1 Úvod

Tato učebnice je cvičební skriptum k předmětu Cvičení z populační ekologie živočichů přednášené na Přírodovědecké fakultě Masarykovy university. Populační ekologie je i praktická disciplína, které se využívá v různých odvětvích, jako je ochrana druhů, biologický boj proti škůdcům, epidemiologie, nebo lov hospodářsky významných druhů. Je založena na aplikaci matematických postupů v praktických problémech. Proto je nezbytné si tyto postupy procvičit na konkrétních datech. A to je právě cílem této učebnice. Konkrétně učebnice obsahuje 24 příkladů, které jsou podrobně probírány v průběhu cvičení.

Řešení všech příkladů je založené na použití softwaru. Nalezení řešení jednodušších příkladů by šlo i pouze s pomocí tužky a papíru. Ve složitějších příkladech, a těch je tady docela dost, je však nutné použít software. My jsme zvolili program s názvem R (R Development Core Team 2011), protože v něm lze vyřešit všechny příklady z této učebnice (většina by jich šla vyřešit také v MS Excelu, ale ne všechny). **R** je volně dostupné prostředí pro matematické a statistické modelování. Obsahuje obrovské množství metod a funkcí, a proto je to nejkomplexnější volně dostupný software. Pro seznámení se s prostředím **R**, tj. instalace a jeho základní ovládání, doporučujeme nastudovat si kapitolu 2 z knihy Pekár & Brabec (2009). K osvěžení práce s maticemi doporučujeme nastudovat kapitolu 4 z knihy Pekár & Brabec (2012). K řešení budete potřebovat tři extra balíčky příkazů, které je potřeba doinstalovat. Jmenovitě deSolve, rootSolve a Rramas. Dále doporučujeme zopakovat si základy integrálního a diferenciálního počtu běžně probírány na hodinách matematiky na střední škole (např. Hrubý & Kubát 2010).

V několika kapitolách je použit regresní (lineární nebo nelineární) model. Chybí však diagnostika výsledného odhadu. To proto, že cílem úloh není nalezení nejlepšího modelu/odhadu, ale pouze ukázat jak na dané data aplikovat vybraný model. Při skutečné analýze, je diagnostika nedílnou součástí nalezení správného výsledku. Pro tipy jak se diagnostika dělá viz. Pekár & Brabec (2009, 2012).

První kapitola, která následuje po úvodu je věnována přehledu základních matematických funkcí a jejich vlastností. Některé z těchto funkcí jsou pak použity v příkladech. Důležité je pochopit chování funkcí a význam parametrů, abyste je dokázali používat správně. Pak následují příklady. Vybrali jsme takové s jakými se můžete potkat ve svém výzkumu nebo později v praxi. U každého příkladu je nejprve stručně popsán problém, následují data a dále je seznam otázek. V části Řešení je popsán postup a poskytnut prostor pro samostatné nalezení odpovědi na položené otázky. Tato část obsahuje tučná šedá okna pro výpis příkazů v syntaxi prostředí **R**, prázdné tabulky pro vepsání odhadnutých koeficientů a také prázdné grafy pro překreslení výsledků do učebnice. Ke konci většiny příkladů je Poznámka, která obsahuje doplňkové informace k dané problematice. Na závěr každého příkladu je Tahák, který obsahuje syntaxi příkazů. Ty mohou být složitější a náročnější pro pochopení, ale zato efektivnější ve výpočtu. Syntaxe příkazů obsahuje zpravidla minimální množství argumentů, aby nebyly příliš dlouhé.

Na konci učebnice jsou uvedeny návrhy deseti krátkých experimentálních projektů, které jsou součástí cvičení. Projekty mají ukázat jak překonat praktické problémy se studiem populací.

Je v nich popsán doporučený postup, byť ten je nutné doladit dle okamžitých podmínek. Projekty jsou vypracovávány v skupinách, protože jsou docela časově náročné.

Úplně na konci je seznam literatury, který obsahuje jednak učebnice populační ekologie a dále a především doporučené publikace v nichž lze nalézt podrobnější popis té či oné metody.

V textu je použito několik typů fontů. Courier New pro příkazy v prostředí **R** a Times New Roman pro ostatní text. Courier New tučný označuje uživatelem zadávané příkazy a jejich argumenty, Courier New obyčejný pak názvy objektů. Jména packages (balíčků) jsou podtržena. Názvy objektů jsou záměrně v angličtině, abychom se vyhnuly diakritice. Jako oddělovač desetinných míst je použita tečka, nikoliv čárka. K výpočtům byla v tomto dílu použita verze **R** 2.13.0.

Na závěr bychom rádi poděkovali prof. RNDr. Vojtěchovi Jarošíkovi, Csc. za podnětné připomínky k textu.

Stano Pekár
Katka Kintrová

2 Přehled základních funkcí

V této kapitole si představíme několik základních matematických funkcí, jejich grafickou podobu a parametrický zápis. Hlavně se soustředíme na popis jejich chování vzhledem k hodnotám parametrů. Funkce mají v ekologii nezastupitelné místo jak uvidíte v řešených příkladech. Uvědomte si, že existuje nekonečně mnoho funkcí. Ale ne všechny jsou užitečné. Podle pravidla parsimonie jsou nejužitečnější takové, které mají málo parametrů. Navíc je důležité, aby parametry měli biologickou interpretaci.

Odhad hodnot parametrů pro daná data závisí od metody, kterou model fitujeme. Při použití modelu s lineární kombinací parametrů (tj. kde se parametry sčítají nebo odčítají) je situace snazší, protože jejich hodnoty příkaz lineární regrese (**`lm`**) automaticky dopočítá. Při použití metody nelineární regrese (**`nls`**) je nutné navrhnout přibližnou hodnotu každého parametru, aby měl příkaz z čeho startovat. Čím bližší startovací hodnotu parametru navrhneme, tím rychleji ji algoritmus odhadne. To také znamená, že pokud bude startovací hodnota příliš vzdálená, příkaz ohlásí chybu a žádné odhady nepořídí. Proto je nutné se naučit, co který parametr v dané funkci kontroluje. Při pořizování startovacích hodnot je vhodné zjistit, jak se funkce chová ve svých limitách, tedy jmenovitě když $x = 0$ a $x = \infty$. Startovací hodnoty pak vyčíst z grafu pomyslné křivky procházející naměřenými hodnotami. Hodnoty parametrů, které nelze z grafu vyčíst pak je potřeba odhadnout analyticky nebo iterativně.

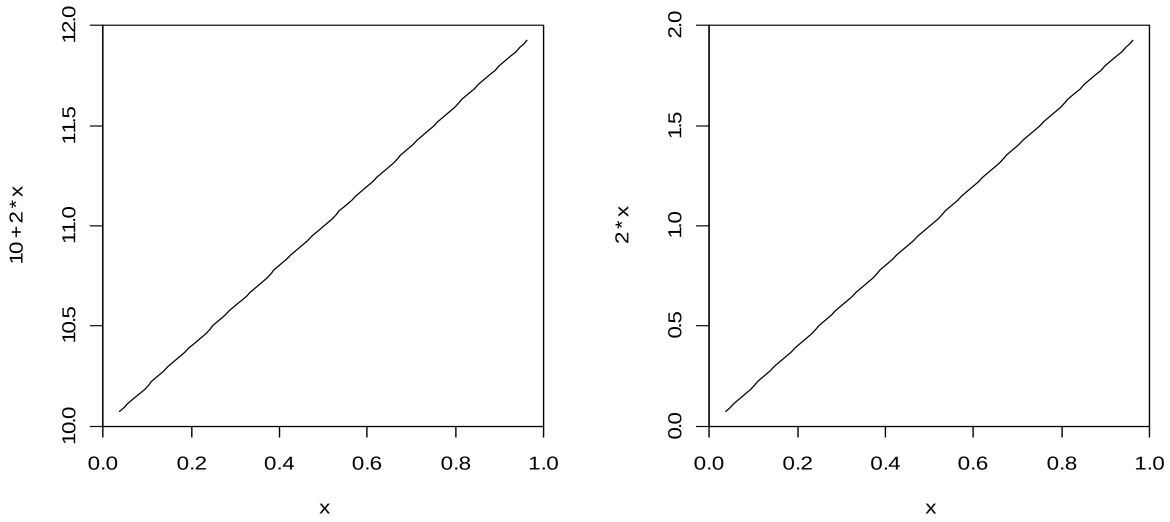
Pro vykreslení funkcí všelijakých tvarů zde použijeme univerzální příkaz **`curve`** s argumentem **`xlim`** (případně **`ylim`**) pro nastavení grafických mezí na osách *x* a *y*. Argumenty jsme však z příkazů vynechali. Funkce vykreslíme pouze pro několik málo hodnot parametrů. Čtenář si pak může vykreslit další pro jiné hodnoty parametrů. Na závěr kapitoly si ukážeme nástroje k vyšetřování průběhu funkcí. Při značení parametrů budeme stříškou značit odhady parametrů.

Byť kreslení funkcí v prostředí R je snadné a rychlé, pro zvídavé čtenáře doporučujeme internetovou stránku s interaktivními grafy, kde můžete parametry funkcí měnit průběžně. Naleznete ji na adrese http://www.karlin.mff.cuni.cz/katedry/kdm/diplomky/jaroslav_richter/ (je třeba mít nainstalován překladač jazyka JAVA).

Lineární funkce

Suverénně nejpoužívanějším modelem je lineární závislost daná vztahem $y = a + bx$. Zde parametr *a* udává průsečík (Intercept), tedy bod, ve kterém přímka protíná osu *y*, když $x = 0$. Parametr *b* určuje sklon přímky, přesněji tangentu úhlu, který svírá přímka s osou *x*. Je-li *b* kladné, přímka roste, je-li *b* záporné, přímka klesá. Speciálním případem lineárního modelu je přímka procházející nulou ($a = 0$): $y = bx$:

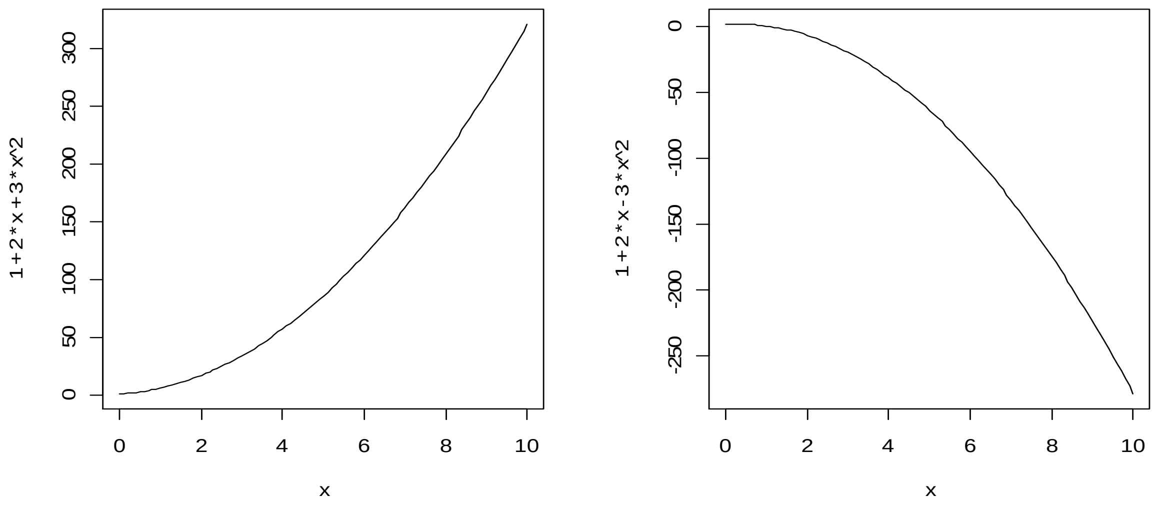
```
curve(10+2*x)
curve(2*x)
```



Polynomická funkce

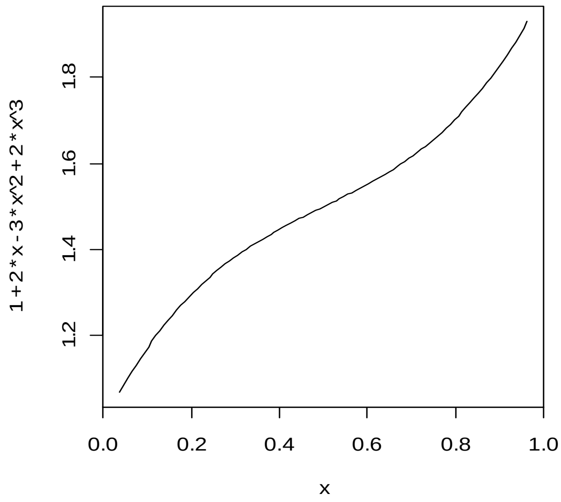
Tato funkce je rozšířením lineárního modelu o polynomické členy (další mocniny *x*). Jejím výsledkem je docela plastická křivka s mnoha přitažlivými vlastnostmi a proto je hojně užívána v mnoha situacích. Kvadratická funkce má tvar $y = a + bx + cx^2$, ve kterém parametry *a* a *b* mají stejné vlastnosti jako v lineární funkci. Parametr *c* udává sílu zakřivení a jeho znaménko směr zakřivení. Pokud je $c > 0$, bude se křivka zakřivovat nahoru, pokud je $c < 0$, bude se zakřivovat dolů:

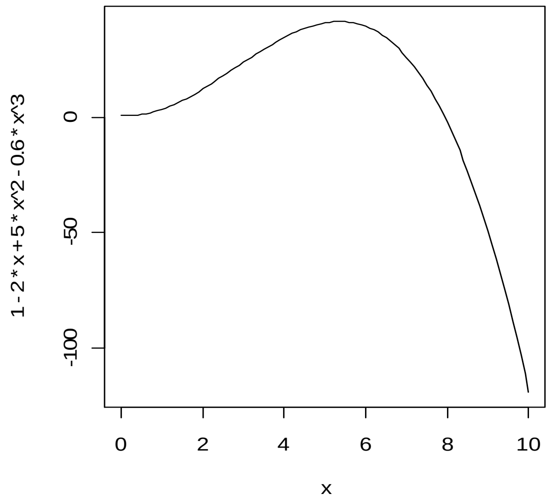
```
curve(1+2*x+3*x^2)
curve(1+2*x-3*x^2)
```



Kubický model zahrnuje i třetí mocninu: $y = a + bx + cx^2 + dx^3$. Parametr *d* udává sílu zakřivení a jeho znaménko směr zakřivení. Střídání znamének před parametry produkuje zvlněný tvar funkce:

```
curve(1+2*x-3*x^2+2*x^3)
curve(1-2*x+5*x^2-0.6*x^3)
```

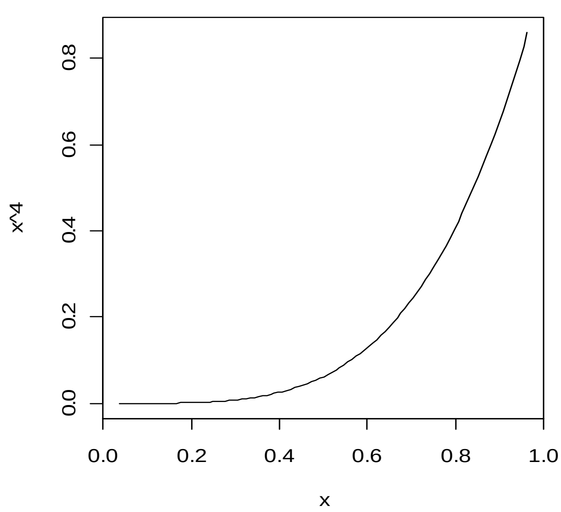


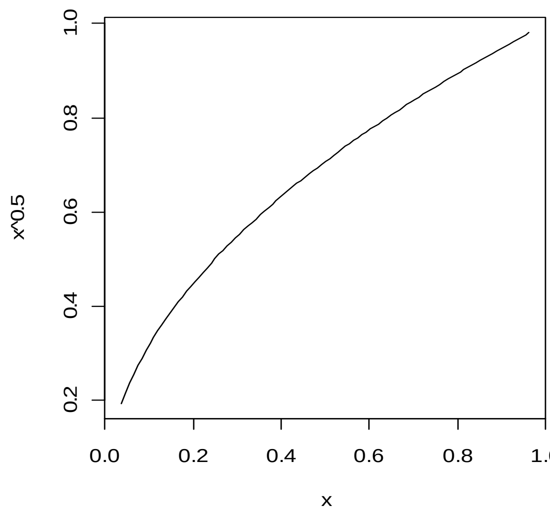


Mocninná funkce

Mocninná funkce s parametrem v exponentu $y = x^b$ se používá pro jednoduchou aproximaci geometrického růstu např. početnosti populace, kde *b* určuje rychlost růstu a je typicky > 1. Zápis funkce můžeme rozvinout o další parametry: $y = a(x-c)^b + d$. Parametr *c* posunuje křivkou podél osy *x*, parametr *a* násobí výslednou hodnotu (tím zrychluje nebo zpomaluje růst funkce) a parametr *d* posunuje celou křivkou podél osy *y*. Parametr *b* může nabývat také hodnot mezi 0 a 1 – pak mluvíme o odmocninné funkci, která má jiný tvar.

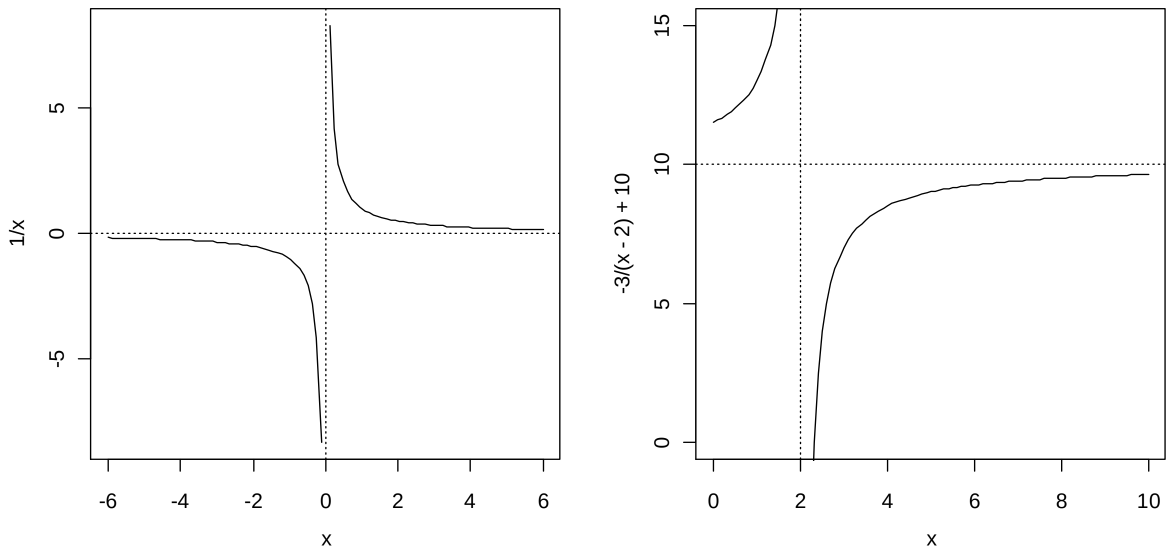
```
curve(x^4)
curve(x^0.5)
```





Je-li exponent záporný, jedná se o funkci lineární lomenou. Funkce $y = \frac{ax+b}{cx+d}$ popisuje asymptotický růst nebo pokles k nějaké hodnotě. Grafem této funkce je hyperbola se středem v bodě $\left[-\frac{d}{c}, \frac{a}{c}\right]$. Parametry $-{}^{d}\!/\!_{c}$ posunují křivkou podél osy x a definují tak svislou asymptotu, parametry ${}^{a}\!/\!_{c}$ posunují křivku podél osy y a definují tak vodorovnou asymptotu. Použijeme-li vyšší mocniny x, mluvíme o funkci lomené:

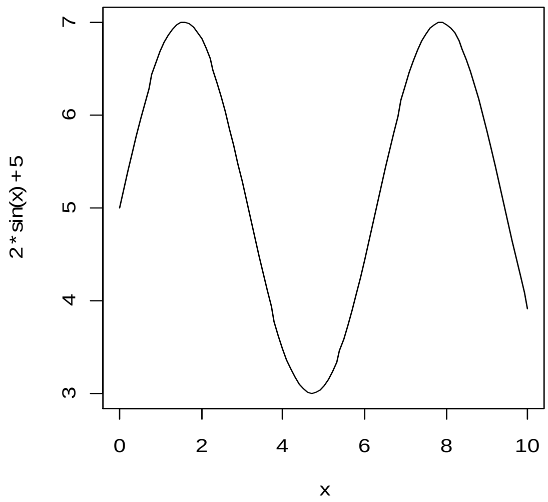
```
curve(1/x)
curve(-3/(x-2)+10)
```

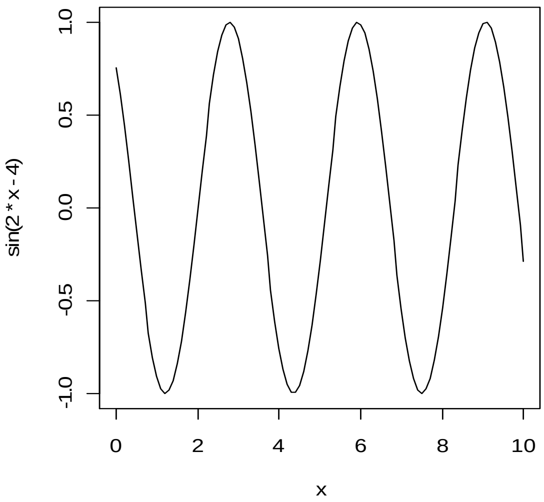


Goniometrické funkce

Pokud se měřené hodnoty opakují v pravidelných cyklech, jako např. teplota, lze je modelovat trigonometrickými funkcemi. Obecná sinová funkce $y = a\sin(b(x-c)) + d$ má čtyři parametry: a určuje výšku amplitudy, b definuje délku periody ($|b| > 1$ periodu zkracuje, $|b| < 1$ prodlužuje), c posouvá křivku podél osy x, zatímco d podél osy y.

```
curve(2*sin(x)+5)
curve(sin(2*x-4))
```

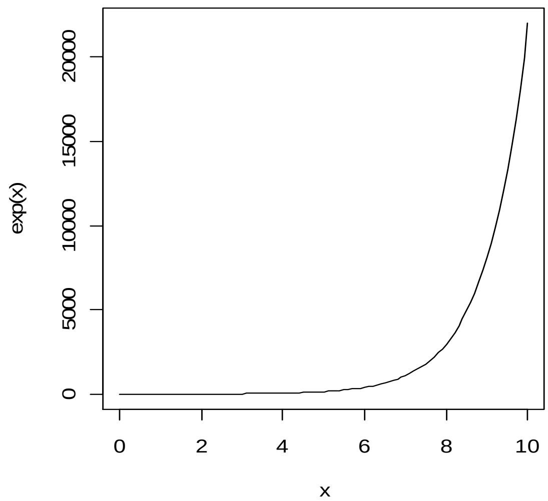


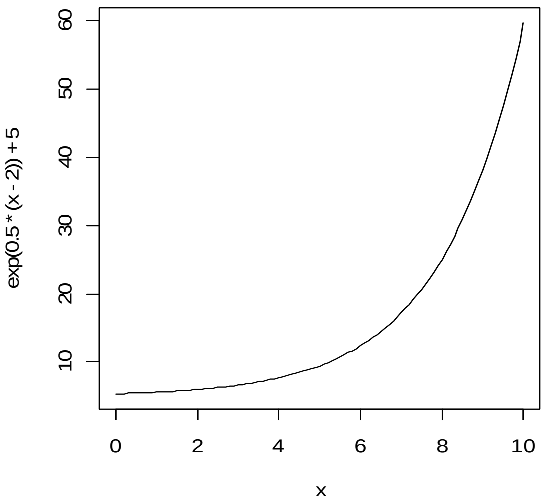


Exponenciální funkce

Na první pohled obdoba funkce mocninné, ovšem rychlost růstu či poklesu křivky je mnohem vyšší. Exponenciální funkce má *x* v exponentu: $y = a^x$, přičemž základ *a* je reálné kladné číslo různé od jedničky. Používá se pro popis jevů, které se s rostoucím *x* velmi rychle zvětšují nebo zmenšují. Přirozená exponenciální funkce v obecném tvaru $y = ae^{b(x-c)} + d$ má 4 parametry ovlivňující průběh křivky: parametr *c* posunuje křivkou podél osy *x*, *b* definuje rychlost zakřivení funkce a jeho znaménko rostoucí nebo klesající tendenci, parametr *a* dále násobí výslednou hodnotu a parametr *d* posunuje celou křivku podél osy *y*.

```
curve(exp(x))
curve(exp(0.5*(x-2))+5)
```





Různé formy exponenciálních funkcí si představíme v několika příkladech, kde si podrobněji popíšeme jejich vlastnosti.

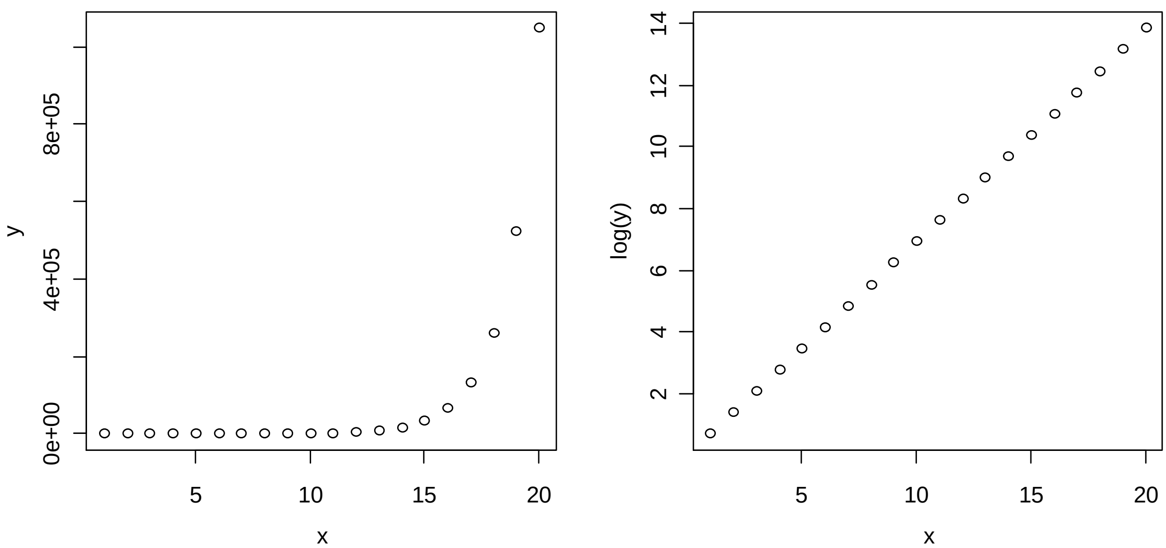
Logaritmická funkce

Tuto funkci používáme nejčastěji v základním tvaru $y = \log_a(x)$, kde $x > 0$, a označuje základ logaritmu. Je to reálné číslo větší než nula a různé od jedničky. Logaritmická funkce je inverzní k funkci exponenciální, proto ji často používáme v případech, kdy studované hodnoty narůstají exponenciálně a my je chceme vyrovnat, tedy zlinearizovat. Běžné značení $\ln(x)$ pro přirozený logaritmus a $\log(x)$ pro desítkový logaritmus není v prostředí **R** zavedené. Tam se přirozený logaritmus zapíše jako **`log(x)`** a desítkový logaritmus jako **`log10(x)`**. Obecný logaritmus o základu a se zapíše jako **`log(x,a)`**.

Předvedeme si linearizaci na jednoduchém příkladu:

$$y = 2^x$$
$$\ln(y) = \ln(2^x) = x\ln(2) = 0.69x$$

```
x<-1:20
y<-2^x
plot(y~x)
plot(log(y)~x)
```



Vyšetřování průběhu funkce

Přesný popis tvaru funkce je důležitý pro různé aplikace, jak později uvidíte. Zejména nás budou zajímat tyto dvě vlastnosti: průsečíky se souřadnými osami a lokální extrémy. Průsečík s osou y získáme dosazením hodnoty $x = 0$. Průsečík s osou x získáme řešením rovnice $f(x) = 0$, tedy pro jaké x je výsledek roven nule?

Lokální maximum nebo minimum funkce zjistíme tak, že spočítáme nulové body (kořeny) první derivace funkce, tedy $f'(x) = 0$. Nalezení první derivace funkce není pro nematematicky vzdělaného člověka snadné. Naštěstí to lze v programu **R** hravě zvládnout.

Nejprve definujeme tvar funkce pomocí příkazu **`expression`**, pak jej příkazem **`D`** zderivujeme podle *x*, což uvedeme do uvozovek, a příkazem **`uniroot.all`** z balíčku rootSolve nalezneme kořeny první derivace. Postup si ukážeme na Gaussově funkci (z kap. 16) s vrcholem $x = 5$.

```
y<-expression(2*exp(-(x-5)^2/3)+8)
D(y,"x")
-(2 * (exp(-(x - 5)^2/3) * (2 * (x - 5)/3)))
library(rootSolve)
max<-uniroot.all(function(x) -(2*(exp(-(x-5)^2/3)*(2*(x-5)/3))),
+ lower=0,upper=10);max
[1] 5
```

3 Odhad velikosti populace

Popis

Chceme zjistit početnost švábů v budově, abychom mohli naplánovat správný typ zásahu insekticidem. V několika místnostech jedné budovy byla sledována početnost švábů v průběhu 4 dnů pomocí metody opakovaných odchytů. Švábi byli odchyceni do pastí s návnadou. Každý den byli švábi v pastech sečteni a označeni barvou (každý den jinou). Všichni živí jedinci byli následně vypuštěni.

Data

Zjištěné hodnoty jsou uspořádány v následující tabulce:

Den odchytu i	**Počet odchycených** n_i	**Počet vypuštěných** a_i	**Počet znovu odchycených (r_{ij}) v den i a označených v den j**		
			r_{i1}	r_{i2}	r_{i3}
1	60	58			
2	125	123	15		
3	154	150	5	38	
4	189	187	9	20	45

Úkoly

1/ Odhadněte velikost populace a její směrodatnou odchylku pro každý den za předpokladu, že populace je uzavřená, tj. nemůže migrovat mezi budovami.

2/ Odhadněte velikost populace pro každý den za předpokladu, že populace je otevřená, tj. jedinci mohou migrovat mezi budovami.

3/ Pro oba předpoklady spočtěte průměrnou populační početnost švábů.

Řešení

1/ Pro odhad velikosti uzavřené populace použijte Petersen-Lincolnův odhad:

$$\hat{N}_i = \frac{a_{i-1} n_i}{r_{i(i-1)}} \text{ pro } i\text{-tý den.}$$

Můžete spočítat odhady pro všechny 4 dny? Zdůvodněte, proč nemůžete:

……………………………………………………………………………………………………

Spočtěte odhady:

Do tabulky zapište výsledky odhadů:

	$\hat{N}$	SD
1		
2		
3		
1-3		

Nyní spočtěte odhady směrodatných odchylek pro velikosti populace podle vzorce (Bailey 1952):

$$SD = \sqrt{\frac{a_{i-1}^2 n_i (n_i - r_{i(i-1)})}{r_{i(i-1)\,i(i-1)}^3}}$$

Výsledky zapište do tabulky výše.

2/ Pro odhad velikosti otevřené populace použijte Jolly-Seber model:

$$\hat{N}_i = \frac{M_i n_i}{r_{i\bullet}} \text{ pro } i\text{-tý den,}$$

$$\text{kde } r_{i\bullet} = \sum_{j=1}^{i-1} r_{ij} \text{ ,}$$

$$M_i = \frac{a_i Z_i}{R_i} + r_{i\bullet} ,$$

$$Z_i = \sum_{k=i+1}^{n} \sum_{j=1}^{i-1} r_{kj} ,$$

$$R_i = \sum_{k=i+1}^{n} r_{ki} ,$$

a n je počet všech odchytových dnů.

Spočtěte hodnoty $r_{i\bullet}$, R_i a Z_i pro každý den do následující tabulky:

$\boldsymbol{i}$	$\boldsymbol{n_i}$	$\boldsymbol{a_i}$	*1*	*2*	*3*	$r_{i\bullet}$
1	60	58				
2	125	123	15			
3	154	150	5	38		
4	189	187	9	20	45	
$\boldsymbol{R}_i$						
$\boldsymbol{Z}_i$						

Nyní spočtěte odhady pro dny 2 a 3 podle vzorce uvedeného výše:

Do tabulky zapište výsledky odhadů:

$\hat{N}_2$	
$\hat{N}_3$	
$\overline{\hat{N}}$	

3/ Průměrnou velikost populace pro oba odhady spočtete jako aritmetický průměr jednotlivých dnů příkazem `mean`:

Výsledky zapište do tabulek výše.

Poznámka

Další přesnější metody odhadu velikosti populace, jako např. Cormack-Jolly-Seber, lze nalézt např. v Amstrup et al. (2005). V prostředí R je dostupných hned několik balíčků (BTSPAS, capwire, CARE1, Distance, mra, mrds, PL.popN, Rcapture, secr, SPACECAP) s jejichž pomocí lze sofistikovaně odhadnout velikost populace pro různé populace a další jejich statistiky (demografické parametry, hustotu, a pod.). Příkazy pro výpočet odhadů v těchto balíčcích však vyžadují jedinečné (nezaměnitelné) označení jedinců.

Tahák

```
N2<-58*125/15; N2
N3<-123*154/38; N3
N4<-150*189/45; N4
sqrt(58^2*125*(125-15)/15^3)
sqrt(123^2*154*(154-38)/38^3)
sqrt(150^2*189*(189-45)/45^3)
M2<-123*14/58+15; M2
M3<-150*29/45+43; M3
No2<-M2*125/15; No2
No3<-M3*154/43; No3
mean(c(N2,N3,N4))
mean(c(No2,No3))
```

4 Diskrétní hustotně-nezávislý růst populace

Popis

Populace ploštic byla monitorována jednou ročně na jaře po dobu deseti let. Na vybrané lokalitě byla vymezena plocha 100 m^2, která byla prosmýkána. Počet všech ulovených ploštic, juvenilních i adultních, byl sčítán a pak byly ploštice navráceny zpátky. Stejný postup se opakoval každý rok.

Data

V průběhu 10 let byly zaznamenány tyto početnosti (*N*) ploštic:

Rok	***N***
1	160
2	172
3	188
4	154
5	176
6	185
7	168
8	194
9	170
10	169

Úkoly

1/ Vytvořte graf početnosti populace ploštic v čase. Spočtěte míru populačního růstu pro každý rok a průměrnou míru růstu.

2/ Zjistěte jak se bude měnit početnost populace ploštic v průběhu dalších 10 let, pokud bychom uvažovali o konstantní průměrné míře populačního růstu a počáteční početnosti 90 jedinců.

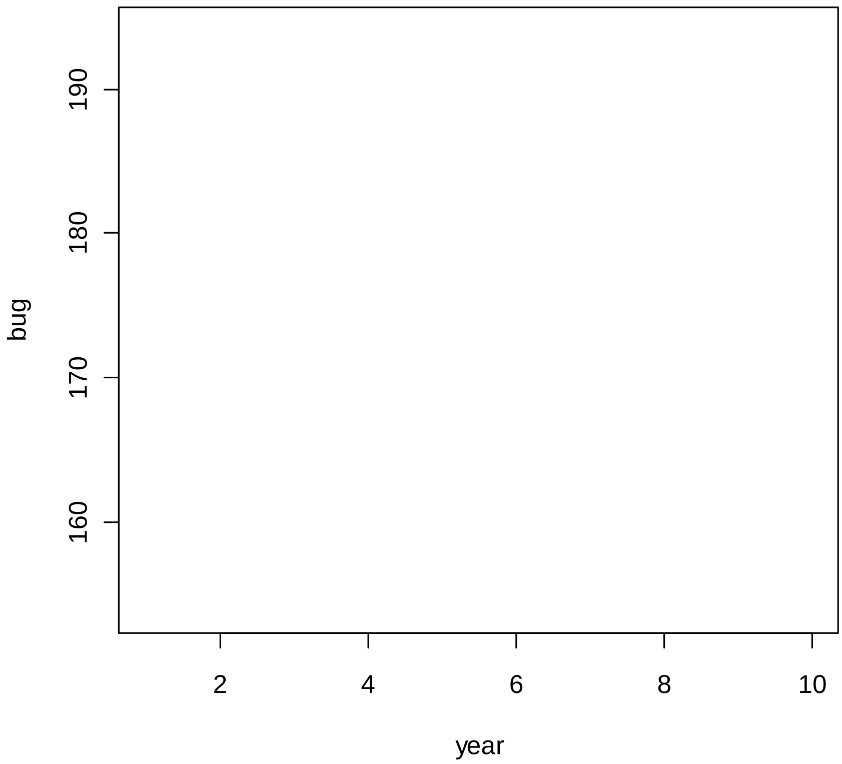
3/ Zjistěte jak by se měnila početnost populace v průběhu 20 let, kdyby míra populačního růstu byla proměnlivá a počáteční početnost populace by měla 100 jedinců.

Řešení

1/ Zaznamenané početnosti populace uložte do vektoru `bug`. Do spojnicového grafu vyneste závislost početnosti na čase pomocí příkazu **`plot`**:

Překreslete graf do sešitu.

Zaznamenané početnosti v průběhu 10 let



Mění se velikost populace v průběhu 10 let? Roste nebo klesá?

……………………………………………………………………………………………………

Ploštice se rozmnožují pouze jednou za sezónu, proto se pro popis jejich populačního růstu hodí diskrétní model:

$$N_t = N_{t-1}\lambda_t \,.$$

Odhad meziroční míry populačního růstu, λ_t, zjistíte úpravou tohoto vzorce:

$$\lambda_t = \frac{N_t}{N_{t-1}} \,.$$

Kolik z hodnot λ_t je větších a kolik menších než 1?

větších	menších

Průměrnou hodnotu míry populačního růstu, $\bar{\lambda}$, spočtete podle vzorce pro geometrický průměr:

$$\bar{\lambda} = \left(\prod_{t=1}^{T} \lambda_t \right)^{\frac{1}{T}} = (\lambda_1 \lambda_2 ... \lambda_T)^{\frac{1}{T}}$$

R

Průměrná hodnota $\bar{\lambda}$ je ……………

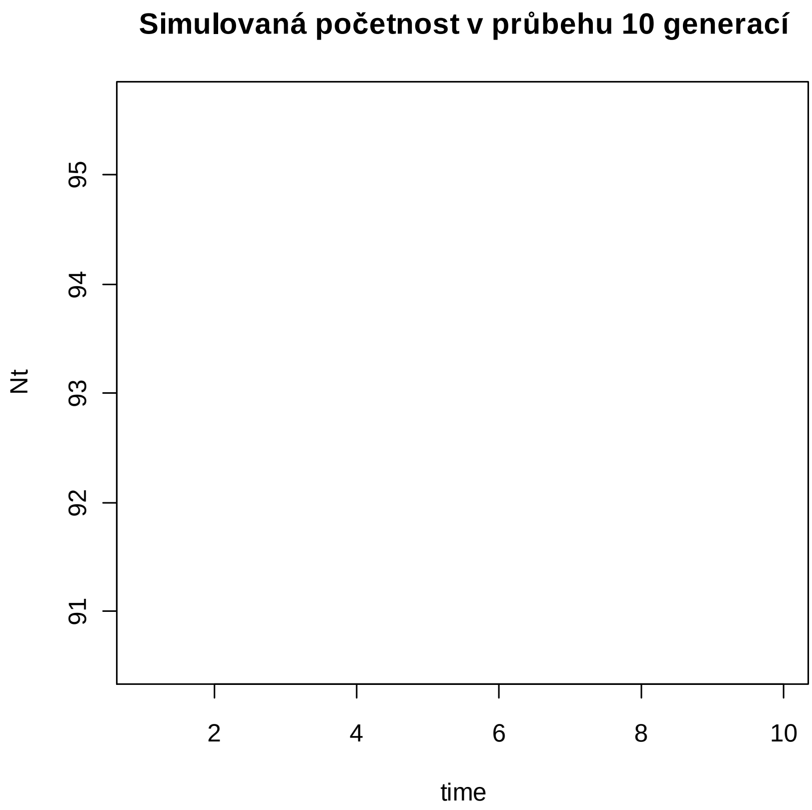
2/ K simulaci změny velikosti populace s konstantní mírou populačního růstu v průběhu dalších 10 let použijte model pro exponenciální růst s počátečním počtem jedinců 90:

$$N_t = N_0 \bar{\lambda}^t$$

Vytvořte vektor `time` s hodnotami 1 až 10.

R

Překreslete grafu do sešitu.



Čekali jste, že populace bude růst? Odůvodněte proč:

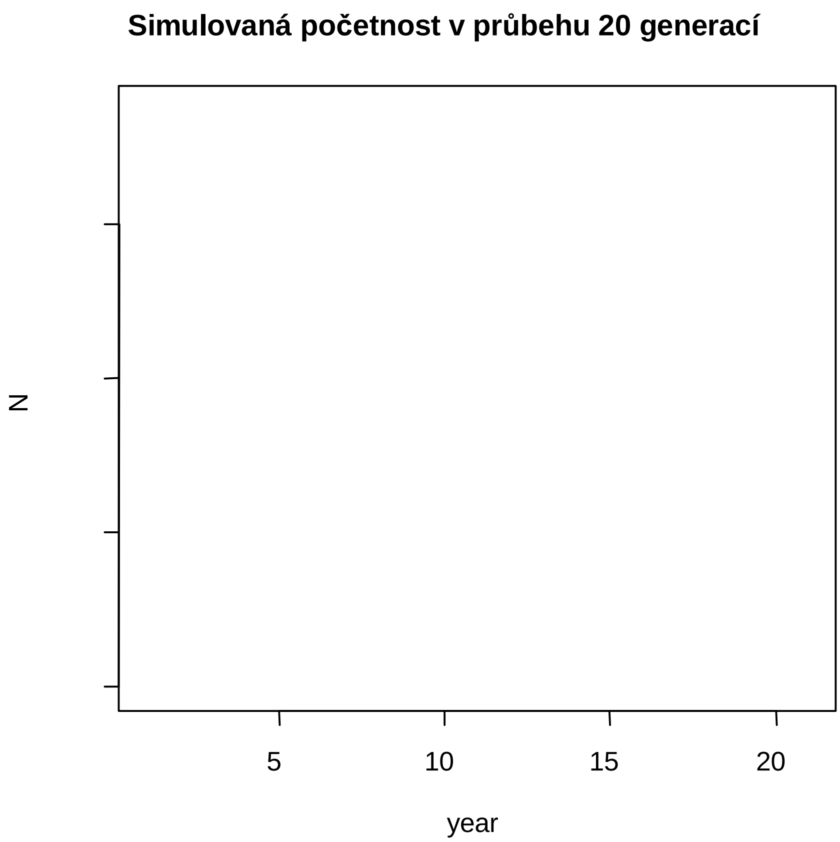
……………………………………………………………………………………………………

……………………………………………………………………………………………………

3/ Nyní budete simulovat populační dynamiku jinak. Míra populačního růstu λ se bude náhodně měnit. V prvním obrázku jsme viděli, že náhodná změna míry populačního růstu je přirozená. K simulaci použijeme předchozí model, ve kterém λ nebude konstantní, ale bude náhodně vybírána z λ_t, příkaz **`sample`**. Počáteční početnost vložte do vektoru `N` a simulaci proveďte příkazem **`for`**.

R

Překreslete grafu do sešitu.



Je výsledkem simulace monotónní růst nebo pokles velikosti populace? Odůvodněte, proč je nebo není?

…………………………………………………………………………………………………

…………………………………………………………………………………………………

Jaká je velikost nasimulované populace po 20 letech? …………………

Tahák

```
bug<-c(160,172,188,154,176,185,168,194,170,169)
plot(bug,type="b",xlab="year")
lambda<-bug[-1]/bug[-10];lambda
L<-prod(lambda)^(1/9);L
time<-1:10
Nt<-90*L^time
plot(time,Nt,type="b")
sim<-sample(lambda,20,replace=T)
N<-numeric(21)
N[1]<-100
for(t in 1:20) N[t+1]<-N[t]*sim[t]
plot(1:21,N,type="b",xlab="year")
```

5 Kontinuální hustotně-nezávislý růst populace

Popis

Velikost populace roztočů byla zaznamenávána v třídenním intervalu po dobu jednoho měsíce. Na počátku byla skupina roztočů vložena do nádoby s moukou a umístěna do klimaboxu se stálou teplotou a vlhkostí. Každý třetí den byla mouka proseta a byl stanoven počet všech jedinců bez ohledu na vývojové stádium. Poté byli všichni vráceni zpět do nádoby.

Data

V průběhu měsíce byly zaznamenány tyto početnosti (*N*):

Den	***N***
3	165
6	145
9	139
12	125
15	105
18	101
21	88
24	81
27	73
30	69

Úkoly

1/ Vytvořte graf závislosti početnosti roztočů na čase. Zjistěte hodnotu vnitřní míry růstu populace a počáteční početnost v čase 0. Zjistěte za jakou dobu (ve dnech) se populace zmenší na polovinu, za předpokladu odhadnuté míry růstu.

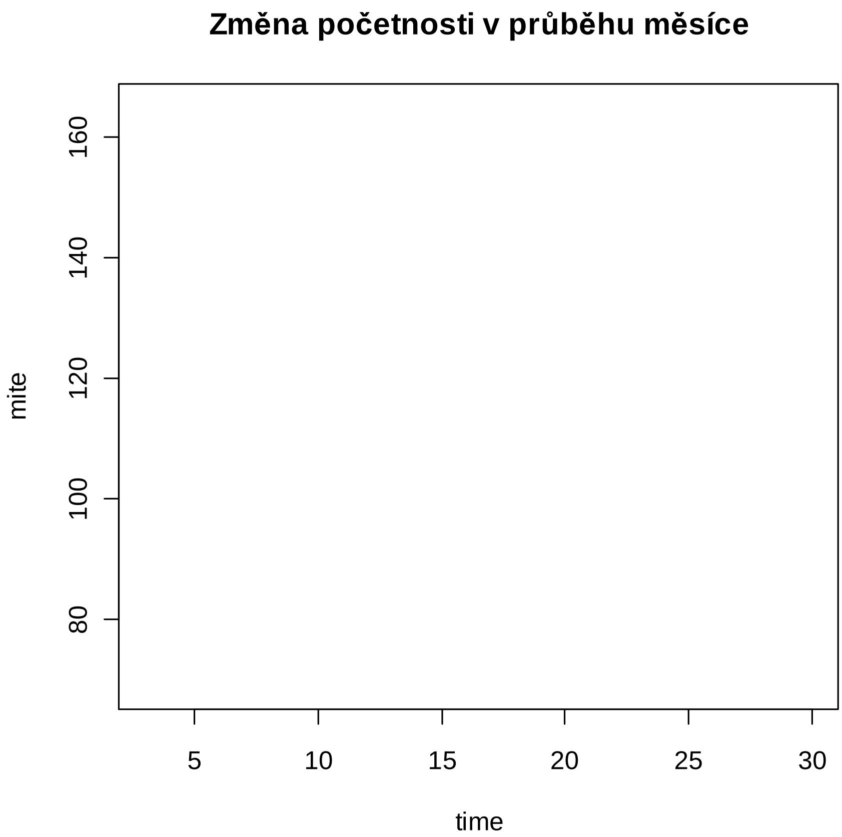
2/ Simulujte růst populace po dobu 5 týdnů pokud má populace na začátku 69 jedinců a s použitím odhadnuté hodnoty míry růstu.

3/ Zjistěte hodnotu *r*, pokud byste znali pouze počáteční (165) a konečnou velikost (69) populace.

Řešení

1/ Početnosti vložte do vektoru s názvem `mite` a čas do vektoru `time`. Zaznamenané početnosti populace vyneste do spojnicového grafu pomocí příkazu **`plot`**. Čas počítáme od třetího dne.

Překreslete graf do sešitu.



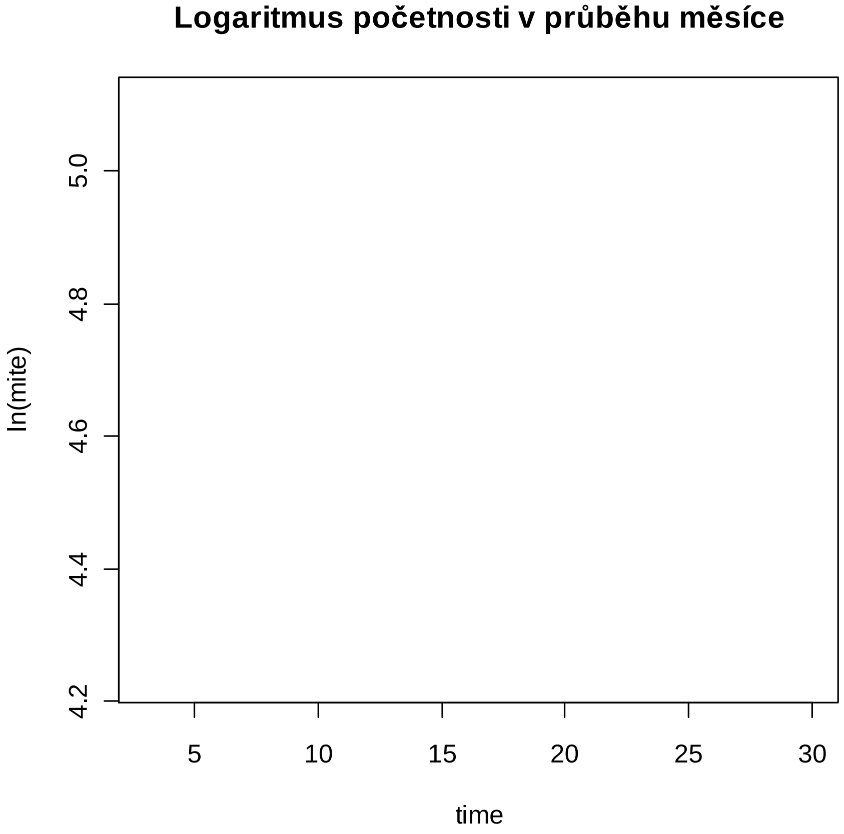
Kontinuálnímu růstu populace roztočů dobře odpovídá exponenciální model. Hodnotu vnitřní míry růstu izolované populace zjistíte linearizací vztahu pro exponenciální růst:

$$N_t = N_0 e^{rt} \quad \rightarrow \quad \ln(N_t) = \ln(N_0) + rt \text{, kde } N_0 = e^{\hat{\alpha}} \text{ a } r = \hat{\beta}.$$

Závislost zlogaritmovaných hodnot N_t na čase t vyneste do grafu a hodnoty parametrů lineárního vztahu odhadněte pomocí lineární regrese, příkaz **`lm`**. Odhady parametrů zjistíte příkazem **`coef`**.

Překreslete body do grafu. Proložte jimi regresní přímku a zapište její rovnici:

....



Jaká je hodnota r? $\hat{r} =$ ………

Jaká je hodnota N_0?

$\hat{N}_0 =$ ………

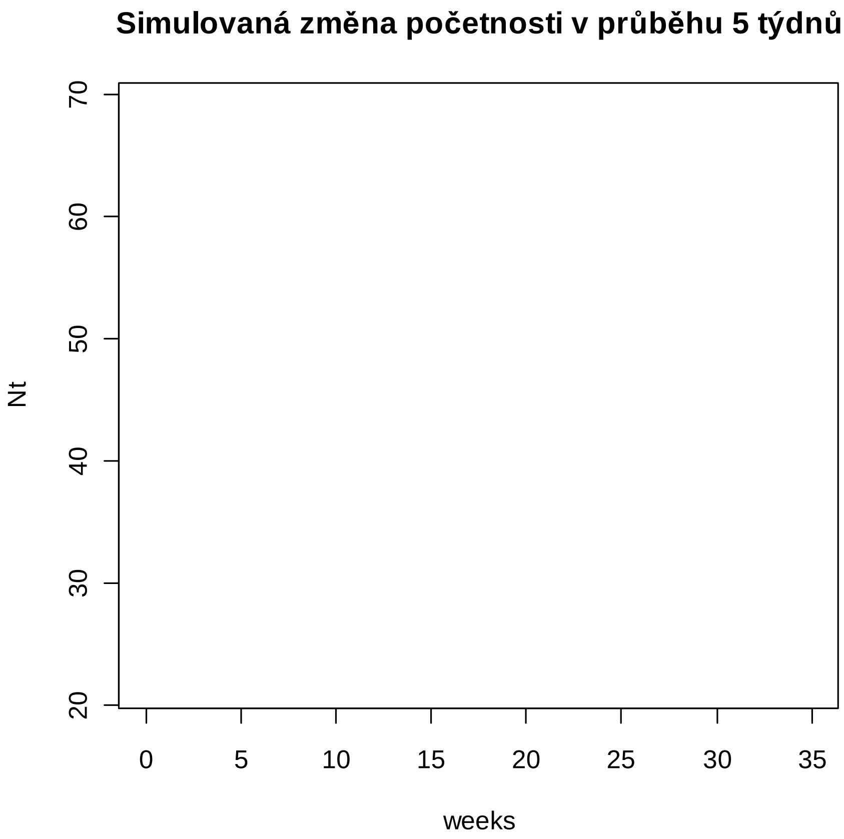
Pokud velikost populace klesla na polovinu, lze levou stranu rovnice exponenciálního růstu nahradit zlomkem: $N_t = \frac{N_0}{2}$. Úpravou rovnice pak vyjádříte čas t, pro který tento okamžik nastane:

$$\frac{N_0}{2} = N_0 e^{rt} \quad \rightarrow \quad \frac{1}{2} = e^{rt} \quad \rightarrow \quad t = \frac{\ln(0.5)}{r}$$

Populace se zmenší na polovinu za …………… dnů.

2/ K simulaci použijte model exponenciálního růstu s počáteční početností ží jedinců a mírou růstu odhadnutou výše. Do vektoru s názvem `weeks` vložte 35 dnů.:

Překreslete grafu do sešitu.



3/ Úpravou vztahu pro logaritmovaný exponenciální růst, tj. vyjmutím *r* na levou stranu dostanete:

$$r = \frac{\ln(N_t) - \ln(N_0)}{t}$$

Hodnota *r* by byla ……………

Poznámka

Odhad parametrů z časové řady je zpravidla zatížen autokorelaci, proto ke správnému odhadu parametrů by místo lineární regrese měl být použít model třídy GLS nebo GEE (viz Pekár & Brabec 2012).

Tahák

```
mite<-c(165, 145, 139, 125, 105, 101, 88, 81, 73, 69)
time<-seq(from=3,to=30,by=3)
plot(time,mite,type="b")
lmi<-log(mite)
plot(time,lmi,;ylab sim<-sample(lambda,20,replace=T)
N<-numeric(21)
N[1]<-100
for(t in 1:20) N[t+1]<-N[t]*sim[t]
plot(1:21,N,type="b",ylab="ln(mite)")
m<-lm(lmi~time)
coef(m)
exp(5.1991688)
log(0.5)/-0.0332169
weeks<-0:35
Nt<-69*exp(-0.033*weeks)
plot(weeks,Nt,type="b")
(log(69)-log(165))/27
```

6 Klíčové faktory

Popis

Ve vybraném biotopu byla po dobu několika generací sledována populace obaleče. Pro každou generaci byla stanovena mortalita všech stádií, způsobená třemi hlavními faktory. Ze zjištěných dat byly následně odhadnuty modely závislosti síly *k*-hodnot na početnosti populace v daném roce.

Data

V jednom roce byly zjištěny následující početnosti (N) stádií (x). V tabulce jsou uvedeny spolu s názvem faktoru způsobujícího největší mortalitu:

Stádium (x)	**N**	**Faktor**
Vajíčka	562	přezimování
Larvy	240	parazitoidi
Kukly	112	predátoři
Dospělci	64	

Výsledkem několikaletého pozorování byly zjištěny tyto odhady lineárních modelů k_x na $\log_{10}(N_x)$:

přezimování: $k_V = 0.48 - 0.04 \log(N_V)$
paraziti: $k_L = 0.55 - 0.09 \log(N_L)$
predátoři: $k_K = 0.30 - 0.03 \log(N_K)$

Úkoly

1/ Spočtěte *k*-hodnoty pro všechny faktory.

2/ Nasimulujte změnu velikosti populace po dobu 10 let, když víte, že poměr pohlaví je 1:1, průměrná plodnost samic je 17 vajíček, a počet vajíček je 562.

Řešení

1/ Odhady *k*-hodnot spočtěte podle vztahu:

$$k_x = \log(N_x) - \log(N_{x+1}) = \log\left(\frac{N_x}{N_{x+1}}\right).$$

R

Odhadnuté hodnoty zaznamenejte do tabulky:

k_V	
k_L	
k_K	

Který faktor způsobil největší mortalitu? ………………………

2/ Odvoďte analyticky model pro početnost dospělců (N_D), který bude vycházet z početnosti vajíček (N_V) a ve kterém budou zakomponovány všechny tři lineární modely pro *k*-hodnoty:

Výsledný model je ve tvaru

$$\log N_D = b + a \log N_V \quad \rightarrow \quad N_D = 10^{\{b + a \log N_V\}}$$

Doplňte hodnoty parametrů:

………………………………………………………………

Nasimulujte populační dynamiku s použitím výše uvedeného modelu po dobu 10 let (generací) s počáteční počtem vajíček 562 a plodností samic 17:

R

Překreslete graf do sešitu.

Je nárůst nebo pokles populace monotónní? Odůvodněte proč je nebo není.

………………………………………………………………………………………………………

………………………………………………………………………………………………………

Tahák

```
log10(562)-log10(240)
log10(240)-log10(112)
log10(112)-log10(64)
N<-numeric(11)
N[1]<-562
for(t in 1:10) N[t+1]<-10^(-1.405+1.168*log10(N[t]))/2*17
plot(1:11,N,type="b",xlab="years")
```