I_(zainfekowani)_dla_roznych_warunkow_poczatkowych",
type = "l", col = "palevioletred3", xlab = "czas_", ylab = "liczba_os b")
lines(c(1:length(I_c)),I_c3v1, col="cadetblue")
lines(c(1:length(I_c)),I_c3v3, col="darksalmon")
lines(c(1:length(I_c)),I_c3v4, col="mediumpurple")
lines(c(1:length(I_c)),I_c3v5, col="steelblue")
legend("topright", c("S_0=9999,I_0=1", "S_0=5000,I_0=5000", "S_0=1200,I_0=8800",
"S_0=6729,I_0=3271", "S_0=9900,I_0=100"),
col = c("palevioletred3", "cadetblue","darksalmon","mediumpurple","steelblue" ),
lty = 1, bty = "n", cex=0.5)

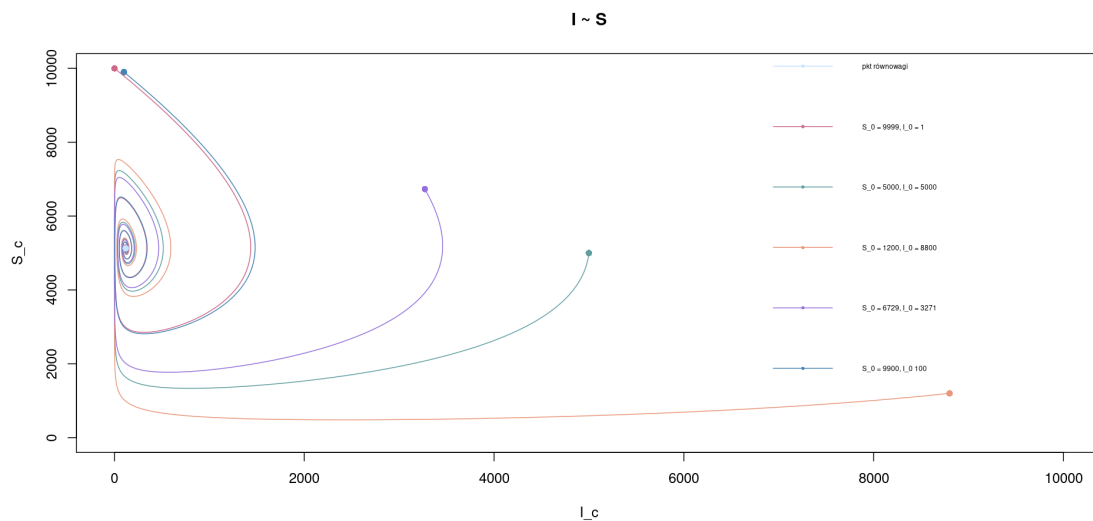
```



Rysunek 36: Wykres zainfekowanych przy parametrach $c) (\beta, \gamma, \mu) = (0.8, 0.4, 0.01)$ i dla różnych warunków początkowych

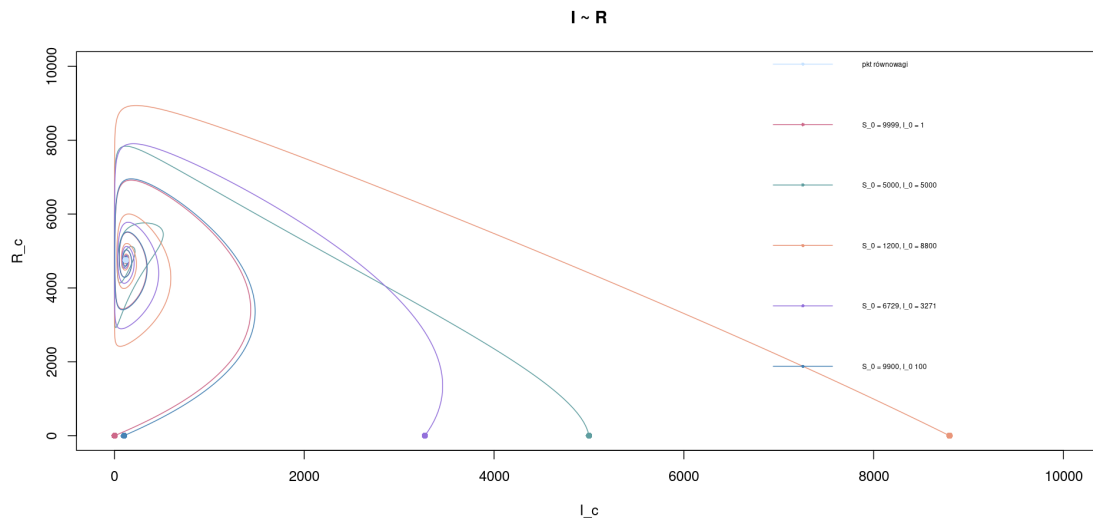
Wykreślmy także portrety fazowe $I \sim S$ oraz $I \sim R$. Rozpoczynając od $I \sim S$.

```
plot(I_c, S_c, main = "I~S", type = "l", col="palevioletred3",
xlim = c(0,10000), ylim = c(0,10000))
lines(I_c3v1, S_c3v1, col = "cadetblue")
lines(I_c3v3, S_c3v3, col = "darksalmon")
lines(I_c3v4, S_c3v4, col = "mediumpurple")
lines(I_c3v5, S_c3v5, col = "steelblue")
points(rownowagaSIR_c[2], rownowagaSIR_c[1], pch = 16, col = "slategray1")
points(I_c[1], y = S_c[1], pch = 16, col = "palevioletred3")
points(I_c3v1[1], y = S_c3v1[1], pch = 16, col = "cadetblue")
points(I_c3v3[1], y = S_c3v3[1], pch = 16, col = "darksalmon")
points(I_c3v4[1], y = S_c3v4[1], pch = 16, col = "mediumpurple")
points(I_c3v5[1], y = S_c3v5[1], pch = 16, col = "steelblue")
legend("topright", c("pkt_rownowagi", "S_0=9999, I_0=1", "S_0=5000, I_0=5000",
"S_0=1200, I_0=8800", "S_0=6729, I_0=3271", "S_0=9900, I_0=100"),
col = c("slategray1", "palevioletred3", "cadetblue", "darksalmon", "mediumpurple", "steelblue"),
lty = 1, bty = "n", cex=0.5, pch=c(16,16), inset = -0.12)
```



Rysunek 37: Portret fazowy $I \sim S$ przy parametrach c $(\beta, \gamma, \mu) = (0.8, 0.4, 0.01)$ i dla różnych warunków początkowych

```
plot(I_c, R_c, main = "I_c ~ R_c", type = "l", col = "palevioletred3",
     xlim = c(0, 10000), ylim = c(0, 10000))
lines(I_c3v1, R_c3v1, col = "cadetblue")
lines(I_c3v3, R_c3v3, col = "darksalmon")
lines(I_c3v4, R_c3v4, col = "mediumpurple")
lines(I_c3v5, R_c3v5, col = "steelblue")
points(rownowagaSIR_c[2], rownowagaSIR_c[3], pch = 16, col = "slategray1")
points(I_c[1], y = R_c[1], pch = 16, col = "palevioletred3")
points(I_c3v1[1], y = R_c3v1[1], pch = 16, col = "cadetblue")
points(I_c3v3[1], y = R_c3v3[1], pch = 16, col = "darksalmon")
points(I_c3v4[1], y = R_c3v4[1], pch = 16, col = "mediumpurple")
points(I_c3v5[1], y = R_c3v5[1], pch = 16, col = "steelblue")
legend("topright", c("pkt rownowagi", "S_0=9999, I_0=1", "S_0=5000, I_0=5000",
" S_0=1200, I_0=8800", "S_0=6729, I_0=3271", "S_0=9900, I_0=100"),
     col = c("slategray1", "palevioletred3", "cadetblue", "darksalmon", "mediumpurple", "steelblue"),
     lty = 1, bty = "n", cex = 0.5, pch = c(16, 16), inset = -0.12)
```



Rysunek 38: Portret fazowy $I \sim R$ przy parametrach c) $(\beta, \gamma, \mu) = (0.8, 0.4, 0.01)$ i dla różnych warunków początkowych

W N I O S E K: Różne warunki początkowe nie wpływają na interpretację modelu przy zadanych parametrach $(\beta, \gamma, \mu) = (0.8, 0.4, 0.01)$. Interpretujemy każdą z tych symulacji analogicznie do tej przeprowadzonej w Zadaniu 2.2.c).

3.d)

Rozważane parametry:

d) $(\beta, \gamma, \mu) = (0.8, 0.7, 0.01)$

Różne warunki początkowe, dla których przeprowadzamy symulacje:

```
# S_0 = 5000
# I_0 = 5000
x_d3v1 = simulation_mickens(5000, 5000, 0, h = 0.1,
b = 0.8, g = 0.7, m = 0.01, TT = 1000, N = 10000)
S_d3v1 = x_d3v1[[1]]
I_d3v1 = x_d3v1[[2]]
R_d3v1 = x_d3v1[[3]]
```

```
# S_0 = 7000
# I_0 = 3000
x_d3v2 = simulation_mickens(7000, 3000, 0, h = 0.1,
b = 0.8, g = 0.7, m = 0.01, TT = 1000, N = 10000)
S_d3v2 = x_d3v2[[1]]
I_d3v2 = x_d3v2[[2]]
R_d3v1 = x_d3v2[[3]]
```

```
# S_0 = 1200
# I_0 = 8800
x_d3v3 = simulation_mickens(1200, 8800, 0, h = 0.1,
b = 0.8, g = 0.7, m = 0.01, TT = 1000, N = 10000)
S_d3v3 = x_d3v3[[1]]
I_d3v3 = x_d3v3[[2]]
R_d3v3 = x_d3v3[[3]]
```

```
# S_0 = 6729
# I_0 = 3271
x_d3v4 = simulation_mickens(6729, 3271, 0, h = 0.1,
b = 0.8, g = 0.7, m = 0.01, TT = 1000, N = 10000)
S_d3v4 = x_d3v4[[1]]
I_d3v4 = x_d3v4[[2]]
R_d3v4 = x_d3v4[[3]]
```

```
# S_0 = 9900
# I_0 = 100
x_d3v5 = simulation_mickens(9900, 100, 0, h = 0.1,
b = 0.8, g = 0.7, m = 0.01, TT = 1000, N = 10000)
S_d3v5 = x_d3v5[[1]]
I_d3v5 = x_d3v5[[2]]
R_d3v5 = x_d3v5[[3]]
```

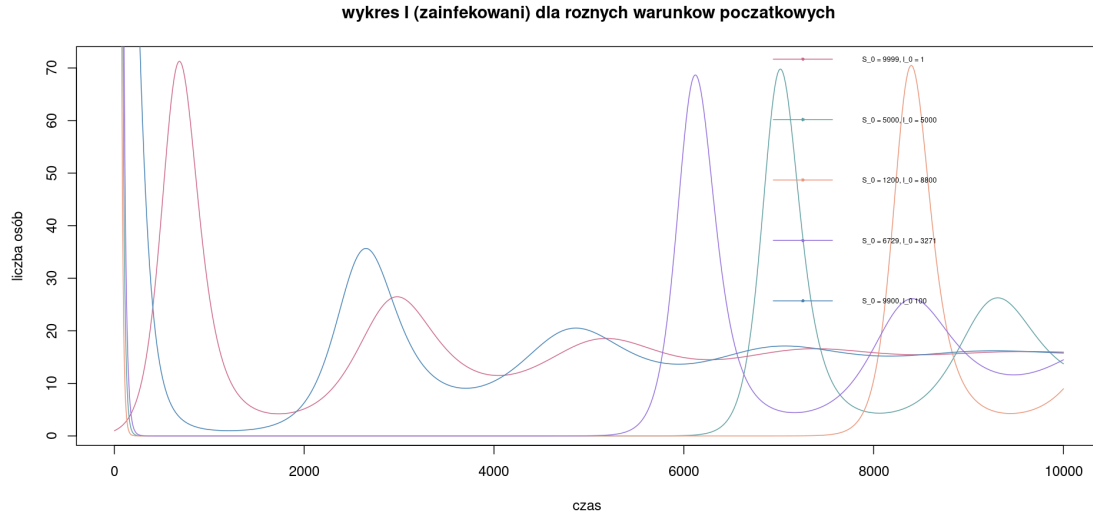
Wyniki tych symulacji przedstawiamy na wspólnym wykresie.

```
plot(c(1:length(I_d)), I_d, main = "wykres_I_(zainfekowani)_dla_roznych_warunkow_poczatkowych",
type = "l", col = "palevioletred3", xlab = "czas", ylab = "liczba_osob")
```

```

lines(c(1:length(I_d)), I_d3v1, col="cadetblue")
lines(c(1:length(I_d)), I_d3v3, col="darksalmon")
lines(c(1:length(I_d)), I_d3v4, col="mediumpurple")
lines(c(1:length(I_d)), I_d3v5, col="steelblue")
legend("topright", c("S_0=9999, I_0=1", "S_0=5000, I_0=5000", "S_0=1200, I_0=8800",
"S_0=6729, I_0=3271", "S_0=9900, I_0=100"), col = c("palevioletred3", "cadetblue",
"darksalmon", "mediumpurple", "steelblue" ),
lty = 1, bty = "n", cex=0.5, pch=c(16,16), inset = -0.12)

```



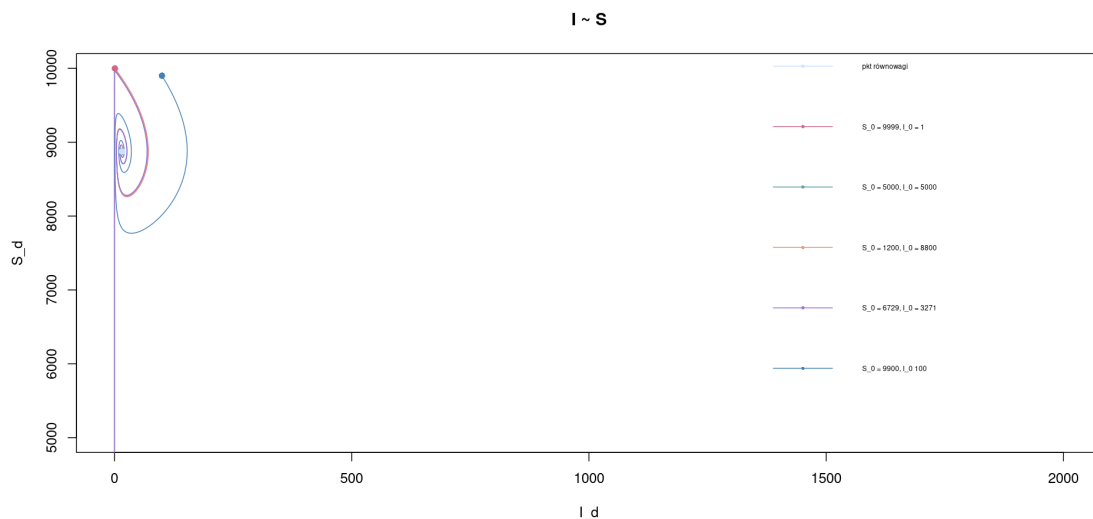
Rysunek 39: Wykres zainfekowanych przy parametrach d) $(\beta, \gamma, \mu) = (0.8, 0.7, 0.01)$ i dla różnych warunków początkowych

Wykreślmy także portrety fazowe $I \sim S$ oraz $I \sim R$. Rozpoczynając od $I \sim S$.

```

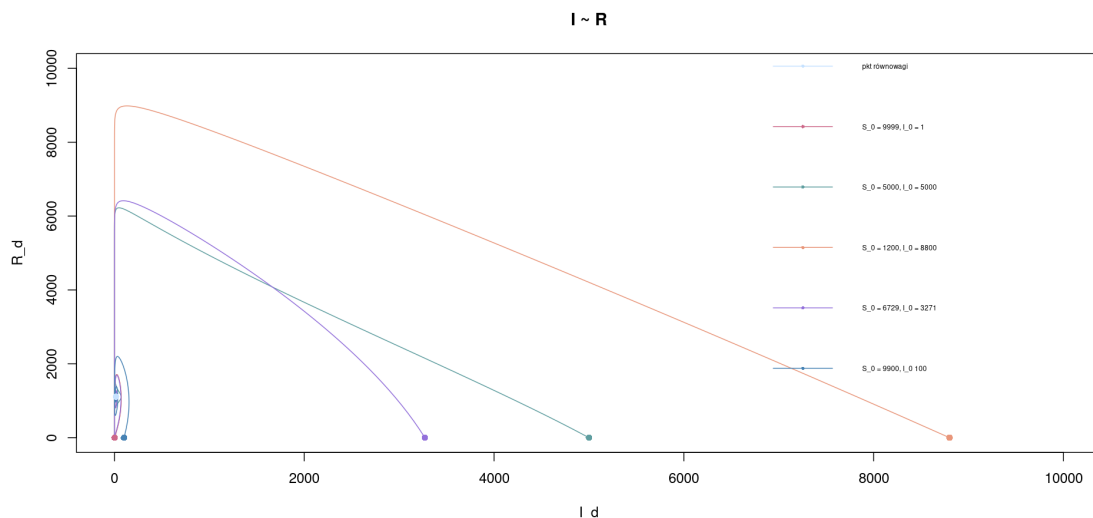
plot(I_d, S_d, main = "I~S", type = "l", col="palevioletred3",
xlim = c(0,2000), ylim= c(5000,10000))
lines(I_d3v1, S_d3v1, col = "cadetblue")
lines(I_d3v3, S_d3v3, col = "darksalmon")
lines(I_d3v4, S_d3v4, col = "mediumpurple")
lines(I_d3v5, S_d3v5, col = "steelblue")
points(rownowagaSIR_d[2], rownowagaSIR_d[1], pch = 16, col = "slategray1")
points(I_d[1], y = S_d[1], pch = 16, col = "palevioletred3")
points(I_d3v1[1], y = S_d3v1[1], pch = 16, col = "cadetblue")
points(I_d3v3[1], y = S_d3v3[1], pch = 16, col = "darksalmon")
points(I_d3v4[1], y = S_d3v4[1], pch = 16, col = "mediumpurple")
points(I_d3v5[1], y = S_d3v5[1], pch = 16, col = "steelblue")
legend("topright", c("pkt_r wnowagi", "S_0=9999, I_0=1", "S_0=5000, I_0=5000",
"S_0=1200, I_0=8800", "S_0=6729, I_0=3271", "S_0=9900, I_0=100"),
col = c("slategray1", "palevioletred3", "cadetblue", "darksalmon", "mediumpurple", "steelblue"),
lty = 1, bty = "n", cex=0.5, pch=c(16,16), inset = -0.12)

```

Rysunek 40: Portret fazowy $I \sim S$ przy parametrach d $(\beta, \gamma, \mu) = (0.8, 0.7, 0.01)$ i dla różnych warunków początkowych

```
plot(I_d, R_d, main = "I~R", type = "l", col="palevioletred3",
     xlim = c(0,10000), ylim=c(0,10000) )
lines(I_d3v1, R_d3v1, col = "cadetblue")
lines(I_d3v3, R_d3v3, col = "darksalmon")
lines(I_d3v4, R_d3v4, col = "mediumpurple")
lines(I_d3v5, R_d3v5, col = "steelblue")
points(rownowagaSIR_d[2], rownowagaSIR_d[3], pch = 16, col = "slategray1")
points(I_d[1], y = R_d[1], pch = 16, col = "palevioletred3")
points(I_d3v1[1], y = R_d3v1[1], pch = 16, col = "cadetblue")
points(I_d3v3[1], y = R_d3v3[1], pch = 16, col = "darksalmon")
points(I_d3v4[1], y = R_d3v4[1], pch = 16, col = "mediumpurple")
points(I_d3v5[1], y = R_d3v5[1], pch = 16, col = "steelblue")
legend("topright", c("pkt rownowagi", "S_0=9999, I_0=1", "S_0=5000, I_0=5000",
" S_0=1200, I_0=8800", "S_0=6729, I_0=3271", "S_0=9900, I_0=100"),
col = c("slategray1", "palevioletred3", "cadetblue", "darksalmon", "mediumpurple", "steelblue"),
lty = 1, bty = "n", cex=0.5, pch=c(16,16), inset = -0.12)
```



Rysunek 41: Portret fazowy $I \sim R$ przy parametrach d $(\beta, \gamma, \mu) = (0.8, 0.7, 0.01)$ i dla różnych warunków początkowych

W N I O S E K: Różne warunki początkowe nie wpływają na interpretację modelu przy zadanych parametrach $(\beta, \gamma, \mu) = (0.8, 0.7, 0.01)$. Interpretujemy każdą z tych symulacji analogicznie do tej przeprowadzonej w Zadaniu 2.2.d).

W N I O S E K D O Z A D A N I A 3: Różne warunki początkowe nie wpływają na interpretację.

Zadanie 4.

Przyjmujemy liczebność populacji N równą 10 000.

Wyznamy rozwiązania modelu przeskalowane do liczebności populacji $N = 1$.

Rozwiązania te będą się mieścić w przedziale $[0, 1]$.

Sprawdzimy charakterystykę trajektorii dla zestawów parametrów z Zadania 2, a następnie dla zestawu b) sprawdzimy kształt trajektorii dla różnych liczebności populacji.

4.a)

Rozważany zestaw parametrów:

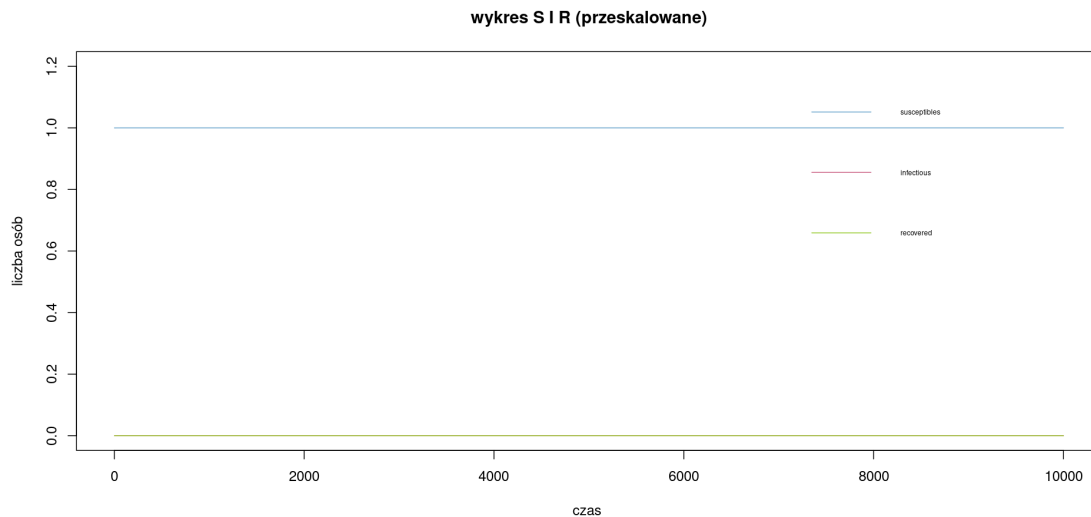
a) $(S_0, I_0, \beta, \gamma, \mu) = (9999, 1, 0.5, 0.8, 0.4)$.

Przeskalowujemy S, I oraz R i otrzymane wyniki przedstawiamy na wspólnym wykresie.

a przeskalowane

```
x_a4 = simulation_mickens(9999, 1, 0, h = 0.1, b = 0.5, g = 0.8, m = 0.4, TT = 1000, N = 10000)
S_a4 = x_a4[[1]]/10000
I_a4 = x_a4[[2]]/10000
R_a4 = x_a4[[3]]/10000
```

```
plot(c(1:length(S_a4)), S_a4, main = "wykres_S_I_R_(przeskalowane)", type = "l",
     col = "skyblue3", xlab = "czas", ylab = "liczba_osob", ylim = c(0,1.2))
lines(c(1:length(I_a4)), I_a4, type = "l", col = "palevioletred3")
lines(c(1:length(R_a4)), R_a4, type = "l", col = "yellowgreen")
legend("topright", c("susceptibles", "infectious", "recovered"),
     col = c("skyblue3", "palevioletred3", "yellowgreen"),
     lty = 1, bty = "n", cex=0.5)
```



Rysunek 42: Wykres SIR (dane przeskalowane) przy parametrach a) $(\beta, \gamma, \mu) = (0.5, 0.8, 0.4)$

W N I O S E K: Charakterystyka nie zmienia się po przeskalowaniu do populacji liczącej $N = 1$ jednostek. Trajektorię interpretujemy tak samo jak w Zadaniu 2.2.a) - ma ona ten sam kształt, jedynie przeskalowany na osi y do odcinka $[0, 1]$.

4.b)

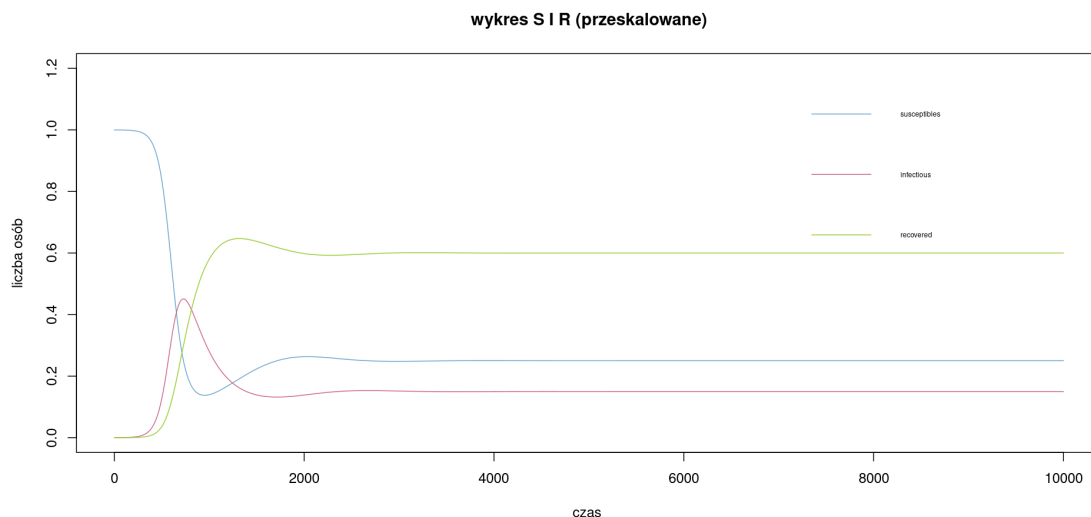
Rozważany zestaw parametrów:

b) $(S_0, I_0, \beta, \gamma, \mu) = (9999, 1, 0.2, 0.04, 0.01)$.

Przeskalowujemy S, I oraz R i otrzymane wyniki przedstawiamy na wspólnym wykresie.

```
# b przeskalowane
x_b4 = simulation_mickens(9999, 1, 0, h = 0.1, b = 0.2, g = 0.04, m = 0.01, TT = 1000, N = 10000)
S_b4 = x_b4[[1]]/10000
I_b4 = x_b4[[2]]/10000
R_b4 = x_b4[[3]]/10000

plot(c(1:length(S_b4)), S_b4, main = "wykres_SIR(przeskalowane)", type = "l",
col = "skyblue3", xlab = "czas", ylab = "liczba_osob", ylim = c(0,1.2))
lines(c(1:length(I_b4)), I_b4, type = "l", col = "palevioletred3")
lines(c(1:length(R_b4)), R_b4, type = "l", col = "yellowgreen")
legend("topright", c("susceptibles", "infectious", "recovered"),
col = c("skyblue3", "palevioletred3", "yellowgreen"), lty = 1, bty = "n", cex=0.5)
```



Rysunek 43: Wykres *SIR* (dane przeskalowane) przy parametrach b) $(\beta, \gamma, \mu) = (0.2, 0.04, 0.01)$

W N I O S E K: Charakterystyka nie zmienia się po przeskalowaniu do populacji liczącej $N = 1$ jednostek. Trajektorię interpretujemy tak samo jak w Zadaniu 2.2.b) - ma ona ten sam kształt, jedynie przeskalowany na osi y do odcinka $[0, 1]$.

4.c)

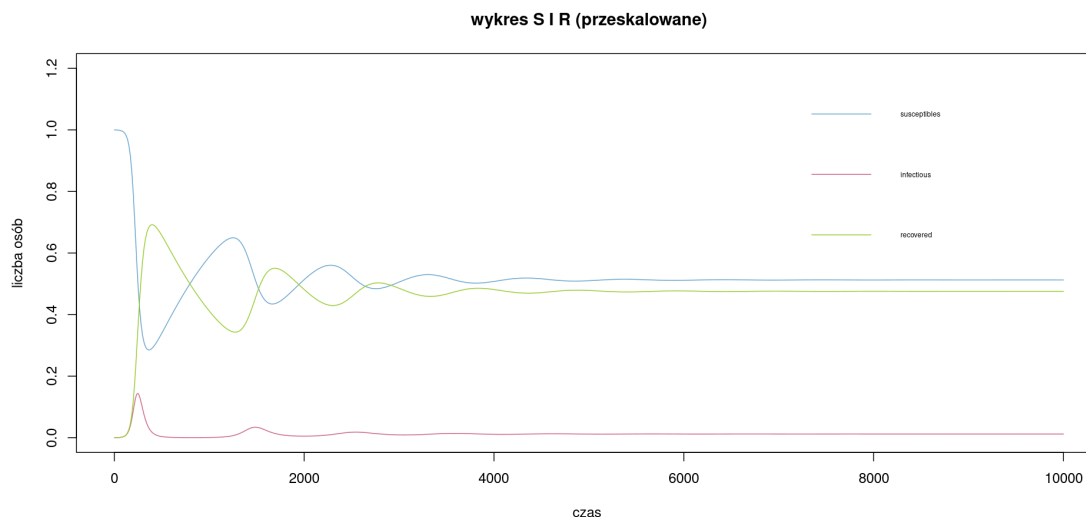
Rozważany zestaw parametrów:

c) $(S_0, I_0, \beta, \gamma, \mu) = (9999, 1, 0.8, 0.4, 0.01)$.

Przeskalowujemy S, I oraz R i otrzymane wyniki przedstawiamy na wspólnym wykresie.

```
# c przeskalowane
x_c4 = simulation_mickens(9999, 1, 0, h = 0.1, b = 0.8, g = 0.4, m = 0.01, TT = 1000, N = 10000)
S_c4 = x_c4[[1]]/10000
I_c4 = x_c4[[2]]/10000
R_c4 = x_c4[[3]]/10000

plot(c(1:length(S_c4)), S_c4, main = "wykres_SIR_(przeskalowane)", type = "l",
col = "skyblue3", xlab = "czas", ylab = "liczba_osob", ylim = c(0,1.2))
lines(c(1:length(I_c4)), I_c4, type = "l", col = "palevioletred3")
lines(c(1:length(R_c4)), R_c4, type = "l", col = "yellowgreen")
legend("topright", c("susceptibles", "infectious", "recovered"),
col = c("skyblue3", "palevioletred3", "yellowgreen"), lty = 1, bty = "n", cex=0.5)
```



Rysunek 44: Wykres SIR (dane przeskalowane) przy parametrach c $(\beta, \gamma, \mu) = (0.8, 0.4, 0.01)$

W N I O S E K: Charakterystyka nie zmienia się po przeskalowaniu do populacji liczącej $N = 1$ jednostek. Trajektorię interpretujemy tak samo jak w Zadaniu 2.2.c) - ma ona ten sam kształt, jedynie przeskalowany na osi y do odcinka $[0, 1]$.

4.d)

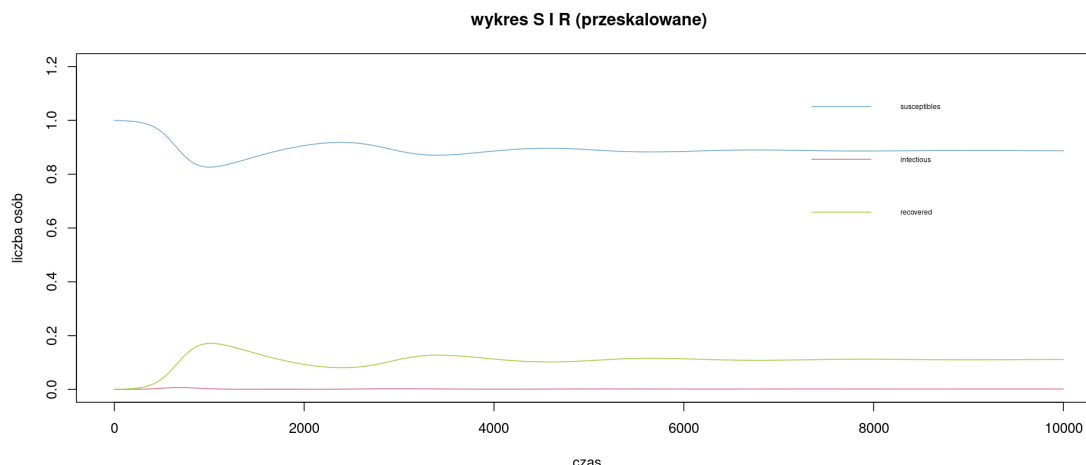
Rozważany zestaw parametrów:

d) $(S_0, I_0, \beta, \gamma, \mu) = (9999, 1, 0.8, 0.7, 0.01)$.

Przeskalowujemy S, I oraz R i otrzymane wyniki przedstawiamy na wspólnym wykresie.

```
# d przeskalowane
x_d4 = simulation_mickens(9999, 1, 0, h = 0.1, b = 0.8, g = 0.7, m = 0.01, TT = 1000, N = 10000)
S_d4 = x_d4[[1]]/10000
I_d4 = x_d4[[2]]/10000
R_d4 = x_d4[[3]]/10000

plot(c(1:length(S_d4)), S_d4, main = "wykres_S_I_R_(przeskalowane)", type = "l",
     col = "skyblue3", xlab = "czas", ylab = "liczba_osob", ylim = c(0,1.2))
lines(c(1:length(I_d4)), I_d4, type = "l", col = "palevioletred3")
lines(c(1:length(R_d4)), R_d4, type = "l", col = "yellowgreen")
legend("topright", c("susceptibles", "infectious", "recovered"),
     col = c("skyblue3", "palevioletred3", "yellowgreen"), lty = 1, bty = "n", cex=0.5)
```



Rysunek 45: Wykres SIR (dane przeskalowane) przy parametrach $d) (\beta, \gamma, \mu) = (0.8, 0.7, 0.01)$

W N I O S E K: Charakterystyka nie zmienia się po przeskalowaniu do populacji liczącej $N = 1$ jednostek. Trajektorię interpretujemy tak samo jak w Zadaniu 2.2.d) - ma ona ten sam kształt, jedynie przeskalowany na osi y do odcinka $[0, 1]$.

4.b) dla różnych liczebności populacji

Rozważamy zestaw parametrów:

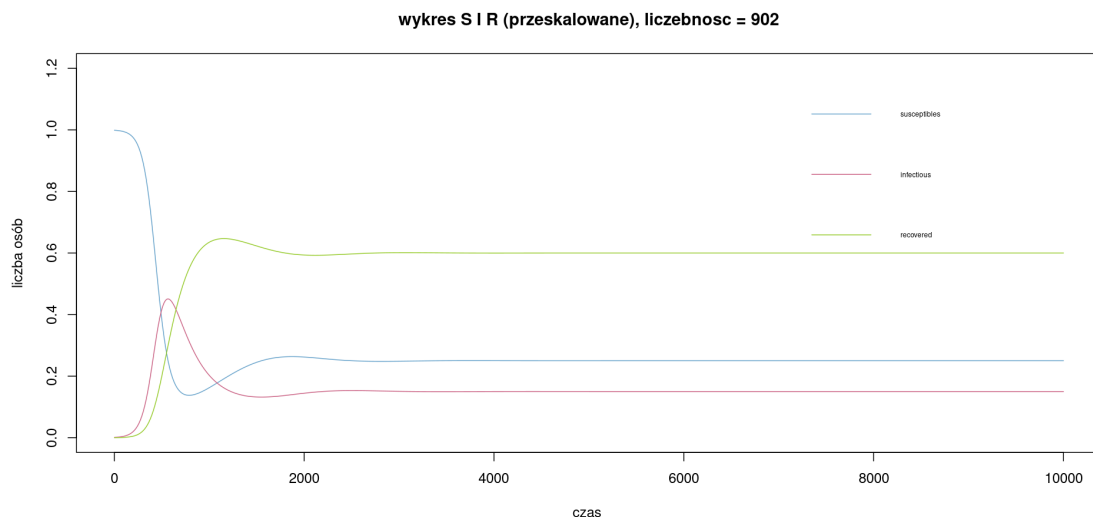
b) $(S_0, I_0, \beta, \gamma, \mu) = (9999, 1, 0.2, 0.04, 0.01)$.

Oraz różne liczebności populacji.

Przeskalowujemy S, I oraz R i otrzymane wyniki przedstawiamy na wspólnym wykresie.

```
# b przeskalowane dla roznych liczebnosci populacji
# liczebosc = 902
x_b4v1 = simulation_mickens(901, 1, 0, h = 0.1,
b = 0.2, g = 0.04, m = 0.01, TT = 1000, N = 902)
S_b4v1 = x_b4v1[[1]]/902
I_b4v1 = x_b4v1[[2]]/902
R_b4v1 = x_b4v1[[3]]/902

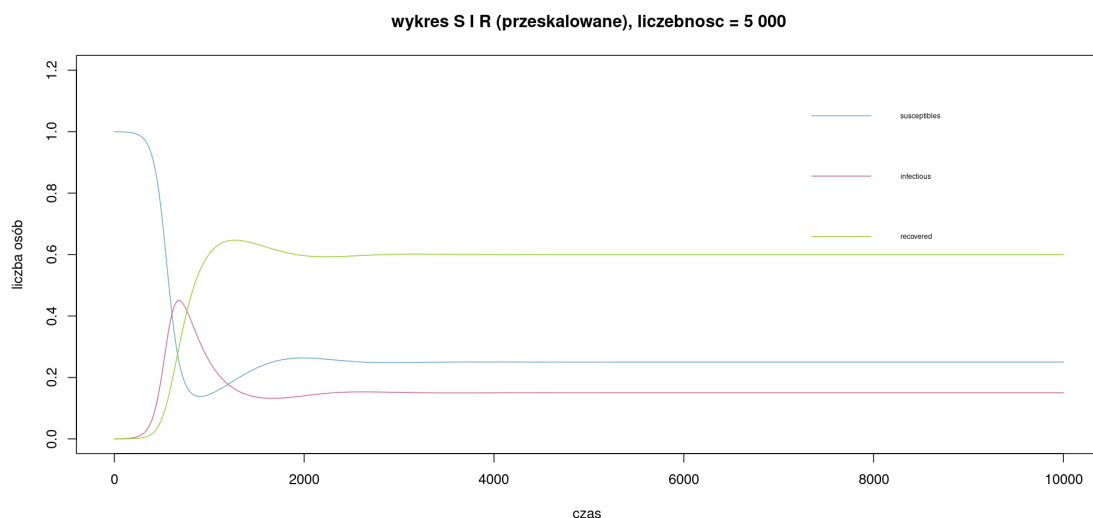
plot(c(1:length(S_b4v1)), S_b4v1, main = "wykres_S_I_R_(przeskalowane),_liczebosc_=_902",
type = "l", col = "skyblue3", xlab = "czas_", ylab = "liczba_osob", ylim = c(0,1.2))
lines(c(1:length(I_b4v1)), I_b4v1, type = "l", col = "palevioletred3")
lines(c(1:length(R_b4v1)), R_b4v1, type = "l", col = "yellowgreen")
legend("topright", c("susceptibles", "infectious", "recovered"),
col = c("skyblue3", "palevioletred3", "yellowgreen"), lty = 1, bty = "n", cex=0.5)
```



Rysunek 46: Wykres *SIR* (dane przeskalowane) przy parametrach $b) (\beta, \gamma, \mu) = (0.2, 0.04, 0.01)$ i dla populacji liczącej 902 osobników

```
# liczebność = 5000
x_b4v2 = simulation_mickens(4999, 1, 0, h = 0.1, b = 0.2, g = 0.04, m = 0.01, TT=1000, N=5000)
S_b4v2 = x_b4v2[[1]]/5000
I_b4v2 = x_b4v2[[2]]/5000
R_b4v2 = x_b4v2[[3]]/5000
```

```
plot(c(1:length(S_b4v2)), S_b4v2, main = "wykres_S_I_R_(przeskalowane),_liczebność_=5_000",
type = "l", col = "skyblue3", xlab = "czas", ylab = "liczba_osob", ylim = c(0,1.2))
lines(c(1:length(I_b4v2)), I_b4v2, type = "l", col = "palevioletred3")
lines(c(1:length(R_b4v2)), R_b4v2, type = "l", col = "yellowgreen")
legend("topright", c("susceptibles", "infectious", "recovered"),
col = c("skyblue3", "palevioletred3", "yellowgreen"), lty = 1, bty = "n", cex=0.5)
```



Rysunek 47: Wykres *SIR* (dane przeskalowane) przy parametrach $b) (\beta, \gamma, \mu) = (0.2, 0.04, 0.01)$ i dla populacji liczącej 5 000 osobników

```
# liczebność = 90 000
x_b4v3 = simulation_mickens(89999, 1, 0, h=0.1, b=0.2, g=0.04, m=0.01, TT=1000, N=90000)
```

```

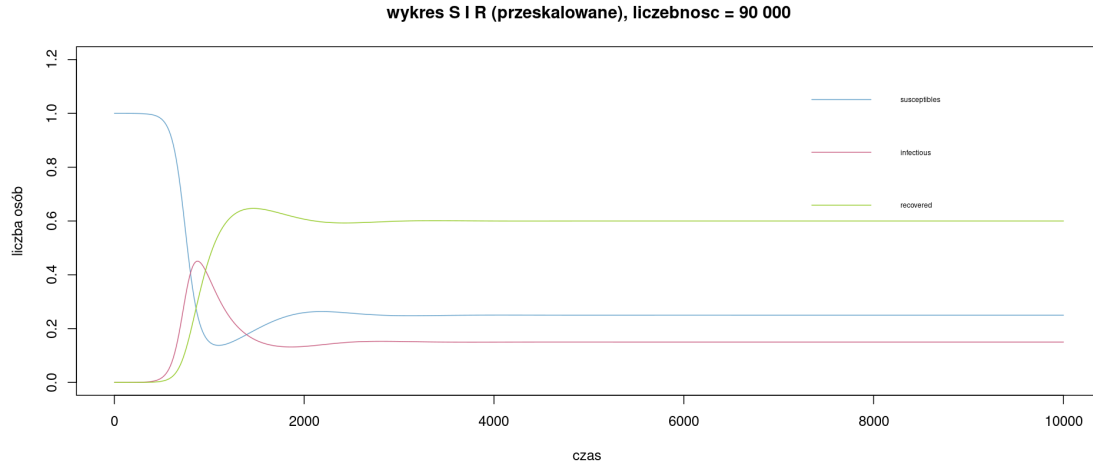
S_b4v3 = x_b4v3[[1]]/90000
I_b4v3 = x_b4v3[[2]]/90000
R_b4v3 = x_b4v3[[3]]/90000

```

```

plot(c(1:length(S_b4v3)), S_b4v3, main = "wykres_S_I_R_(przeskalowane),_liczebosc_=90_000",
type = "l", col = "skyblue3", xlab = "czas_", ylab = "liczba_osob", ylim = c(0,1.2))
lines(c(1:length(I_b4v3)), I_b4v3, type = "l", col = "palevioletred3")
lines(c(1:length(R_b4v3)), R_b4v3, type = "l", col = "yellowgreen")
legend("topright", c("susceptibles", "infectious", "recovered"),
col = c("skyblue3", "palevioletred3", "yellowgreen"), lty = 1, bty = "n", cex=0.5)

```



Rysunek 48: Wykres *SIR* (dane przeskalowane) przy parametrach b) $(\beta, \gamma, \mu) = (0.2, 0.04, 0.01)$ i dla populacji liczącej 90 000 osobników

```

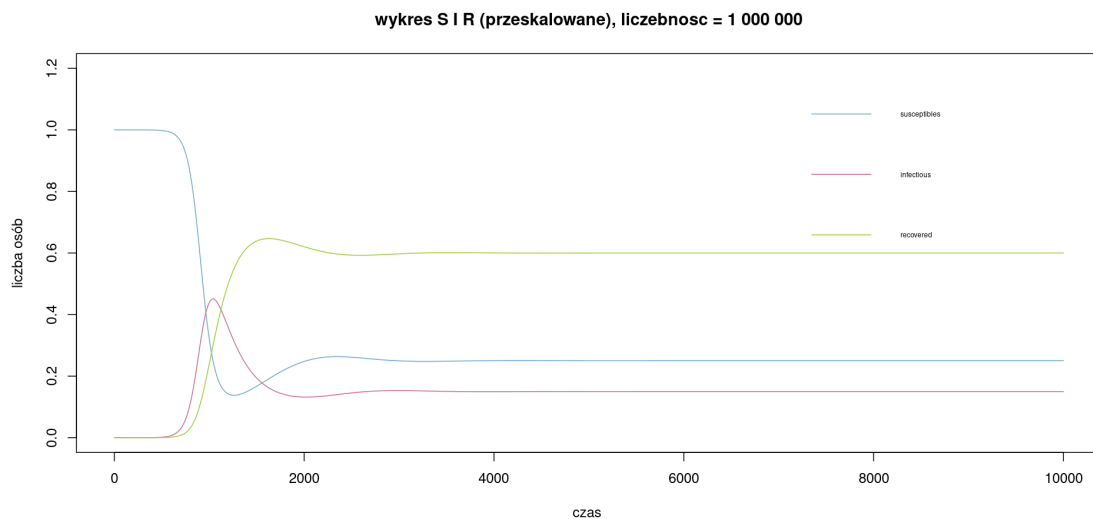
# liczebosc = 1 000 000
x_b4v4 = simulation_mickens(999999, 1, 0, h=0.1, b =0.2, g=0.04, m=0.01, TT=1000, N=1000000)
S_b4v4 = x_b4v4[[1]]/1000000
I_b4v4 = x_b4v4[[2]]/1000000
R_b4v4 = x_b4v4[[3]]/1000000

```

```

plot(c(1:length(S_b4v4)), S_b4v4, main = "wykres_S_I_R_(przeskalowane),_liczebosc_=1_000_000",
type = "l", col = "skyblue3", xlab = "czas_", ylab = "liczba_osob", ylim = c(0,1.2))
lines(c(1:length(I_b4v4)), I_b4v4, type = "l", col = "palevioletred3")
lines(c(1:length(R_b4v4)), R_b4v4, type = "l", col = "yellowgreen")
legend("topright", c("susceptibles", "infectious", "recovered"),
col = c("skyblue3", "palevioletred3", "yellowgreen"), lty = 1, bty = "n", cex=0.5)

```

Rysunek 49: Wykres SIR (dane przeskalowane) przy parametrach b) $(\beta, \gamma, \mu) = (0.2, 0.04, 0.01)$ i dla populacji liczącej 1 000 000 osobników

W N I O S E K: Charakterystyka nie zmienia się po przeskalowaniu różnych liczebności populacji do populacji liczącej $N = 1$ jednostek. Otrzymaliśmy trajektorie, które niczym się od siebie nie różnią. Mają te same kształty, pomimo różnych liczebności populacji.

W N I O S E K D O Z A D A N I A 4: Charakterystyka trajektorii pozostaje bez zmian po przeskalowaniu do liczebności populacji $N = 1$, jedyną różnicą jest to, że rozwiązania zawierają się w przedziale $[0, 1]$. Różne liczebności populacji przed przeskalowaniem również nie mają wpływu na kształt trajektorii.

Estymacja parametrów

Zadanie 5.

Instalujemy pakiet **epimdr** i rozważamy plik z danymi **niamey**. Sumujemy kolumny `cases_1`, `cases_2` oraz `cases_3`. Następnie rysujemy wykres przedstawiający ilość zarażeń dla trzech dystryktów razem.

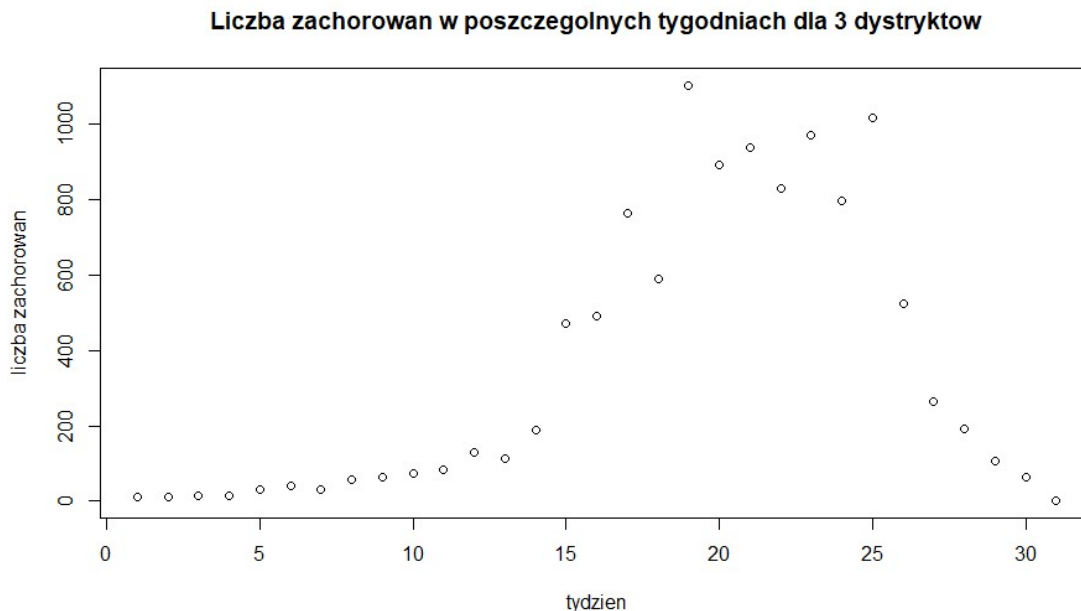
```
# instalujemy pakiet
install.packages("epimdr")
library("epimdr")

# wyświetlamy zbior danych niamey
niamey

# sumujemy wartości cases_1, cases_2, cases_3
suma <- (niamey['cases_1'] + niamey['cases_2'] + niamey['cases_3'])
suma

install.packages("dplyr")
library("dplyr")
# wyświetlamy numery tygodni (1-31)
tyg <- select(niamey, absweek)
tyg

# tworzymy tabelkę zawierającą numer tygodnia oraz liczbę zachorowań dla trzech dystryktów razem
dane <- cbind(tyg, suma)
dane
plot(dane, xlab='tydzien', ylab='liczba_zachorowan',
main='Liczba_zachorowan_w_poszczegolnych_tygodniach_dla_3_dystryktow')
```



Rysunek 50: Wykres przedstawiający liczbę zachorowań w poszczególnych tygodniach dla 3 dystryktów.

Zadanie 6.

Posiadane dane dotyczą trzech różnych dystryktów jednego miasta, dla których nie znamy liczebności populacji, znamy natomiast orientacyjną liczebność populacji w Niamey w latach 2003/2004, która wynosi $N = 769\,454$. Zatem w zadaniu będziemy rozważać dane jako suma informacji ze wszystkich trzech dystryktów. Jako warunki początkowe dla populacji zainfekowanych I_0 przyjmujemy pierwszy element ze zbioru danych, natomiast $S_0 = N - I_0$. Średni wiek życia w Niamey to zaledwie 15 lat, więc przyjmujemy parametr $\mu = 1/(15 \cdot 365)$.

$$I_0 = 11$$

$$S_0 = 769454 - 11 = 769443$$

Dla zestawu parametrów $(\beta, \gamma) = (0.79, 0.7)$ wyznaczamy współczynnik R_0 .

zgodnie ze wzorem

$R_0 = b / (m + g)$

$$m = 1/(15 \cdot 365)$$

$$b = 0.79$$

$$g = 0.7$$

$$R_{0_6} \leftarrow b / (m+g)$$

$$R_{0_6}$$

Dla takiego zestawu parametrów współczynnik R_0 wynosi 1.128277.

Rozwiązanie numeryczne modelu dla podanych parametrów i wartości początkowych.

ponownie przypominamy metode numeryczna mickensa

$S_{n1} \leftarrow \text{function}(h, m, N, S_n, b, I_n) \text{ return } ((h \cdot m \cdot N + S_n) / (1 + h \cdot m + (b/N) \cdot h \cdot I_n))$

$I_{n1} \leftarrow \text{function}(b, N, h, S_{n1}, I_n, g, m) \text{ return } (((b/N) \cdot h \cdot S_{n1} \cdot I_n + I_n) / (1 + g \cdot h + m \cdot h))$

$R_{n1} \leftarrow \text{function}(g, h, I_{n1}, R_n, m) \text{ return } ((g \cdot h \cdot I_{n1} + R_n) / (1 + m \cdot h))$

funkcja dokonujaca symulacji metody numerycznej mickensa

tutaj zmieniamy R_0 , niech bedzie liczone z paramterow $b/(m+g)$,

wczesniej bylo argumentem funkcji

$\text{simulation_mickens_2} \leftarrow \text{function}(S_0, I_0, h, b, g, m, TT, N) \{$
 $n = TT / h$

$S = \{\}$

$S[1] \leftarrow S_0$

$I = \{\}$

$I[1] \leftarrow I_0$

$R = \{\}$

$R[1] \leftarrow b / (m+g)$

 for(i in 2:n) {

```

    S_n_1 = S_n1(h, m, N, S[i-1], b, I[i-1])
    S[i] = S_n_1

    I_n_1 = I_n1(b, N, h, S[i], I[i-1], g, m)
    I[i] = I_n_1

    R_n_1 = R_n1(g, h, I[i], R[i-1], m)
    R[i] = R_n_1
}
return(list(S,I,R))
}

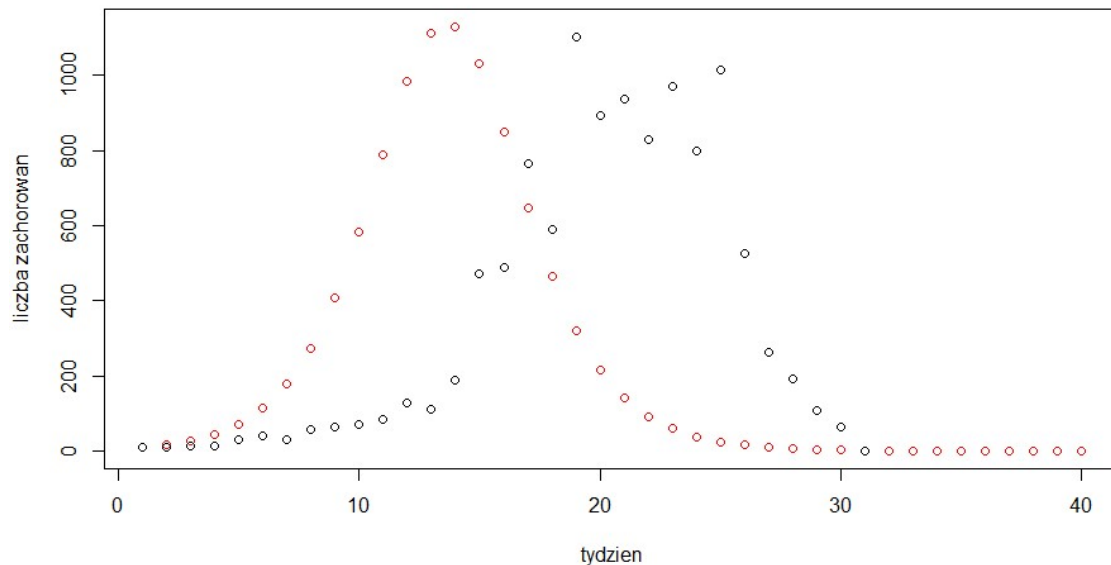
# podajemy parametry i wartosci wejsciowe
b = 0.79
g = 0.7
m = 1/(15*365)
R_0 = b/(m+g)
N = 769454
I_0 = dane[1,2]
S_0 = N - I_0
h = 5
TT = 1000

symulacja_6 <- simulation_mickens_2(S_0, I_0, h, b, g, m, TT, N)
symulacja_6

# wybieramy z symulacji druga wartosc, odpowiadajaca grupie zarazonych,
# przeliczamy to na tygodnie
numeryczne <- symulacja_6[[2]][seq(1,length(symulacja_6[[2]]),h)]
numeryczne

plot(numeryczne, xlab='tydzien', ylab='liczba_zachorowan', col='red')
points(dane)

```



Rysunek 51: Na jednym wykresie przedstawiono dane (kolor czarny) oraz rozwiązanie numeryczne (kolor czerwony).

Możemy zauważyć, że kształtem wykresy są podobne, aczkolwiek rozwiązanie numeryczne szybciej rośnie, zatem będzie potrzeba estymacji parametrów β, γ .

Zadanie 7.

Korzystamy z metody estymacji najmniejszych kwadratów. W tym celu minimalizujemy funkcję celu

$$BNK(p) = \sum_{j=1}^{31} ((I_{dane})_j - (I_{num})_j)^2,$$

gdzie p to wektor parametrów do estymacji, I_{dane} to odnotowane dane o przypadkach zarażonych, a I_{num} to rozwiązanie numeryczne dla grupy zarażonych I .

definiujemy funkcje BNK

par = **c**(b, g)

BNK <- **function**(**par**, h) {

 wyniki <- **c**()

I_0 = dane[1,2]

 N = 769454

S_0 = N - **I_0**

 m = 1/(15*365)

 TT = 1000

 sym <- simulation_mickens_2(**S_0**, **I_0**, h, b, g, m, TT, N)

 num <- sym[[2]][seq(1, length(sym[[2]]), h)]

for(i in 1:31){

 wyniki[i] <- (dane\$cases_1[i] - num[i])^2

```
    }  
    return(sum(wyniki))  
}
```

```
b = 0.79  
g = 0.7  
h = 5
```

```
BNK(par, h)
```

Wartość BNK dla warunków początkowych $\beta = 0.79$ i $\gamma = 0.7$ oraz $h=5$ wyszła nam 9046024. Zatem jak najbardziej jest potrzeba estymacji parametrów.

Kod

```
#####  
# Burczyk Aleksandra  
# Gmaj Monika  
# Jankowska Kinga  
# Langowski Kamil  
#  
#####  
# PROJEKT 2  
#####
```

```
#####  
# ZADANIE 2.1  
# Wyprowadzenie niestandardowej metody Mickensa  
# dla modelu SIR  
#  $dS(t) / dt = m * N - (b / N) * S(t) * I(t) - m * S(t)$   
#  $dI(t) / dt = (b / N) * S(t) * I(t) - g * I(t) - m * I(t)$   
#  $R(t) = N - S(t) - I(t)$   
#####
```

```
# implementacja metody mickensa  
# wyznaczone  
#  $S_{\{n+1\}} = S_{n1}$   
#  $I_{\{n+1\}} = I_{n1}$   
#  $R_{\{n+1\}} = R_{n1}$   
#  $S_{\{n\}} = S_n, I_{\{n\}} = I_n, R_{\{n\}} = R_n$ 
```

```
S_n1 <- function(h, m, N, Sn, b, In) return((h*m*N+Sn)/(1+h*m+(b/N)*h*In))
```

```
I_n1 <- function(b, N, h, S_n1, In, g, m) return(((b/N)*h*S_n1*In + In)/(1+g*h+m*h))
```

```
R_n1 <- function(g, h, I_n1, Rn, m) return((g*h*I_n1+Rn)/(1+m*h))
```

```
# funkcja dokonujaca symulacji metody numerycznej mickensa  
simulation_mickens <- function(S_0, I_0, R_0, h, b,g, m, TT, N) {  
  n = TT / h
```

```
  S={}  
  S[1] <- S_0
```

```
  I={}  
  I[1] <- I_0
```

```
  R={}  
  R[1] <- R_0
```

```
  for(i in 2:n) {  
    S_n1 = S_n1(h, m, N, S[i-1], b, I[i-1])  
    S[i] = S_n1
```

```
    I_n1 = I_n1(b, N, h, S[i], I[i-1], g, m)  
    I[i] = I_n1
```

```
    R_n1 = R_n1(g, h, I[i], R[i-1], m)
```

```

    R[i] = R_n_1
  }
  return(list(S,I,R))
}

```

```

# funkcja wyznaczajaca wspolczynnik R0
# zgodnie ze wzorem
#  $R_0 = b / (m + g)$ 
R_0 <- function(b, g, m) return( b / (g + m))

```

```

# funkcja wyznaczajaca punkt rownowagi modelu
pkt_S <- function(N, R_0) return (N/R_0)

```

```

pkt_I <- function(N, m, b, R_0) return ((N * m * (R_0 - 1)) / (b))

```

```

pkt_R <- function(N, g, b, R_0) return ((N * g * (R_0 - 1)) / (b))

```

```

#####
# ZADANIE 2.2 a
# Wyprobujemy 4 zestawy parametrow, zaczynajac od a
# a)  $(S_0, I_0, b, g, m) = (N - 1, 1, 0.5, 0.8, 0.4)$ ,
#####

```

```

# - wyznaczymy punkt rownowagi oraz wartosc R0

```

```

#   wyznaczamy wspolczynnik R_0
#   parametry:  $b = 0.5, g = 0.8, m = 0.4$ 
R0_a = R_0(b = 0.5, g = 0.8, m= 0.4)
R0_a

```

```

#   wyznaczamy punkt rownowagi
#   parametry  $b = 0.5, g = 0.8, m = 0.4, N = 10000$ 
rownowagaSIR_a <- c(pkt_S(N = 10000, R0_a), pkt_I(N = 10000, b = 0.5, m= 0.4, R0_a),
  pkt_R(N = 10000, b = 0.5, g = 0.8, R0_a))
rownowagaSIR_a

```

```

# - przedstawimy na jednym wykresie rozwiazanie numeryczne modelu  $(S, I, R)$ 
#   rozwiazanie numeryczne modelu SIR
x_a = simulation_mickens(9999, 1, 0, h = 0.1, b = 0.5, g = 0.8, m = 0.4, TT = 1000, N = 10000)
S_a = x_a[[1]]
I_a = x_a[[2]]
R_a = x_a[[3]]
df_a = data.frame(S_a, I_a, R_a)

```

```

plot(c(1:length(S_a)), S_a, main = "wykres_S_w_czasie", type = "l", col = "skyblue3",
  xlab = "czas_", ylab = "liczba_osob")
legend("right", c("susceptibles"), col = c("skyblue3"), lty = 1, bty = "n")

```

```

plot(c(1:length(I_a)), I_a, main = "wykres_I_w_czasie", type = "l", col = "palevioletred3",
  xlab = "czas_", ylab = "liczba_osob")
legend("right", c("infectious"), col = c("palevioletred3"), lty = 1, bty = "n")

```

```

plot(c(1:length(R_a)), R_a, main = "wykres_R_w_czasie", type = "l", col = "yellowgreen",
  xlab = "czas_", ylab = "liczba_osob")
legend("right", c("recovered"), col = c("yellowgreen"), lty = 1, bty = "n")

```

```
plot(c(1:length(R_a)), R_a, main = "wykres_SIR", type = "l", col = "yellowgreen",
     xlab = "czas_", ylab = "liczba_osob")
lines(c(1:length(I_a)), I_a, type = "l", col = "palevioletred3")
lines(c(1:length(S_a)), S_a, type = "l", col = "skyblue3")
legend("right", c("susceptibles", "infectious", "recovered"),
     col = c("skyblue3", "palevioletred3", "yellowgreen"), lty = 1, bty = "n")
```

```
# - wyznaczmy portrety fazowe dla par  $I$  -  $S$ ,  $S$  -  $R$ ,  $I$  -  $R$ 
# wraz z zaznaczonymi punktami początkowymi oraz punktami równowagi
```

```
#  $I$  -  $S$ 
plot(I_a, S_a, main = "~I~S", type = "l")
points(rownowagaSIR_a[2], rownowagaSIR_a[1], pch = 16, col = "darkorchid1")
points(I_a[1], y = S_a[1], pch = 16, col = "coral2")
legend("topright", c("pkt_rownowagi", "pkt_początkowy"),
     col = c("darkorchid1", "coral2"), cex = 0.5, pch = c(16, 16))
```

```
#  $S$  -  $R$ 
plot(S_a, R_a, main = "~S~R", type = "l")
points(rownowagaSIR_a[1], rownowagaSIR_a[3], pch = 16, col = "darkorchid1")
points(S_a[1], y = R_a[1], pch = 16, col = "coral2")
legend("topright", c("pkt_rownowagi", "pkt_początkowy"),
     col = c("darkorchid1", "coral2"), cex = 0.5, pch = c(16, 16))
```

```
#  $I$  -  $R$ 
plot(I_a, R_a, main = "~I~R", type = "l")
points(rownowagaSIR_a[2], rownowagaSIR_a[3], pch = 16, col = "darkorchid1")
points(I_a[1], y = R_a[1], pch = 16, col = "coral2")
legend("topright", c("pkt_rownowagi", "pkt_początkowy"),
     col = c("darkorchid1", "coral2"), cex = 0.5, pch = c(16, 16))
```

```
# - wyznaczmy trójwymiarowy portret fazowy
#  $S$  -  $I$  -  $R$ 
# install.packages("plotly")
library("plotly")
plot_ly(df_a, x = df_a$S_a, y = df_a$I_a, z = df_a$R_a, main = "~S~I~R",
mode = 'lines', opacity = 1)
```

```
#####
# ZADANIE 2.2 b
# b)  $(S_0, I_0, b, g, m) = (N - 1, 1, 0.2, 0.04, 0.01)$ 
#####
```

```
# - wyznaczmy punkt równowagi oraz wartość  $R_0$ 
```

```
# wyznaczamy współczynnik  $R_0$ 
# parametry:  $b = 0.2$ ,  $g = 0.04$ ,  $m = 0.01$ 
R0_b = R_0(b = 0.2, g = 0.04, m = 0.01)
R0_b
```

```
# wyznaczamy punkt równowagi
# parametry  $b = 0.2$ ,  $g = 0.04$ ,  $m = 0.01$ ,  $N = 10000$ 
```

```

rownowagaSIR_b <- c(pkt_S(N = 10000, R0_b), pkt_I(N = 10000, b = 0.2, m = 0.01, R0_b),
                    pkt_R(N = 10000, b = 0.2, g = 0.04, R0_b))
rownowagaSIR_b

# - przedstawimy na jednym wykresie rozwiązanie numeryczne modelu (S, I, R)
# rozwiązanie numeryczne modelu SIR
x_b = simulation_mickens(9999, 1, 0, h = 0.1, b = 0.2, g = 0.04, m = 0.01, TT = 1000, N = 10000)
S_b = x_b[[1]]
I_b = x_b[[2]]
R_b = x_b[[3]]
df_b = data.frame(S_b, I_b, R_b)

plot(c(1:length(S_b)), S_b, main = "wykres_S_w_czasie", type = "l", col = "skyblue3",
      xlab = "czas_", ylab = "liczba_osob")
legend("right", c("susceptibles"), col = c("skyblue3"), lty = 1, bty = "n")

plot(c(1:length(I_b)), I_b, main = "wykres_I_w_czasie", type = "l", col = "palevioletred3",
      xlab = "czas_", ylab = "liczba_osob")
legend("right", c("infectious"), col = c("palevioletred3"), lty = 1, bty = "n")

plot(c(1:length(R_b)), R_b, main = "wykres_R_w_czasie", type = "l", col = "yellowgreen",
      xlab = "czas_", ylab = "liczba_osob")
legend("right", c("recovered"), col = c("yellowgreen"), lty = 1, bty = "n")

plot(c(1:length(S_b)), S_b, main = "wykres_S_I_R", type = "l", col = "skyblue3",
      xlab = "czas_", ylab = "liczba_osob")
lines(c(1:length(I_b)), I_b, type = "l", col = "palevioletred3")
lines(c(1:length(R_b)), R_b, type = "l", col = "yellowgreen")
legend("topright", c("susceptibles", "infectious", "recovered"),
      col = c("skyblue3", "palevioletred3", "yellowgreen"), lty = 1, bty = "n", cex=0.5)

# - wyznaczymy portrety fazowe dla par I      S, S      R, I      R
# wraz z zaznaczonymi punktami początkowymi oraz punktami rownowagi

# I      S
plot(I_b, S_b, main = "_I_~_S_", type = "l")
points(rownowagaSIR_b[2], rownowagaSIR_b[1], pch = 16, col = "darkorchid1")
points(I_b[1], y = S_b[1], pch = 16, col = "coral2")
legend("topright", c("pkt_rownowagi", "pkt_początkowy"),
      col = c("darkorchid1", "coral2"), cex=0.5, pch=c(16,16))

# S      R
plot(S_b, R_b, main = "_S_~_R_", type = "l")
points(rownowagaSIR_b[1], rownowagaSIR_b[3], pch = 16, col = "darkorchid1")
points(S_b[1], y = R_b[1], pch = 16, col = "coral2")
legend("topright", c("pkt_rownowagi", "pkt_początkowy"),
      col = c("darkorchid1", "coral2"), cex=0.5, pch=c(16,16))

# I      R
plot(I_b, R_b, main = "_I_~_R_", type = "l")
points(rownowagaSIR_b[2], rownowagaSIR_b[3], pch = 16, col = "darkorchid1")
points(I_b[1], y = R_b[1], pch = 16, col = "coral2")
legend("topright", c("pkt_rownowagi", "pkt_początkowy"),
      col = c("darkorchid1", "coral2"), cex=0.5, pch=c(16,16))

# - wyznaczymy trójwymiarowy portret fazowy
# S      I      R

```

```

# install.packages("plotly")
library("plotly")
plot_ly(df_b, x=df_b$S_b, y=~df_b$I_b, z=~df_b$R_b, main = "S_b I_b R_b",
mode = 'lines', opacity = 1)

#####
# ZADANIE 2.2 c
# c) ( $S_0$ ,  $I_0$ ,  $b$ ,  $g$ ,  $m$ ) = ( $N - 1$ ,  $1$ ,  $0.8$ ,  $0.4$ ,  $0.01$ )
#####

# - wyznaczmy punkt równowagi oraz wartość  $R_0$ 

# wyznaczamy współczynnik  $R_0$ 
# parametry:  $b = 0.8$ ,  $g = 0.4$ ,  $m = 0.01$ 
R0_c = R_0(b = 0.8, g = 0.4, m = 0.01)
R0_c

# wyznaczamy punkt równowagi
# parametry  $b = 0.2$ ,  $g = 0.04$ ,  $m = 0.01$ ,  $N = 10000$ 
rownowagaSIR_c <- c(pkt_S(N = 10000, R0_c), pkt_I(N = 10000, b = 0.8, m = 0.01, R0_c),
pkt_R(N = 10000, b = 0.8, g = 0.4, R0_c))
rownowagaSIR_c

# - przedstawimy na jednym wykresie rozwiązanie numeryczne modelu ( $S$ ,  $I$ ,  $R$ )
# rozwiązanie numeryczne modelu SIR
x_c = simulation_mickens(9999, 1, 0, h = 0.1, b = 0.8, g = 0.4, m = 0.01, TT = 1000, N = 10000)
S_c = x_c[[1]]
I_c = x_c[[2]]
R_c = x_c[[3]]
df_c = data.frame(S_c, I_c, R_c)

plot(c(1:length(S_c)), S_c, main = "wykres_S_w_czasie", type = "l", col = "skyblue3",
xlab = "czas_", ylab = "liczba_osob")
legend("right", c("susceptibles"), col = c("skyblue3"), lty = 1, bty = "n")

plot(c(1:length(I_c)), I_c, main = "wykres_I_w_czasie", type = "l", col = "palevioletred3",
xlab = "czas_", ylab = "liczba_osob")
legend("right", c("infectious"), col = c("palevioletred3"), lty = 1, bty = "n")

plot(c(1:length(R_c)), R_c, main = "wykres_R_w_czasie", type = "l", col = "yellowgreen",
xlab = "czas_", ylab = "liczba_osob")
legend("right", c("recovered"), col = c("yellowgreen"), lty = 1, bty = "n")

plot(c(1:length(R_c)), R_c, main = "wykres_SIR", type = "l", col = "yellowgreen",
xlab = "czas_", ylab = "liczba_osob")
lines(c(1:length(I_c)), I_c, type = "l", col = "palevioletred3")
lines(c(1:length(S_c)), S_c, type = "l", col = "skyblue3")
legend("right", c("susceptibles", "infectious", "recovered"),
col = c("skyblue3", "palevioletred3", "yellowgreen"), lty = 1, bty = "n", cex=0.5)

# - wyznaczmy portrety fazowe dla par  $I$  vs  $S$ ,  $S$  vs  $R$ ,  $I$  vs  $R$ 
# wraz z zaznaczonymi punktami początkowymi oraz punktami równowagi

#  $I$  vs  $S$ 
plot(I_c, S_c, main = "I~S", type = "l")
points(rownowagaSIR_c[2], rownowagaSIR_c[1], pch = 16, col = "darkorchid1")
points(I_c[1], y = S_c[1], pch = 16, col = "coral2")

```

```

legend("topright", c("pkt_rownowagi", "pkt_poczatkowy"),
      col = c("darkorchid1", "coral2"), cex=0.5, pch=c(16,16))

# S      R
plot(S_c, R_c, main = "S~R", type = "l")
points(rownowagaSIR_c[1], rownowagaSIR_c[3], pch = 16, col = "darkorchid1")
points(S_c[1], y = R_c[1], pch = 16, col = "coral2")
legend("topright", c("pkt_rownowagi", "pkt_poczatkowy"),
      col = c("darkorchid1", "coral2"), cex=0.5, pch=c(16,16))

# I      R
plot(I_c, R_c, main = "I~R", type = "l")
points(rownowagaSIR_c[2], rownowagaSIR_c[3], pch = 16, col = "darkorchid1")
points(I_c[1], y = R_c[1], pch = 16, col = "coral2")
legend("topright", c("pkt_rownowagi", "pkt_poczatkowy"),
      col = c("darkorchid1", "coral2"), cex=0.5, pch=c(16,16))

# - wyznaczymy trojwymiarowy portret fazowy
# S      I      R
# install.packages("plotly")
library("plotly")
plot_ly(df_c, x=df_c$S_c, y=df_c$I_c, z=df_c$R_c, main = "S~I~R",
mode = 'lines', opacity = 1)

#####
# ZADANIE 2.2 d
# d)  $(S_0, I_0, b, g, m) = (N - 1, 1, 0.8, 0.7, 0.01)$ 
#####

# - wyznaczymy punkt rownowagi oraz wartosc R0

#   wyznaczamy wspolczynnik R0
#   parametry:  $b = 0.8, g = 0.7, m = 0.01$ 
R0_d = R_0(b = 0.8, g = 0.7, m = 0.01)
R0_d

#   wyznaczamy punkt rownowagi
#   parametry  $b = 0.2, g = 0.04, m = 0.01, N = 10000$ 
rownowagaSIR_d <- c(pkt_S(N = 10000, R0_d), pkt_I(N = 10000, b = 0.8, m = 0.01, R0_d),
                  pkt_R(N = 10000, b = 0.8, g = 0.7, R0_d))
rownowagaSIR_d

# - przedstawimy na jednym wykresie rozwiazanie numeryczne modelu (S, I, R)
#   rozwiazanie numeryczne modelu SIR
x_d = simulation_mickens(9999, 1, 0, h = 0.1, b = 0.8, g = 0.7, m = 0.01, TT = 1000, N = 10000)
S_d = x_d[[1]]
I_d = x_d[[2]]
R_d = x_d[[3]]
df_d = data.frame(S_d, I_d, R_d)

plot(c(1:length(S_d)), S_d, main = "wykres_S_w_czasie", type = "l", col = "skyblue3",
     xlab = "czas_", ylab = "liczba_osob")
legend("right", c("susceptibles"), col = c("skyblue3"), lty = 1, bty = "n")

plot(c(1:length(I_d)), I_d, main = "wykres_I_w_czasie", type = "l", col = "palevioletred3",
     xlab = "czas_", ylab = "liczba_osob")

```

```

legend("right", c("infectious"), col = c("palevioletred3"), lty = 1, bty = "n")

plot(c(1:length(R_d)), R_d, main = "wykres_R_w_czasie", type = "l", col = "yellowgreen",
xlab = "czas_", ylab = "liczba_osob")
legend("right", c("recovered"), col = c("yellowgreen"), lty = 1, bty = "n")

plot(c(1:length(R_d)), R_d, main = "wykres_SIR", type = "l", col = "yellowgreen",
xlab = "czas_", ylab = "liczba_osob")
lines(c(1:length(I_d)), I_d, type = "l", col = "palevioletred3")
lines(c(1:length(S_d)), S_d, type = "l", col = "skyblue3")
legend("right", c("susceptibles", "infectious", "recovered"),
col = c("skyblue3", "palevioletred3", "yellowgreen"), lty = 1, bty = "n")

# - wyznaczymy portrety fazowe dla par I S, S R, I R
# wraz z zaznaczonymi punktami początkowymi oraz punktami równowagi

# I S
plot(I_d, S_d, main = "~I~S", type = "l")
points(rownowagaSIR_d[2], rownowagaSIR_d[1], pch = 16, col = "darkorchid1")
points(I_d[1], y = S_d[1], pch = 16, col = "coral2")
legend("topright", c("pkt_rownowagi", "pkt_początkowy"),
col = c("darkorchid1", "coral2"), cex=0.5, pch=c(16,16))

# S R
plot(S_d, R_d, main = "~S~R", type = "l")
points(rownowagaSIR_d[1], rownowagaSIR_d[3], pch = 16, col = "darkorchid1")
points(S_d[1], y = R_d[1], pch = 16, col = "coral2")
legend("topright", c("pkt_rownowagi", "pkt_początkowy"),
col = c("darkorchid1", "coral2"), cex=0.5, pch=c(16,16))

# I R
plot(I_d, R_d, main = "~I~R", type = "l")
points(rownowagaSIR_d[2], rownowagaSIR_d[3], pch = 16, col = "darkorchid1")
points(I_d[1], y = R_d[1], pch = 16, col = "coral2")
legend("topright", c("pkt_rownowagi", "pkt_początkowy"),
col = c("darkorchid1", "coral2"), cex=0.5, pch=c(16,16))

# - wyznaczymy trójwymiarowy portret fazowy
# S I R
# install.packages("plotly")
library("plotly")
plot_ly(df_d, x=df_d$S_d, y=~df_d$I_d, z=~df_d$R_d, main = "~S~I~R",
mode = 'lines', opacity = 1)

#####
# ZADANIE 3
#
#####

# dla podpunktu b) z poprzedniego zadania
# parametry: b = 0.2, g = 0.04, m = 0.01

# S_0 = 5000

```

```

# I_0 = 5000
x_b3v1 = simulation_mickens(5000, 5000, 0, h = 0.1,
b = 0.2, g = 0.04, m = 0.01, TT = 1000, N = 10000)
S_b3v1 = x_b3v1[[1]]
I_b3v1 = x_b3v1[[2]]
R_b3v1 = x_b3v1[[3]]

# S_0 = 7000
# I_0 = 3000
x_b3v2 = simulation_mickens(7000, 3000, 0, h = 0.1,
b = 0.2, g = 0.04, m = 0.01, TT = 1000, N = 10000)
S_b3v2 = x_b3v2[[1]]
I_b3v2 = x_b3v2[[2]]
R_b3v1 = x_b3v2[[3]]

# S_0 = 1200
# I_0 = 8800
x_b3v3 = simulation_mickens(1200,8800, 0, h = 0.1,
b = 0.2, g = 0.04, m = 0.01, TT = 1000, N = 10000)
S_b3v3 = x_b3v3[[1]]
I_b3v3 = x_b3v3[[2]]
R_b3v3 = x_b3v3[[3]]

# S_0 = 6729
# I_0 = 3271
x_b3v4 = simulation_mickens(6729, 3271, 0, h = 0.1,
b = 0.2, g = 0.04, m = 0.01, TT = 1000, N = 10000)
S_b3v4 = x_b3v4[[1]]
I_b3v4 = x_b3v4[[2]]
R_b3v4 = x_b3v4[[3]]

# S_0 = 9900
# I_0 = 100
x_b3v5 = simulation_mickens(9900, 100, 0, h = 0.1,
b = 0.2, g = 0.04, m = 0.01, TT = 1000, N = 10000)
S_b3v5 = x_b3v5[[1]]
I_b3v5 = x_b3v5[[2]]
R_b3v5 = x_b3v5[[3]]

# org, 1, 3, 4, 5
plot(c(1:length(I_b)), I_b, main = "wykres_I_(zainfekowani)_dla_roznych_warunkow_poczatkowych",
type = "l", col = "palevioletred3", xlab = "czas", ylab = "liczba_osob")
lines(c(1:length(I_b)),I_b3v1, col="cadetblue")
lines(c(1:length(I_b)),I_b3v3, col="darksalmon")
lines(c(1:length(I_b)),I_b3v4, col="mediumpurple")
lines(c(1:length(I_b)),I_b3v5, col="steelblue")
legend("topright", c("S_0=9999,I_0=1",
"S_0=5000,I_0=5000", "S_0=1200,I_0=8800",
"S_0=6729,I_0=3271", "S_0=9900,I_0=100")
, col = c("palevioletred3", "cadetblue","darksalmon","mediumpurple",
"steelblue" ),
lty = 1, bty = "n", cex=0.5)

```

```

# I S
plot(I_b, S_b, main = "I~S", type = "l", col="palevioletred3", xlim = c(0,10000))
lines(I_b3v1, S_b3v1, col = "cadetblue")
lines(I_b3v3, S_b3v3, col = "darksalmon")
lines(I_b3v4, S_b3v4, col = "mediumpurple")
lines(I_b3v5, S_b3v5, col = "steelblue")
points(rownowagaSIR_b[2], rownowagaSIR_b[1], pch = 16, col = "slategray1")
points(I_b[1], y = S_b[1], pch = 16, col = "palevioletred3")
points(I_b3v1[1], y = S_b3v1[1], pch = 16, col = "cadetblue")
points(I_b3v3[1], y = S_b3v3[1], pch = 16, col = "darksalmon")
points(I_b3v4[1], y = S_b3v4[1], pch = 16, col = "mediumpurple")
points(I_b3v5[1], y = S_b3v5[1], pch = 16, col = "steelblue")
legend("topright", c("pkt_rownowagi", "S_0=9999, I_0=1",
                    "S_0=5000, I_0=5000", "S_0=1200, I_0=8800",
                    "S_0=6729, I_0=3271", "S_0=9900, I_0=100"),
      , col = c("slategray1", "palevioletred3", "cadetblue", "darksalmon", "mediumpurple",
                "steelblue" ),
      lty = 1, bty = "n", cex=0.5, pch=c(16,16), inset = -0.12)

```

```

# I R
plot(I_b, R_b, main = "I~R", type = "l", col="palevioletred3", xlim = c(0,10000))
lines(I_b3v1, R_b3v1, col = "cadetblue")
lines(I_b3v3, R_b3v3, col = "darksalmon")
lines(I_b3v4, R_b3v4, col = "mediumpurple")
lines(I_b3v5, R_b3v5, col = "steelblue")
points(rownowagaSIR_b[2], rownowagaSIR_b[3], pch = 16, col = "slategray1")
points(I_b[1], y = R_b[1], pch = 16, col = "palevioletred3")
points(I_b3v1[1], y = R_b3v1[1], pch = 16, col = "cadetblue")
points(I_b3v3[1], y = R_b3v3[1], pch = 16, col = "darksalmon")
points(I_b3v4[1], y = R_b3v4[1], pch = 16, col = "mediumpurple")
points(I_b3v5[1], y = R_b3v5[1], pch = 16, col = "steelblue")
legend("topright", c("pkt_rownowagi", "S_0=9999, I_0=1",
                    "S_0=5000, I_0=5000", "S_0=1200, I_0=8800",
                    "S_0=6729, I_0=3271", "S_0=9900, I_0=100"),
      , col = c("slategray1", "palevioletred3", "cadetblue", "darksalmon", "mediumpurple",
                "steelblue" ),
      lty = 1, bty = "n", cex=0.5, pch=c(16,16), inset = -0.12)

```

```

# dla podpunktu c) z poprzedniego zadania
# parametry: b = 0.8, g = 0.4, m = 0.01

```

```

# S_0 = 5000
# I_0 = 5000
x_c3v1 = simulation_mickens(5000, 5000, 0, h = 0.1,
b = 0.8, g = 0.4, m = 0.01, TT = 1000, N = 10000)
S_c3v1 = x_c3v1[[1]]
I_c3v1 = x_c3v1[[2]]
R_c3v1 = x_c3v1[[3]]

```

```

# S_0 = 7000
# I_0 = 3000
x_c3v2 = simulation_mickens(7000, 3000, 0, h = 0.1,
b = 0.8, g = 0.4, m = 0.01, TT = 1000, N = 10000)
S_c3v2 = x_c3v2[[1]]
I_c3v2 = x_c3v2[[2]]
R_c3v1 = x_c3v2[[3]]

# S_0 = 1200
# I_0 = 8800
x_c3v3 = simulation_mickens(1200,8800, 0, h = 0.1,
b = 0.8, g = 0.4, m = 0.01, TT = 1000, N = 10000)
S_c3v3 = x_c3v3[[1]]
I_c3v3 = x_c3v3[[2]]
R_c3v3 = x_c3v3[[3]]

# S_0 = 6729
# I_0 = 3271
x_c3v4 = simulation_mickens(6729, 3271, 0, h = 0.1,
b = 0.8, g = 0.4, m = 0.01, TT = 1000, N = 10000)
S_c3v4 = x_c3v4[[1]]
I_c3v4 = x_c3v4[[2]]
R_c3v4 = x_c3v4[[3]]

# S_0 = 9900
# I_0 = 100
x_c3v5 = simulation_mickens(9900, 100, 0, h = 0.1,
b = 0.8, g = 0.4, m = 0.01, TT = 1000, N = 10000)
S_c3v5 = x_c3v5[[1]]
I_c3v5 = x_c3v5[[2]]
R_c3v5 = x_c3v5[[3]]

# org, 1, 3, 4, 5
plot(c(1:length(I_c)), I_c, main = "wykres_I_(zainfekowani)_dla_roznych_warunkow_poczatkowych",
type = "l", col = "palevioletred3", xlab = "czas", ylab = "liczba_osob")
lines(c(1:length(I_c)), I_c3v1, col="cadetblue")
lines(c(1:length(I_c)), I_c3v3, col="darksalmon")
lines(c(1:length(I_c)), I_c3v4, col="mediumpurple")
lines(c(1:length(I_c)), I_c3v5, col="steelblue")
legend("topright", c("S_0=9999, I_0=1",
                    "S_0=5000, I_0=5000", "S_0=1200, I_0=8800",
                    "S_0=6729, I_0=3271", "S_0=9900, I_0=100"),
, col = c("palevioletred3", "cadetblue", "darksalmon", "mediumpurple",
          "steelblue" ),
lty = 1, bty = "n", cex=0.5)

# I S
plot(I_c, S_c, main = "I~S", type = "l", col="palevioletred3",
xlim = c(0,10000), ylim = c(0,10000))
lines(I_c3v1, S_c3v1, col = "cadetblue")
lines(I_c3v3, S_c3v3, col = "darksalmon")
lines(I_c3v4, S_c3v4, col = "mediumpurple")
lines(I_c3v5, S_c3v5, col = "steelblue")
points(rownowagaSIR_c[2], rownowagaSIR_c[1], pch = 16, col = "slategray1")

```

```

points(I_c[1], y = S_c[1], pch = 16, col = "palevioletred3")
points(I_c3v1[1], y = S_c3v1[1], pch = 16, col = "cadetblue")
points(I_c3v3[1], y = S_c3v3[1], pch = 16, col = "darksalmon")
points(I_c3v4[1], y = S_c3v4[1], pch = 16, col = "mediumpurple")
points(I_c3v5[1], y = S_c3v5[1], pch = 16, col = "steelblue")
legend("topright", c("pkt_rownowagi", "S_0=9999, I_0=1",
                    "S_0=5000, I_0=5000", "S_0=1200, I_0=8800",
                    "S_0=6729, I_0=3271", "S_0=9900, I_0=100"),
      , col = c("slategray1", "palevioletred3", "cadetblue", "darksalmon", "mediumpurple",
                "steelblue" ),
      lty = 1, bty = "n", cex=0.5, pch=c(16,16), inset = -0.12)

```

```

# I R
plot(I_c, R_c, main = "I~R", type = "l", col="palevioletred3",
xlim = c(0,10000), ylim = c(0,10000))
lines(I_c3v1, R_c3v1, col = "cadetblue")
lines(I_c3v3, R_c3v3, col = "darksalmon")
lines(I_c3v4, R_c3v4, col = "mediumpurple")
lines(I_c3v5, R_c3v5, col = "steelblue")
points(rownowagaSIR_c[2], rownowagaSIR_c[3], pch = 16, col = "slategray1")
points(I_c[1], y = R_c[1], pch = 16, col = "palevioletred3")
points(I_c3v1[1], y = R_c3v1[1], pch = 16, col = "cadetblue")
points(I_c3v3[1], y = R_c3v3[1], pch = 16, col = "darksalmon")
points(I_c3v4[1], y = R_c3v4[1], pch = 16, col = "mediumpurple")
points(I_c3v5[1], y = R_c3v5[1], pch = 16, col = "steelblue")
legend("topright", c("pkt_rownowagi", "S_0=9999, I_0=1",
                    "S_0=5000, I_0=5000", "S_0=1200, I_0=8800",
                    "S_0=6729, I_0=3271", "S_0=9900, I_0=100"),
      , col = c("slategray1", "palevioletred3", "cadetblue", "darksalmon", "mediumpurple",
                "steelblue" ),
      lty = 1, bty = "n", cex=0.5, pch=c(16,16), inset = -0.12)

```

```

# dla podpunktu d) z poprzedniego zadania
# parametry: b = 0.8, g = 0.7, m = 0.01

```

```

# S_0 = 5000
# I_0 = 5000
x_d3v1 = simulation_mickens(5000, 5000, 0, h = 0.1,
b = 0.8, g = 0.7, m = 0.01, TT = 1000, N = 10000)
S_d3v1 = x_d3v1[[1]]
I_d3v1 = x_d3v1[[2]]
R_d3v1 = x_d3v1[[3]]

```

```

# S_0 = 7000
# I_0 = 3000
x_d3v2 = simulation_mickens(7000, 3000, 0, h = 0.1,
b = 0.8, g = 0.7, m = 0.01, TT = 1000, N = 10000)
S_d3v2 = x_d3v2[[1]]
I_d3v2 = x_d3v2[[2]]
R_d3v1 = x_d3v2[[3]]

```

```

# S_0 = 1200
# I_0 = 8800
x_d3v3 = simulation_mickens(1200, 8800, 0, h = 0.1,
b = 0.8, g = 0.7, m = 0.01, TT = 1000, N = 10000)

```

```

S_d3v3 = x_d3v3[[1]]
I_d3v3 = x_d3v3[[2]]
R_d3v3 = x_d3v3[[3]]

```

```

# S_0 = 6729
# I_0 = 3271
x_d3v4 = simulation_mickens(6729, 3271, 0, h = 0.1,
b = 0.8, g = 0.7, m = 0.01, TT = 1000, N = 10000)
S_d3v4 = x_d3v4[[1]]
I_d3v4 = x_d3v4[[2]]
R_d3v4 = x_d3v4[[3]]

```

```

# S_0 = 9900
# I_0 = 100
x_d3v5 = simulation_mickens(9900, 100, 0, h = 0.1,
b = 0.8, g = 0.7, m = 0.01, TT = 1000, N = 10000)
S_d3v5 = x_d3v5[[1]]
I_d3v5 = x_d3v5[[2]]
R_d3v5 = x_d3v5[[3]]

```

```

# org, 1, 3, 4, 5
plot(c(1:length(I_d)), I_d, main = "wykres_I_(zainfekowani)_dla_roznych_warunkow_poczatkowych",
type = "l", col = "palevioletred3", xlab = "czas", ylab = "liczba_osob")
lines(c(1:length(I_d)), I_d3v1, col="cadetblue")
lines(c(1:length(I_d)), I_d3v3, col="darksalmon")
lines(c(1:length(I_d)), I_d3v4, col="mediumpurple")
lines(c(1:length(I_d)), I_d3v5, col="steelblue")
legend("topright", c("S_0=9999, I_0=1",
                    "S_0=5000, I_0=5000", "S_0=1200, I_0=8800",
                    "S_0=6729, I_0=3271", "S_0=9900, I_0=100"),
, col = c("palevioletred3", "cadetblue", "darksalmon", "mediumpurple",
          "steelblue"),
lty = 1, bty = "n", cex=0.5, pch=c(16,16), inset = -0.12)

```

```

# I S
plot(I_d, S_d, main = "I~S", type = "l", col="palevioletred3",
xlim = c(0,2000), ylim= c(5000,10000))
lines(I_d3v1, S_d3v1, col = "cadetblue")
lines(I_d3v3, S_d3v3, col = "darksalmon")
lines(I_d3v4, S_d3v4, col = "mediumpurple")
lines(I_d3v5, S_d3v5, col = "steelblue")
points(rownowagaSIR_d[2], rownowagaSIR_d[1], pch = 16, col = "slategray1")
points(I_d[1], y = S_d[1], pch = 16, col = "palevioletred3")
points(I_d3v1[1], y = S_d3v1[1], pch = 16, col = "cadetblue")
points(I_d3v3[1], y = S_d3v3[1], pch = 16, col = "darksalmon")
points(I_d3v4[1], y = S_d3v4[1], pch = 16, col = "mediumpurple")
points(I_d3v5[1], y = S_d3v5[1], pch = 16, col = "steelblue")
legend("topright", c("pkt_rownowagi", "S_0=9999, I_0=1",
                    "S_0=5000, I_0=5000", "S_0=1200, I_0=8800",
                    "S_0=6729, I_0=3271", "S_0=9900, I_0=100"),
, col = c("slategray1", "palevioletred3", "cadetblue", "darksalmon", "mediumpurple",
          "steelblue"),
lty = 1, bty = "n", cex=0.5, pch=c(16,16), inset = -0.12)

```

```

# I R
plot(I_d, R_d, main = "I~R", type = "l", col="palevioletred3",
xlim = c(0,10000), ylim=c(0,10000) )
lines(I_d3v1, R_d3v1, col = "cadetblue")
lines(I_d3v3, R_d3v3, col = "darksalmon")
lines(I_d3v4, R_d3v4, col = "mediumpurple")
lines(I_d3v5, R_d3v5, col = "steelblue")
points(rownowagaSIR_d[2], rownowagaSIR_d[3], pch = 16, col = "slategray1")
points(I_d[1], y = R_d[1], pch = 16, col = "palevioletred3")
points(I_d3v1[1], y = R_d3v1[1], pch = 16, col = "cadetblue")
points(I_d3v3[1], y = R_d3v3[1], pch = 16, col = "darksalmon")
points(I_d3v4[1], y = R_d3v4[1], pch = 16, col = "mediumpurple")
points(I_d3v5[1], y = R_d3v5[1], pch = 16, col = "steelblue")
legend("topright", c("pkt_rownowagi", "S_0=9999, I_0=1",
                     "S_0=5000, I_0=5000", "S_0=1200, I_0=8800",
                     "S_0=6729, I_0=3271", "S_0=9900, I_0=100"),
      , col = c("slategray1", "palevioletred3", "cadetblue", "darksalmon", "mediumpurple",
                "steelblue" ),
      lty = 1, bty = "n", cex=0.5, pch=c(16,16), inset = -0.12)

```

```

#####
# ZADANIE 4
#
#####

```

```

# a przeskalowane
x_a4 = simulation_mickens(9999, 1, 0, h = 0.1,
b = 0.5 , g = 0.8, m = 0.4, TT = 1000, N = 10000)
S_a4 = x_a4[[1]]/10000
I_a4 = x_a4[[2]]/10000
R_a4 = x_a4[[3]]/10000

plot(c(1:length(S_a4)), S_a4, main = "wykres_S_I_R(przeskalowane)",
type = "l", col = "skyblue3", xlab = "czas", ylab = "liczba_osob", ylim = c(0,1.2))
lines(c(1:length(I_a4)), I_a4, type = "l", col = "palevioletred3")
lines(c(1:length(R_a4)), R_a4, type = "l", col = "yellowgreen")
legend("topright", c("susceptibles", "infectious", "recovered"),
col = c("skyblue3", "palevioletred3", "yellowgreen"), lty = 1, bty = "n", cex=0.5)

```

```

# b przeskalowane
x_b4 = simulation_mickens(9999, 1, 0, h = 0.1,
b = 0.2, g = 0.04, m = 0.01, TT = 1000, N = 10000)
S_b4 = x_b4[[1]]/10000
I_b4 = x_b4[[2]]/10000
R_b4 = x_b4[[3]]/10000

plot(c(1:length(S_b4)), S_b4, main = "wykres_S_I_R(przeskalowane)",
type = "l", col = "skyblue3", xlab = "czas", ylab = "liczba_osob", ylim = c(0,1.2))
lines(c(1:length(I_b4)), I_b4, type = "l", col = "palevioletred3")
lines(c(1:length(R_b4)), R_b4, type = "l", col = "yellowgreen")
legend("topright", c("susceptibles", "infectious", "recovered"),
col = c("skyblue3", "palevioletred3", "yellowgreen"), lty = 1, bty = "n", cex=0.5)

```

```

# c przeskalowane

```

```

x_c4 = simulation_mickens(9999, 1, 0, h = 0.1,
b = 0.8, g = 0.4, m = 0.01, TT = 1000, N = 10000)
S_c4 = x_c4[[1]]/10000
I_c4 = x_c4[[2]]/10000
R_c4 = x_c4[[3]]/10000

```

```

plot(c(1:length(S_c4)), S_c4, main = "wykres_S_I_R_(przeskalowane)",
type = "l", col = "skyblue3", xlab = "czas_", ylab = "liczba_osob", ylim = c(0,1.2))
lines(c(1:length(I_c4)), I_c4, type = "l", col = "palevioletred3")
lines(c(1:length(R_c4)), R_c4, type = "l", col = "yellowgreen")
legend("topright", c("susceptibles", "infectious", "recovered"),
col = c("skyblue3", "palevioletred3", "yellowgreen"), lty = 1, bty = "n", cex=0.5)

```

```

# d przeskalowane
x_d4 = simulation_mickens(9999, 1, 0, h = 0.1,
b = 0.8, g = 0.7, m = 0.01, TT = 1000, N = 10000)
S_d4 = x_d4[[1]]/10000
I_d4 = x_d4[[2]]/10000
R_d4 = x_d4[[3]]/10000

```

```

plot(c(1:length(S_d4)), S_d4, main = "wykres_S_I_R_(przeskalowane)",
type = "l", col = "skyblue3", xlab = "czas_", ylab = "liczba_osob", ylim = c(0,1.2))
lines(c(1:length(I_d4)), I_d4, type = "l", col = "palevioletred3")
lines(c(1:length(R_d4)), R_d4, type = "l", col = "yellowgreen")
legend("topright", c("susceptibles", "infectious", "recovered"),
col = c("skyblue3", "palevioletred3", "yellowgreen"), lty = 1, bty = "n", cex=0.5)

```

```

# b przeskalowane dla roznych liczebnosci populacji
# liczebnosc = 902

```

```

x_b4v1 = simulation_mickens(901, 1, 0, h = 0.1,
b = 0.2, g = 0.04, m = 0.01, TT = 1000, N = 902)
S_b4v1 = x_b4v1[[1]]/902
I_b4v1 = x_b4v1[[2]]/902
R_b4v1 = x_b4v1[[3]]/902

```

```

plot(c(1:length(S_b4v1)), S_b4v1, main = "wykres_S_I_R_(przeskalowane),_liczebnosc_=_902",
type = "l", col = "skyblue3", xlab = "czas_", ylab = "liczba_osob", ylim = c(0,1.2))
lines(c(1:length(I_b4v1)), I_b4v1, type = "l", col = "palevioletred3")
lines(c(1:length(R_b4v1)), R_b4v1, type = "l", col = "yellowgreen")
legend("topright", c("susceptibles", "infectious", "recovered"),
col = c("skyblue3", "palevioletred3", "yellowgreen"), lty = 1, bty = "n", cex=0.5)

```

```

# liczebnosc = 5000
x_b4v2 = simulation_mickens(4999, 1, 0, h = 0.1,
b = 0.2, g = 0.04, m = 0.01, TT = 1000, N = 5000)
S_b4v2 = x_b4v2[[1]]/5000
I_b4v2 = x_b4v2[[2]]/5000
R_b4v2 = x_b4v2[[3]]/5000

```

```

plot(c(1:length(S_b4v2)), S_b4v2, main = "wykres_S_I_R_(przeskalowane),_liczebnosc_=_5_000",
type = "l", col = "skyblue3", xlab = "czas_", ylab = "liczba_osob", ylim = c(0,1.2))
lines(c(1:length(I_b4v2)), I_b4v2, type = "l", col = "palevioletred3")
lines(c(1:length(R_b4v2)), R_b4v2, type = "l", col = "yellowgreen")
legend("topright", c("susceptibles", "infectious", "recovered"),
col = c("skyblue3", "palevioletred3", "yellowgreen"), lty = 1, bty = "n", cex=0.5)

```

```

# liczebność = 90 000
x_b4v3 = simulation_mickens(89999, 1, 0, h = 0.1,
b = 0.2, g = 0.04, m = 0.01, TT = 1000, N = 90000)
S_b4v3 = x_b4v3[[1]]/90000
I_b4v3 = x_b4v3[[2]]/90000
R_b4v3 = x_b4v3[[3]]/90000

plot(c(1:length(S_b4v3)), S_b4v3, main = "wykres_S_I_R_(przeskalowane),_liczebność_=_90_000",
type = "l", col = "skyblue3", xlab = "czas_", ylab = "liczba_osob", ylim = c(0,1.2))
lines(c(1:length(I_b4v3)), I_b4v3, type = "l", col = "palevioletred3")
lines(c(1:length(R_b4v3)), R_b4v3, type = "l", col = "yellowgreen")
legend("topright", c("susceptibles", "infectious", "recovered"),
col = c("skyblue3", "palevioletred3", "yellowgreen"), lty = 1, bty = "n", cex=0.5)

# liczebność = 1 000 000
x_b4v4 = simulation_mickens(999999, 1, 0, h = 0.1,
b = 0.2, g = 0.04, m = 0.01, TT = 1000, N = 1000000)
S_b4v4 = x_b4v4[[1]]/1000000
I_b4v4 = x_b4v4[[2]]/1000000
R_b4v4 = x_b4v4[[3]]/1000000

plot(c(1:length(S_b4v4)), S_b4v4, main = "wykres_S_I_R_(przeskalowane),_liczebność_=_1_000_000",
type = "l", col = "skyblue3", xlab = "czas_", ylab = "liczba_osob", ylim = c(0,1.2))
lines(c(1:length(I_b4v4)), I_b4v4, type = "l", col = "palevioletred3")
lines(c(1:length(R_b4v4)), R_b4v4, type = "l", col = "yellowgreen")
legend("topright", c("susceptibles", "infectious", "recovered"),
col = c("skyblue3", "palevioletred3", "yellowgreen"), lty = 1, bty = "n", cex=0.5)

#####
# ZADANIE 5
#
#####

# instalujemy pakiet
install.packages("epimdr")
library("epimdr")

#wyswietlamy zbior danych niamey
niamey

# sumujemy wartosci cases_1, cases_2, cases_3
suma <- (niamey['cases_1'] + niamey['cases_2'] +niamey['cases_3'])
suma

install.packages("dplyr")
library("dplyr")
# wyswietlamy numery tygodni (1-31)
tyg <- select(niamey, absweek)
tyg

# tworzymy tabelke zawierajaca numer tygodnia oraz liczbe zachorowan dla trzech dystrktow razem
dane <- cbind(tyg, suma)
dane
plot(dane, xlab='tydzien', ylab='liczba_zachorowan', main='Liczba_zachorowan_w_poszczegolnych_tygo

```

```
#####  
# ZADANIE 6
```

```
#
```

```
#####
```

```
# zgodnie ze wzorem
```

```
#  $R_0 = b / (m + g)$ 
```

```
m = 1/(15*365)
```

```
b = 0.79
```

```
g = 0.7
```

```
R_0_6 <- b / (m+g)
```

```
R_0_6
```

```
# ponownie przypominamy metode numeryczna mickensa
```

```
S_n1 <- function(h, m, N, Sn, b, In) return((h*m*N+Sn)/(1+h*m+(b/N)*h*In))
```

```
I_n1 <- function(b, N, h, S_n1, In, g, m) return((b/N)*h*S_n1*In + In)/(1+g*h+m*h))
```

```
R_n1 <- function(g, h, I_n1, Rn, m) return((g*h*I_n1+Rn)/(1+m*h))
```

```
# funkcja dokonujaca symulacji metody numerycznej mickensa
```

```
# tutaj zmieniamy R_0, niech bedzie liczone z paramterow b/(m+g), wczesniej bylo argumentem funkcji
```

```
simulation_mickens_2 <- function(S_0, I_0, h, b, g, m, TT, N) {
```

```
  n = TT / h
```

```
  S={}
```

```
  S[1] <- S_0
```

```
  I={}
```

```
  I[1] <- I_0
```

```
  R={}
```

```
  R[1] <- b/(m+g)
```

```
  for(i in 2:n) {  
    S_n_1 = S_n1(h, m, N, S[i-1], b, I[i-1])  
    S[i] = S_n_1
```

```
    I_n_1 = I_n1(b, N, h, S[i], I[i-1], g, m)  
    I[i] = I_n_1
```

```
    R_n_1 = R_n1(g, h, I[i], R[i-1], m)  
    R[i] = R_n_1
```

```
  }  
  return(list(S,I,R))  
}
```

```
# podajemy parametry i wartosci wejsciowe
```

```
b = 0.79
```

```
g = 0.7
```

```
m = 1/(15*365)
```

```
R_0 = b/(m+g)
```

```
N = 769454
```

```
I_0 = dane[1,2]
```

```
S_0 = N - I_0
```

```
h = 5
```

```
TT = 1000
```

```
symulacja_6 <- simulation_mickens_2(S_0, I_0, h, b, g, m, TT, N)
```

```
symulacja_6
```

```
# wybieramy z symulacji druga wartosc, odpowiadaja grupie zarazonych, przeliczamy to na tygodnie
numeryczne <- symulacja_6[[2]][seq(1,length(symulacja_6[[2]]),h)]
numeryczne
```

```
plot(numeryczne, xlab='tydzien', ylab='liczba_zachorowan', col='red')
points(dane)
```

```
#####
# ZADANIE 7
```

```
#
#####
# definiujemy funkcje BNK
```

```
par = c(b,g)
BNK <- function(par, h) {
  wyniki <- c()
  I_0 = dane[1,2]
  N = 769454
  S_0 = N - I_0
  m = 1/(15*365)
  TT = 1000
  sym <- simulation_mickens_2(S_0, I_0, h, b, g, m, TT, N)
  num <- sym[[2]][seq(1,length(sym[[2]]),h)]
  for(i in 1:31){
    wyniki[i] <- (dane$cases_1[i] - num[i])^2
  }
  return(sum(wyniki))
}
```

```
b = 0.79
g = 0.7
h = 5
```

```
BNK(par, h)
```

```
#####
```