

▲ Popis

Ke kontrole skladištních roztočů rodu *Acarus* se používá jejich přirozený predátor roztoč rodu *Cheyletus*. Úspěšná inokulace predátora se vyplatí pouze za podmínek, kdy je schopný se dále množit. Chceme zjistit, jestli lze skladištní roztoče rodu *Acarus* regulovat pomocí přirozeného nepřitele.

▲ Data

Parametry k simulačnímu modelu byly získány z těchto experimentů:

Nosná kapacita prostředí (K_H) a vnitřní míra populačního růstu (r_H) roztočů rodu *Acarus* byly odhadnuty sledováním jejich množení bez přítomnosti predátora a při konstantním množství dostupné potravy. Bylo zjištěno, že $K_H = 2000$ jedinců/kg a $r_H = 0.23$.

Přirozená mortalita roztočů rodu *Cheyletus* (d), jeho vnitřní míra populačního růstu (r_P) a jeho účinnost přeměny biomasy na potomstvo (f) byly zjištěny podobně jako v kap. 22 tak, že predátor byl sledován od vylíhnutí do vykladení při různých (konstantních) hustotách kořisti. Parametry byly odhadnuty na $d = 0.1$, $r_P = 0.15$ a $f = 0.14$.

Vyhledávací účinnost predátora (a) a čas zpracování kořisti (T_h) byly zjištěny postupem popsaným v kap. 21. Parametry byly odhadnuty na $a = 0.002$ a $T_h = 0.04$. Maximální míra predace, c , byla odhadnuta na 1.3 jedinců/den.

Oba druhy roztočů jsou partenogenetické, tudíž není potřeba zohlednit poměr pohlaví.

▲ Úkoly

- 1/ Odhadněte, kolik jedinců predátora (*Cheyletus*) by mělo být vypuštěno, pokud kořisti (*Acarus*) je 900 jedinců na jednotku plochy.
- 2/ Zjistěte, jestli je *Cheyletus* schopný snížit početnost roztočů rodu *Acarus* pod ekonomický práh, který byl stanoven na hodnotu 300 jedinců/kg? Celý proces nasimulujte po dobu 200 dnů. Počáteční početnost roztočů rodu *Acarus* je 900 jedinců.

▲ Řešení

1/ Pro odhad počtu predátorů pro vypuštění použijte vzorec podle Sabelis et al. (2002):

$$\frac{P}{H} > \frac{r_H - r_P}{c},$$

kde P a H jsou počty predátorů a kořisti. Úpravou dostanete:

$$P > \frac{H(r_H - r_P)}{c}$$

$$> 900 * (0.23 - 0.15) / 1.3$$

Počáteční počet predátorů by měl být větší než jedinců.

2/ Abyste využili všechny zjištěné hodnoty parametrů, použijte Rosenzweig-MacArthurův model, který obsahuje závislost na hustotě v populaci kořisti (H), funkční odpověď typu II v populaci kořisti a numerickou odpověď predátora (P) na množství zkonzumované kořisti. *Acarus* i *Cheyletus* se množí takřka neustále, proto je k simulaci vhodný systém diferenciálních rovnic:

$$\frac{dH}{dt} = r_H H \left(1 - \frac{H}{K_H} \right) - \frac{aH}{1 + aHT_h} P$$

$$\frac{dP}{dt} = f \frac{aH}{1 + aHT_h} P - dP$$

Jako počáteční hodnotu pro počet vypuštěných predátorů použijte odhad zjištěný výše. Simulaci proveďte Eulerovou metodou, což znamená upravit systém rovnic na diskrétní podobu s krátkým časovým krokem, řekněme $\Delta = 0.1$:

$$H_{i,t2} = H_{i,t1} + \Delta \cdot \frac{dH_i}{dt} \qquad P_{i,t2} = P_{i,t1} + \Delta \cdot \frac{dP_i}{dt}$$

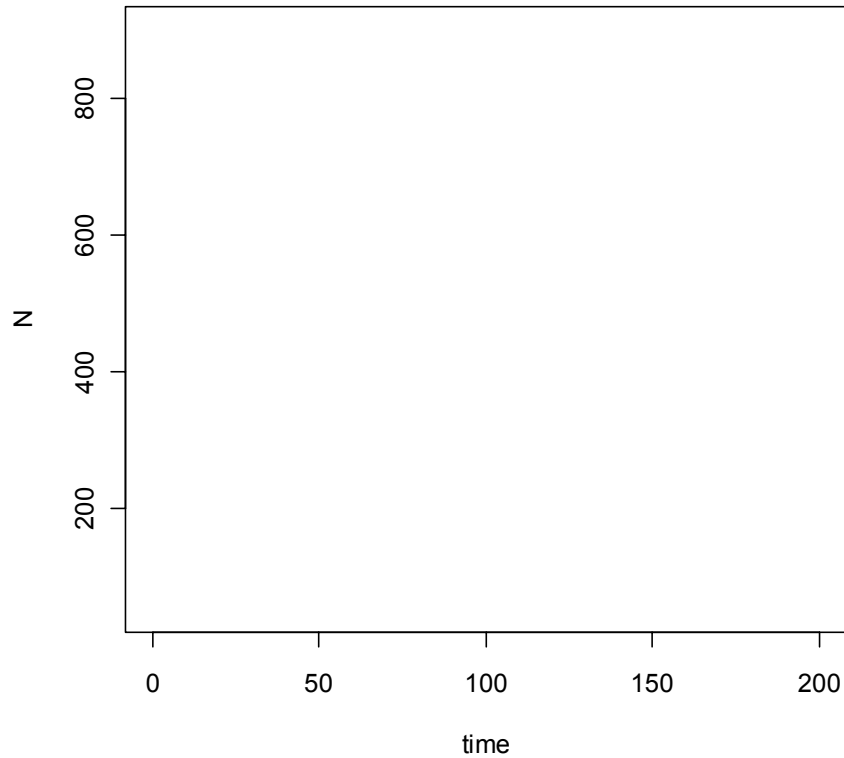
Do objektu `delta` vložte délku kroku. Ve vektoru `time` specifikujeme čas pro simulaci. Definujeme hodnoty všech parametrů a vytvoříme matici `N` se dvěma sloupci. Na jejich prvních pozicích budou počáteční početnosti. Simulaci proveďte příkazem `for` a výsledek vynesete příkazem `matplot`.

```
> delta <- 0.1
> time <- seq(0,200,by=delta)
> rH <- 0.23; KH <- 2000; d <- 0.1; f <- 0.14; a <- 0.002; Th <- 0.4
> N <- matrix(ncol=2,nrow=length(time))
> N[1,] <- c(900,60)
> for(i in 1:(length(time)-1)){
+   N[i+1,1] <- N[i,1]+delta*(rH*N[i,1]*(1-N[i,1]/KH)-
+   a*N[i,1]*(1/1+a*N[i,1]*Th)*N[i,2])
+   N[i+1,2] <- N[i,2]+delta*(f*a*N[i,1]*(1/1+a*N[i,1]*Th)*N[i,2]-d*N[i,2])}
```

```
> matplot(time,N,type="l",lty=1:2,col=1:2)
> legend("topright",c("Acarus","Cheyletus"),lty=1:2,col=1:2)
```

Překreslete graf do sešitu.

Simulace početnosti kořisti a predátorů



Podají se predátorům snížit početnost roztočů rodu *Acarus* pod požadovaný práh? Pokud ano, zjistěte pohledem do matice *N* za jak dlouho.

```
> N
> time[503]
```

.....

▲ Poznámka

Model predátor–kořist lze po úpravě použít i pro diskrétně se množící predátory (jednou za rok), kteří ale loví kořist takřka neustále. To se týká mnoha členovců, jako jsou pavouci nebo hmyz.

24

Model intraguildové predace

▲ Popis

V hypotetické skupině dravých prvoků jsou dva druhy predátorů, velcí a malí, kteří loví stejnou kořist. Je to systém asymetrické intraguildové predace, takže tělesně větší prvoci loví kromě kořisti také malé prvoky. Malí prvoci však nikdy neuloví větší. Zajímá nás, za jakých podmínek může být takový systém stabilní.

▲ Data

O celém systému máme pouze kusé informace. Víme, že populace kořisti (H) roste v závislosti na její hustotě s parametry $K_H = 200$ a $r_H = 1$. Velcí prvoci (L) mají poloviční efektivitu v lovu kořisti H než malí prvoci (S), tj. $a_{LH} = 0.5a_S$, přičemž $a_S = 0.04$. V lovu prvoka S je efektivita prvoka L poloviční ve srovnání s jeho lovem kořisti, tedy $a_{LS} = 0.5a_{LH}$. Ovšem uloveného prvoka S dokáže prvek L přeměnit na potomstvo dvakrát lépe než kořist, tedy $f_{LS} = 2f_{LH}$. Biomasu kořisti dokáže prvek S přeměnit na potomstvo dvakrát lépe než prvek L : $f_S = 2f_{LH}$, přičemž $f_S = 0.06$. Přírozená mortalita prvoků L (d_L) i S (d_S) je stejná, řekněme 0.1. Počáteční početnost prvoka S je dvojnásobkem početnosti L a početnost kořisti H je dvojnásobkem početnosti S .

▲ Úkoly

- 1/** Sestavte model popsaného systému. Vycházejte ze systému diferenciálních rovnic Lotky a Volterry pro predátora a kořist. Pro oba predátory uvažujte funkční odpověď typu I s jim vlastní konstantní vyhledávací účinností (a) a jim vlastní účinností přeměny živin (f).
- 2/** Výsledný model simulujte po dobu 100 dnů. Počáteční početnost prvoka L je 20 jedinců.
- 3/** S využitím Jacobiho matice stability zjistěte, je-li systém globálně stabilní.

▲ Řešení

1/ Podle popisu parametrů sestavte systém rovnic modelu Lotky a Volterry:

$$\frac{dH}{dt} =$$

$$\frac{dS}{dt} =$$

$$\frac{dL}{dt} =$$

2/ Vytvořte funkci s názvem `igp` obsahující systém diferenciálních rovnic:

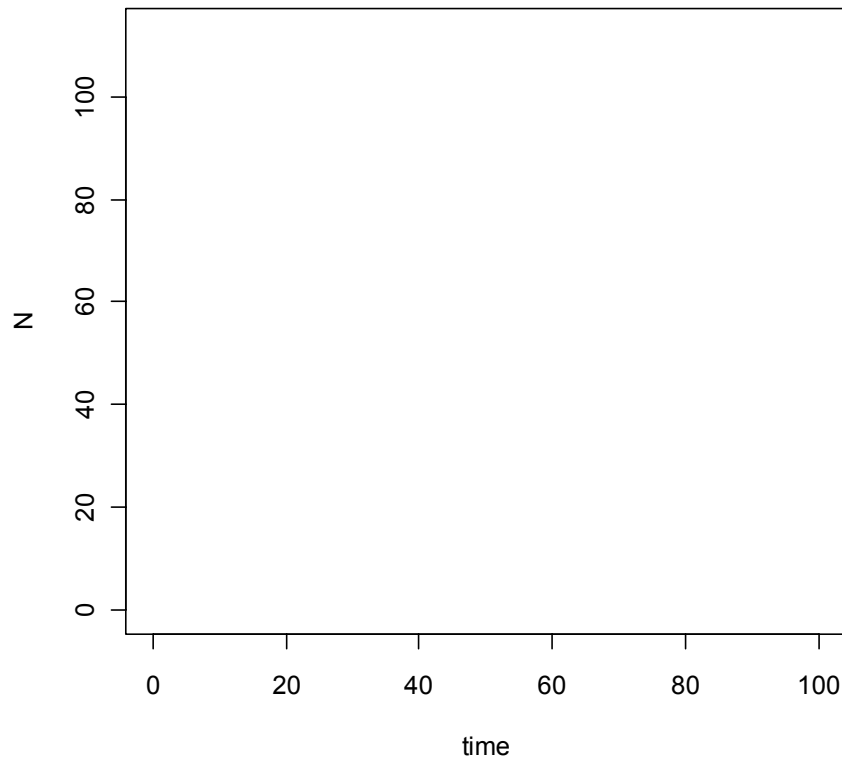
```
> igp<-function(t,y,param){
+ with(as.list(c(y,param)),{
+ dH.dt<-rh*H*(1-H/Kh)-alh*H*L-as*H*S
+ dS.dt<-fs*as*H*S-als*S*L-ds*S
+ dL.dt<-flh*alh*H*L+fls*als*S*L-dl*L
+ return(list(c(dH.dt,dS.dt,dL.dt)))}}
```

Definujte hodnoty parametrů (`parametry`), počáteční početnosti (`initial`) a čas (`time`). Proveďte simulaci příkazem `ode` z balíčku `deSolve` po dobu 100 dnů a vložte ji do objektu `sim`. Graf dynamik sestrojte příkazem `matplot`.

```
> parametry <- c(rh=1,Kh=200,alh=0.02,as=0.04,als=0.01,fs=0.06,
+ flh=0.03,fls=0.06,ds=0.1,dl=0.1)
> initial <- c(H=80,M=40,V=20)
> time <- seq(0,100,0.1)
> library(deSolve)
> sim <- ode(y=initial,times=time,func=igp,parms=parametry)
> matplot(time,sim[,-1],type="l",ylab="N",col=1:3)
> legend("topright",c("prey","small","large"),lty=1:3,col=1:3)
```

Překreslete graf do sešitu.

Simulace početnosti prvků



Konečné početnosti populací všech členů zjistíte pomocí příkazu **tail**.

```
> tail(sim)
```

Konečné početnosti jsou: $L = \dots\dots\dots$ $S = \dots\dots\dots$ $H = \dots\dots\dots$

Popište, jak simulace dopadla.

.....

2/ Jacobiho matici stability pro dynamický systém několika populací sestrojíme z parciálních derivací diferenciálních rovnic, které popisují změnu v jednotlivých populacích. V našem případě to budou tyto parciální derivace:

$$\mathbf{J} = \begin{pmatrix} \frac{\partial dH/dt}{\partial H} & \frac{\partial dH/dt}{\partial M} & \frac{\partial dH/dt}{\partial V} \\ \frac{\partial dM/dt}{\partial H} & \frac{\partial dM/dt}{\partial M} & \frac{\partial dM/dt}{\partial V} \\ \frac{\partial dV/dt}{\partial H} & \frac{\partial dV/dt}{\partial M} & \frac{\partial dV/dt}{\partial V} \end{pmatrix}$$

Pomocí příkazu **D** získáte parciální derivace pro všechny prvky matice **J**:

```
> D(expression(rh*H*(1-H/Kh)-avh*H*V-am*H*M), "H")
```

```

> D(expression(rh*H*(1-H/Kh)-avh*H*V-am*H*M), "M")
> D(expression(rh*H*(1-H/Kh)-avh*H*V-am*H*M), "V")
> D(expression(fm*am*H*M-avm*M*V-dm*M), "H")
> D(expression(fm*am*H*M-avm*M*V-dm*M), "M")
> D(expression(fm*am*H*M-avm*M*V-dm*M), "V")
> D(expression(fvh*avh*H*V+fvm*avm*M*V-dv*V), "H")
> D(expression(fvh*avh*H*V+fvm*avm*M*V-dv*V), "M")
> D(expression(fvh*avh*H*V+fvm*avm*M*V-dv*V), "V")

```

Sestavte matici **J** z tvarů parciálních derivací:

$$\mathbf{J} = \begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix}$$

Dosadíme-li do výsledných výrazů parciálních derivací hodnoty parametrů a početnosti, které považujeme za stabilní, můžeme pomocí vlastních čísel výsledné číselné matice zjistit, je-li tento stav globálně stabilní. Taková matice má všechny reálné části vlastních čísel záporné. Kladné vlastní číslo indikuje systém labilní.

Výpočet proveďte takto: definujte stabilní početnosti (z konce simulace), zpřístupněte jednotlivé parametry pomocí příkazu **attach** a spočítejte hodnoty parciálních derivací pomocí příkazu **deriv**. Ten umožňuje zadat více parciálních derivací najednou (písmena v uvozovkách spojená do vektoru). Výsledek proto uložíme do zvláštní proměnné a vyvoláme jen číselný výsledek pomocí příkazu **eval**. Zajímá nás poslední řádek výpisu.

```

> H=42; M=19; V=1
> attach(as.list(parameters))
> dH <- deriv(~rh*H*(1-H/Kh)-avh*H*V-am*H*M,c("H","M","V"))
> eval(dH)
> dM <- deriv(~fm*am*H*M-avm*M*V-dm*M,c("H","M","V"))
> eval(dM)
> dV <- deriv(~fvh*avh*H*V+fvm*avm*M*V-dv*V,c("H","M","V"))
> eval(dV)

```

Zapište zjištěné hodnoty do matice **JAK**:

$$\mathbf{JAK} = \begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix}$$

Ze získaných hodnot sestavíme matici **JAK** a spočítáme její vlastní čísla (příkaz **eigen**).

```

> JAK <- matrix(nrow=3,ncol=3, c(-1.8,-3.2,-1.6,0.096,-0.108,-0.4,0.012,
+ 0.012,-0.028),byrow=T)
> JAK
> eigen(JAK)

```

Zapište hodnoty vlastních čísel matice **JAK**:

Je systém stabilní, nebo labilní?

.....

▲ Popis

Ve skladu potravin se rozmnožili zavíječi. Jejich hustota je 50 jedinců na 10 kg mouky. Na jejich kontrolu se jako bioagens používají parazitoidi. Firmy nabízejí tři druhy parazitoidů (označme je A, B, C), které se liší v hodnotách dvou parametrů. Vyberte nejvhodnější druh parazitoida.

▲ Data

Pro populaci zavíječe platí tyto hodnoty parametrů: konečná míra růstu populace $\lambda = 2.3$ a nosná kapacita prostředí $K = 600$. Pro tři druhy parazitoidů je známa vyhledávací účinnost (a) a průměrný počet samic vylíhnutých z jednoho hostitele (c):

	druh A	druh B	druh C
a	0.003	0.1	0.008
c	1	2	1/3

▲ Úkoly

- 1/ S použitím Nicholson-Baileyho modelu vyberte nejvhodnější druh parazitoida pro kontrolu zavíječe. Na počátku bude vypuštěn jeden parazitoid na 10 kg mouky. Simulaci proveďte pro 20 generací.

▲ Řešení

- 1/ Diferenční model Nicholsona a Baileyho pro diskrétně se množícího parazitoida a hostitele s náhodným vyhledáváním a s populačním růstem hostitele závislým na hustotě má následující strukturu:

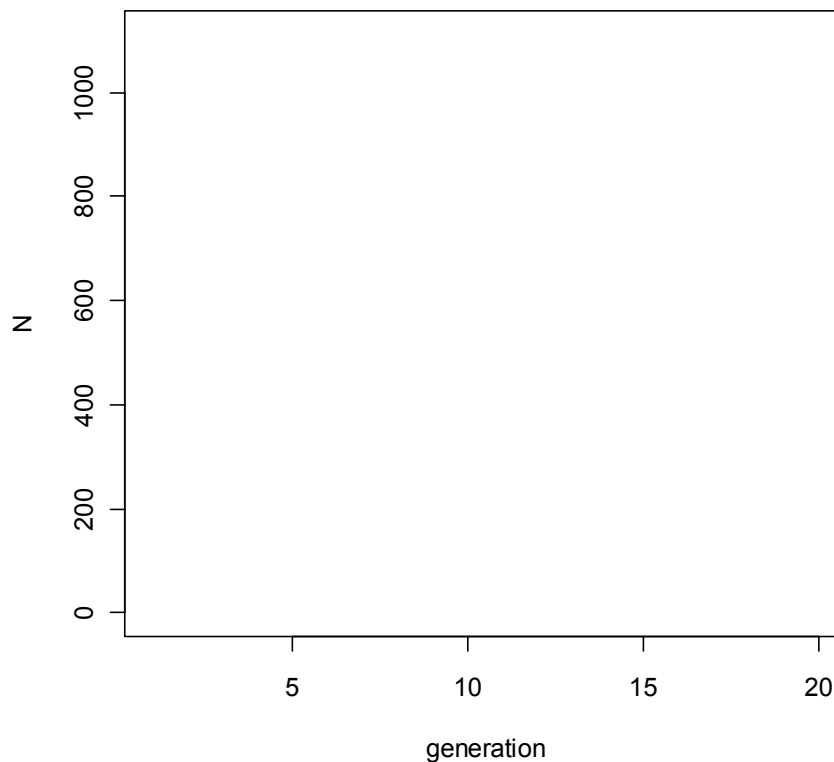
$$H_{t+1} = \lambda H_t e^{\left(1 - \frac{H_t}{K}\right) - aP_t} \quad P_{t+1} = cH_t(1 - e^{-aP_t})$$

Pro první druh parazitoida (A) je $a = 0.003$ a $c = 1$. Specifikujte délku simulací (time). Sestavte matici HP, do které vložíte výsledky simulací. Na první pozici matice vložte počáteční početnosti. Definujte parametry. Simulaci proveďte příkazem **for**. Graf dynamiky sestrojte příkazem **matplot**.

```
> time <- 20
> HP <- data.frame(H=numeric(time),P=numeric(time))
> HP[1,] <- c(50,1)
> L=2.3; K=600; a=0.003; c=1
> for (t in 1:(time-1)){
+   HP[t+1,1] <- L*HP[t,1]*exp((K-HP[t,1])/K-a*HP[t,2])
+   HP[t+1,2] <- c*HP[t,1]*(1-exp(-a*HP[t,2]))}
> matplot(HP,type="l", xlab="generation",ylab="N",lty=1:2,col=1)
> legend("topright",c("H","P"),lty=1:2)
```

Překreslete graf do sešitu.

Simulace dynamiky pro parazitoida A

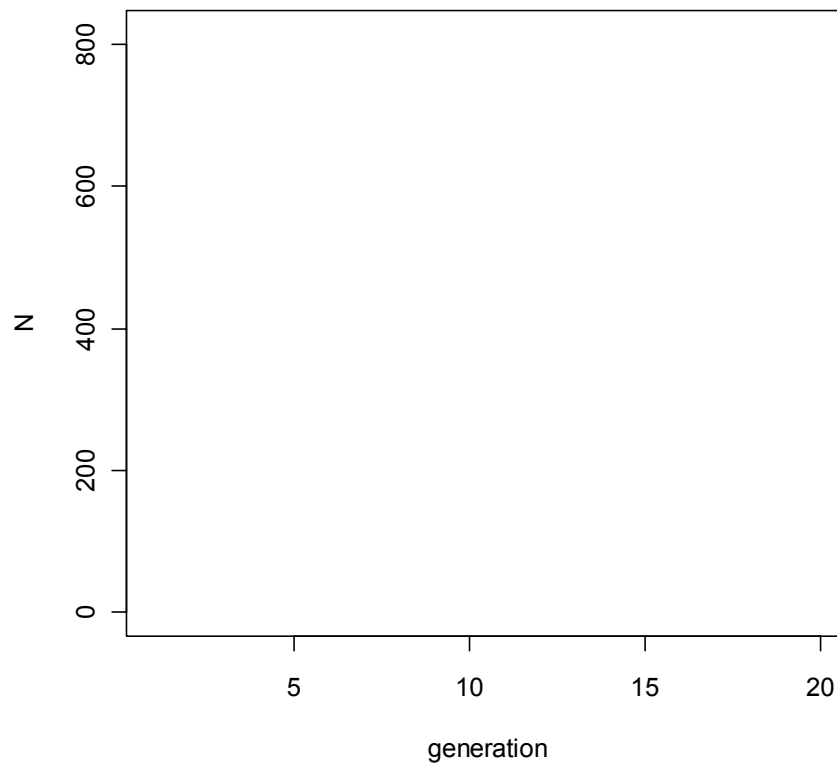


V simulaci pro parazitoida druhu B změňte hodnoty parametrů $a = 0.1$ a $c = 2$, jinak postupujte stejně jako u druhu A.

```
> L=2.3; K=600; a=0.1; c=2
> for (t in 1:(time-1)){
+   HP[t+1,1] <- L*HP[t,1]*exp((K-HP[t,1])/K-a*HP[t,2])
+   HP[t+1,2] <- c*HP[t,1]*(1-exp(-a*HP[t,2]))}
> matplot(HP,type="l",xlab="generation",ylab="N",lty=1:2,col=1)
> legend("topright",c("H","P"),lty=1:2)
```

Překreslete graf do sešitu.

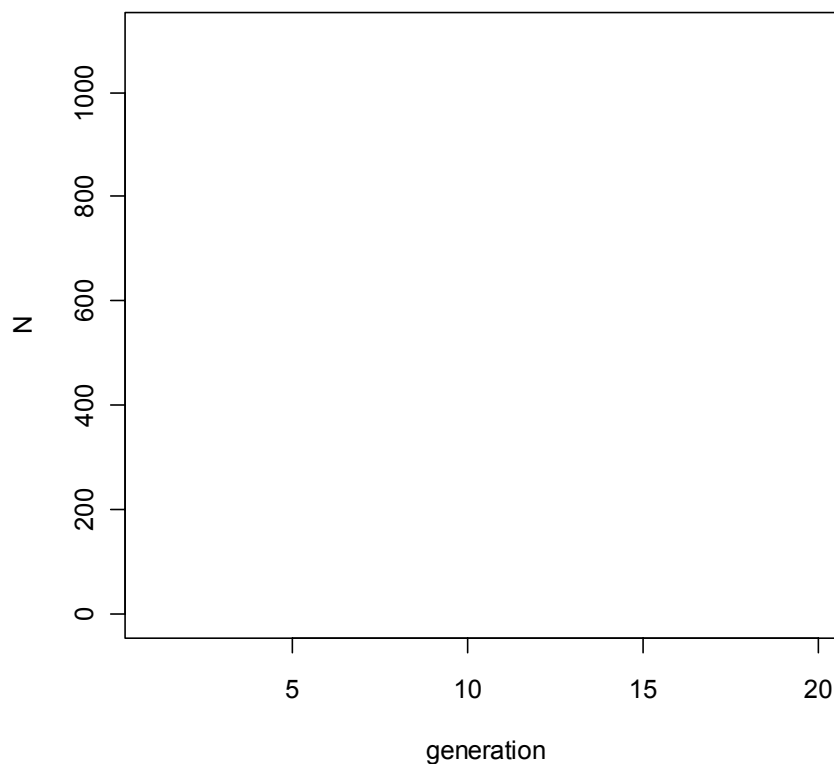
Simulace dynamiky pro parazitoida B



Podobně pro třetí druh parazitoida C změňte hodnoty parametrů: $a = 0.008$ a $c = 1/3$.

```
> L=2.3; K=600; a=0.008; c=1/3
> for (t in 1:(time-1)){
+   HP[t+1,1]<-L*HP[t,1]*exp((K-HP[t,1])/K-a*HP[t,2])
+   HP[t+1,2]<-c*HP[t,1]*(1-exp(-a*HP[t,2]))}
> matplot(HP,type="l",xlab="generation",ylab="N",lty=1:2,col=1)
> legend("topright",c("H","P"),lty=1:2)
```

Překreslete graf do sešitu.

Simulace dynamiky pro parazitoida C

Který druh parazitoida byste vybrali? Zdůvodněte proč.

.....

.....

▲ Poznámka

Tento typ modelu lze upravit pro nesespecializované parazitoidy např. nahrazením funkční odpovědi typu II typem III, který popisuje přepínání mezi typy kořisti (Hassell & Comins 1978).

26

Model patogen–hostitel

▲ Popis

Ve dvou městech se objevila vzteklna. Přinesla ji liška, která nakazila v každém městě jednoho volně pobíhajícího psa. V prvním městě je většina psů očkovaná, a tudíž ke vzteklině imunní. Ve druhém městě je očkovaných pouze pár psů. Zajímá nás osud šíření vztekliny v obou městech.

▲ Data

V obou městech je stejný počet psů, 20 jedinců/km². Jejich přirozená mortalita (m) v průběhu studie je zanedbatelná stejně jako jejich natalita (n), řekněme $m = n = 0$. V prvním městě je 75 % psů očkovaných, ve druhém pouze 5 %. Pes se nakazí od druhého psa v průměru jednou za 10 dnů (T). Inkubační doba nemoci (U) je 10 dnů. Délka nemoci (D) je 5 dnů. Nemoc je smrtelná.

▲ Úkoly

- 1/ Sestavte diferenciální systém rovnic pro vztah patogen–hostitel s inkubační dobou a odhadněte parametry z uvedených charakteristik.
- 2/ Nasimulujte dynamiku systému pro každé město zvlášť po dobu 60 dnů. Zjistěte, jestli dojde v obou městech k epidemii, tj. bude-li nakaženo více než 50 % psů.
- 3/ Zjistěte, jak ovlivní šíření nemoci přísná izolace psů.

▲ Řešení

- 1/ Model SEIR zahrnuje náchylné (S), nakažené jedince (E), tj. kteří ještě nejsou infekční, infekční (I) a rezistentní (R) jedince. Systém diferenciálních rovnic modelu bude následující:

$$\begin{aligned}\frac{dS}{dt} &= n(S + E + I + R) - \beta IS - mS \\ \frac{dE}{dt} &= \beta IS - \varphi E - mE \\ \frac{dI}{dt} &= \varphi E - \delta I - mI \\ \frac{dR}{dt} &= -mR\end{aligned}$$

kde β je parametr nakažlivosti, φ je parametr inkubace a δ je parametr úmrtnosti. Hodnoty těchto parametrů odhadněte podle těchto vzorců:

$$\begin{aligned}\beta &= \frac{1}{T}, & \text{kde } T \text{ je doba, za kterou se nakazí další pes;} \\ \varphi &= \frac{1}{U}, & \text{kde } U \text{ je inkubační doba;} \\ \delta &= \frac{1}{D}, & \text{kde } D \text{ je doba trvání nemoci.}\end{aligned}$$

Natalita a mortalita jsou zanedbatelné: $n = m = 0$.

2/ Vytvořte funkci s názvem `seir` podle systému diferenciálních rovnic výše:

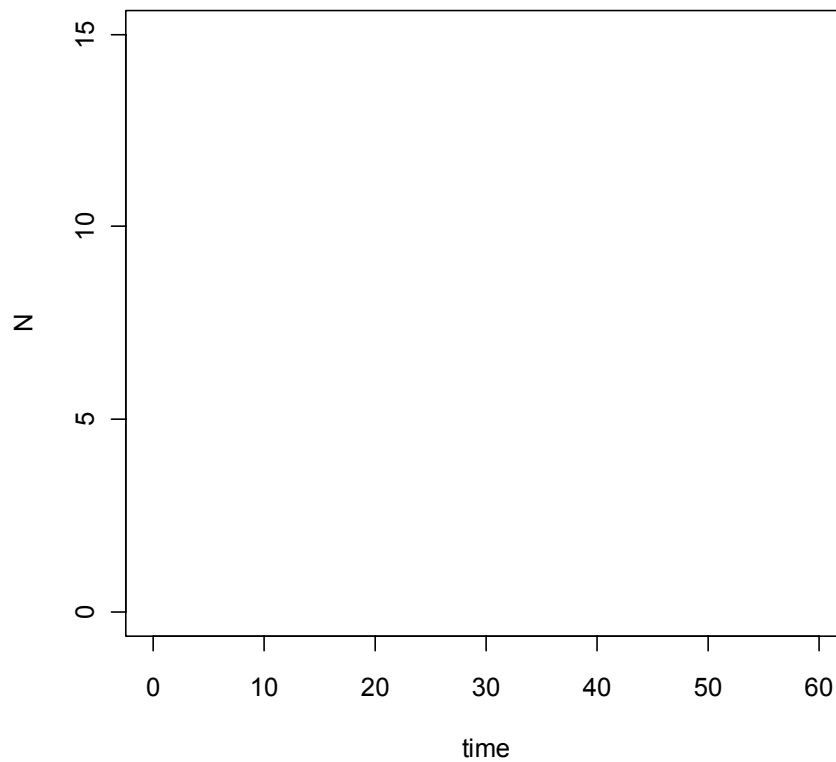
```
> seir <- function(t,y,param){
+ with(as.list(c(y,param)),{
+ ds.dt <- n*(S+E+I+R)-b*I*S-m*S
+ de.dt <- b*I*S-f*E-m*E
+ di.dt <- f*E-d*I-m*I
+ dr.dt <- -m*R
+ return(list(c(ds.dt,de.dt,di.dt,dr.dt)))})}
```

Definujte hodnoty parametrů (`parametry`), počáteční početnosti (`city75`) a času (`time`) pro první město se 75 % očkovaných psů. Proveďte simulaci příkazem `ode` z balíčku `deSolve`. Dynamiku vykreslete příkazem `matplot`.

```
> parametry <- c(n=0,m=0,b=1/10,f=1/10,d=1/5)
> time <- seq(0,60,0.1)
> city75 <- c(S=4,E=1,I=0,R=15)
> library(deSolve)
> sim1 <- ode(y=city75,times=time,func=seir,parms=parametry)
> matplot(time,sim1[,-1],type="l",ylab="N",lty=1,col=1:4)
> legend("right",c("S","E","I","R"),lty=1,col=1:4)
```

Překreslete graf do sešitu.

Simulace vztekliny v proočkovaném městě

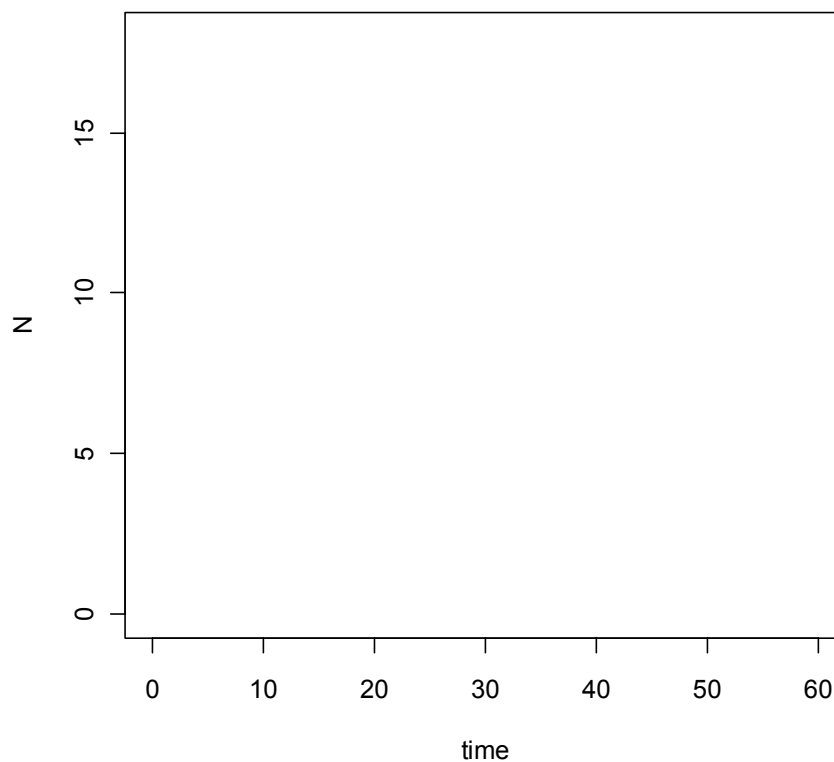


Nyní nasimulujte dynamiku ve druhém městě, kde je pouze 5 % očkovaných psů, s použitím stejného postupu jako v předchozím městě, ale za jiných počátečních početností pro S a R (`city05`).

```
> city05 <- c(S=18,E=1,I=0,R=1)
> sim2 <- ode(y=city05,times=time,func=seir,parms=parametry)
> matplot(time,sim2[,-1],type="l",ylab="N",lty=1,col=1:4)
> legend("right",c("S","E","I","R"),lty=1,col=1:4)
```

Překreslete graf do sešitu.

Simulace vztekliny v neočkovaném městě



Došlo v některém městě k epidemii? Pokud ano, ve kterém?

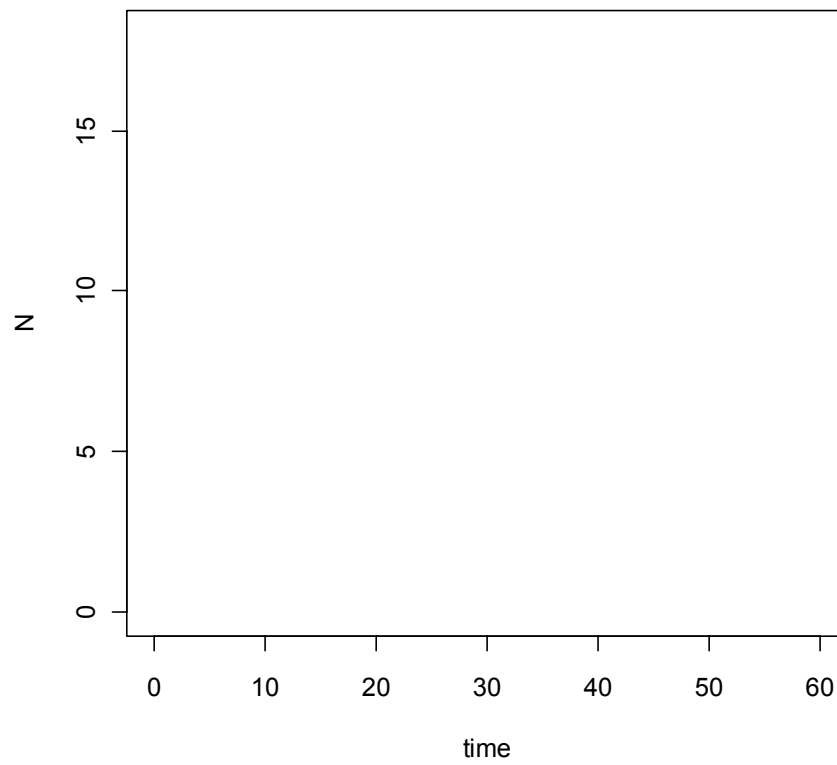
.....

3/ Zjistěte simulací, co se stane, když budou psi ve druhém městě izolováni do takové míry, že se infekční rychlost zmenší na 1 přenos za rok, tj.

$$\beta = \frac{1}{365}$$

```
> parametry <- c(n=0,m=0,b=1/365,f=1/10,d=1/5)
> sim3 <- ode(y=city05,times=time,func=seir,parms=parametry)
> matplot(time,sim3[,-1],type="l",ylab="N",lty=1,col=1:4)
> legend("right",c("S","E","I","R"),lty=1,col=1:4)
```

Překreslete graf do sešitu.

Simulace vztekliny v izolovaném městě

Zabránila izolace šíření?

.....

.....