Moderní regresní metody

Petr Šmilauer
Biologická fakulta JU
České Budějovice
(c) 1998 - 2007

Obsah

Úvod

Regresní modely mají ve svém širším pojetí téměř výlučné postavení mezi statistickými metodami užívanými v různých vědních oboreh. Je to proto, že se snaží jednoduchým způsobem popsat vztahy mezi různými vlastnostmi (charakteristkami) studovaných objektů a procesů. Příkladem mohou být dva snad nejjednodušší modely. Prvním je regresní přímka, pomocí které je vztah mezi dvěma kvantitativně měřenými charakteristikami (Y a X) vyjádřen rovnicí $Y = a + b*X + \varepsilon$. Parametry (regresní koeficienty) *a* a *b* mají nějaké konkrétní číselné hodnoty, které se snažíme odhadnout na základě údajů, které jsme shromáždili (náš vzorek), zatímco symbol ε představuje stochastickou (nedeterministickou) část modelu.

Analýza variance se ve své nejjednodušší podobě (tzv. jednocestná ANOVA) liší od modelu regresní přímky jen tím, že v ní místo kvantitativní proměnné X používáme k vysvětlování hodnot proměnné Y kvalitativní proměnnou (faktor) X. Parametr *b* ovšem v takovém případě nemůže být jeden koeficient, můžeme si ale představit, že představuje odchylky průměrů Y u jednotlivých skupin (definovaných hodnotami faktoru X) od celkového průměru proměnné Y.

Představa, že mezi lineární regresí a analýzou variance není podstatný rozdíl, je jedním z prvních názorových posunů, kterými potřebujeme projít pro porozumění složitějším regresním metodám, v nichž jsou tyto jednodušší modely různým způsobem rozšiřovány, aby se zvýšila jejich realističnost a tím i použitelnost pro naše data.

Důležité pro to, abychom se naučili statistické modely používat, je i uvědomění si toho, že jde "jen o modely". Náš svět obsahuje sice spoustu zákonitostí, ale také spoustu náhodnosti, a nemůžeme očekávat, že předpovědi, založené na regresních modelech budou dokonalé. Přesto musíme vždy trochu pochybovat a i v okamžiku, kdy se dobereme nějakého výsledku, se musíme ohlédnout přes rameno a ptát se: zvolil jsem správné charakteristiky jako vysvětlující proměnné? Zvolil jsem vhodný druh modelu? Není model zbytečně složitý? Pro zodpovězení takovýchto otázek existují poměrně dobré postupy a my se alespoň s některými z nich v tomto textu seznámíme.

Tím se dostávám k asi nejpodstatnějšímu sdělení pro čtenáře úvodu (je ovšem pošetilé do úvodu něco podstatného dávat, protože zkušení čtenáři úvod skoro vždy přeskakují). Nejdůležitější kapitolou celé učebnice je kapitola první, která (skoro) vůbec nemluví o moderních regresních metodách, nýbrž o použití metod klasické lineární regrese a analýzy variance. Přesto si jsem jistý, že její zvládnutí je nezbytné pro čtenáře téměř kterékoliv z následujících kapitol (s výjimkou větší části kapitoly o analýze přežívání). Je to proto, že většina moderních postupů představovaných v této učebnici vychází ze společné obecné filozofie hledání, testování a ověřování statistických modelů a to se odráží i v obdobném způsobu jejich provedení v programu R. Tyto víceméně společné aspekty jsem umístil právě do kapitoly 1 a v dalších kapitolách se více zaměřuji na specifika jednotlivých metod.

Tato učebnice obsahuje nejen relativně malé množství statistické teorie (i když – uznávám – dosti velké na to, aby udolalo statného biologa, zejména při večerním čtení), ale také praktické příklady, povětšinou z oblasti přírodních věd. Příklady jsou řešeny pomocí metod přítomných v programu R. Zvládnutí tohoto software je možná o trochu

obtížnější než u programů s mnoha okénky a dialog boxy, odměnou jsou ale nečekané možnosti, které tento program nabízí. Úvod k práci s jazykem S, který program R používá, je v samostatném textu, v těchto skriptech základní znalost předpokládám. Příklady jsou vhodné pro užití s verzí 2.3 programu R, ale snad i s trochu staršími (2.x) verzemi a především s verzemi novějšími. Aktuální verze tohoto programu je v okamžiku publikace těchto skript k dispozici na webové adrese *http://www.cran.r-project.org* pod odkazem "Download and Install R".

Jakékoliv připomínky čtenářů zaslané na adresu *petrsm@jcu.cz* mne nejen nepopudí, ale ve většině případů i potěší.

Petr Šmilauer České Budějovice, 4. ledna 2007

1 Klasický lineární model a analýza variance

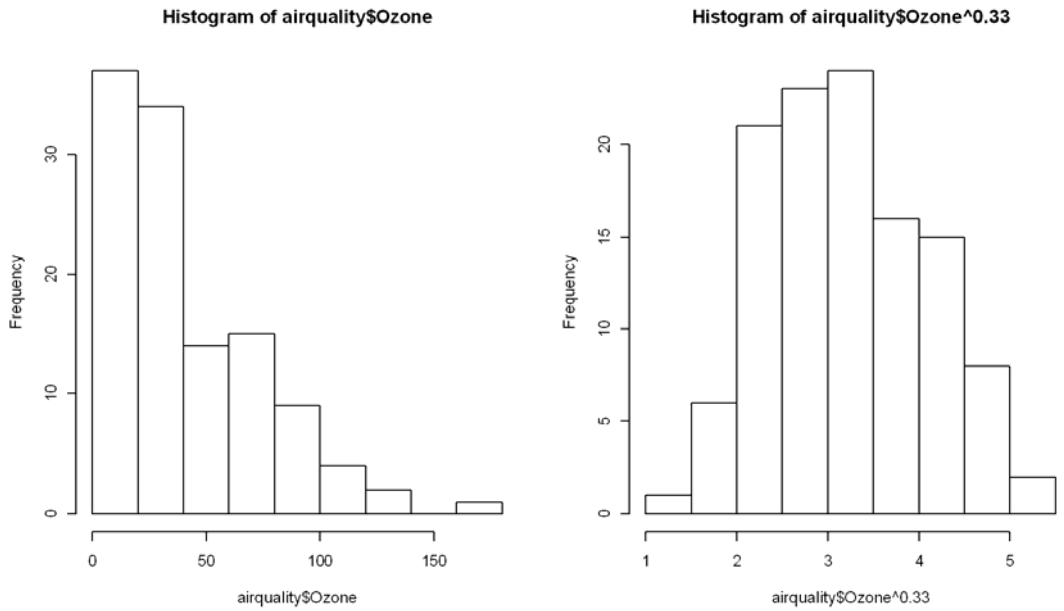
Motivační příklad

V tomto příkladu se ptáme, jak se mění koncentrace ozónu ve spodních vrstvách atmosféry (kde je považován za toxickou látku) v závislosti na meteorologických podmínkách. Datový rámec *airquality* obsahuje údaje o koncentraci ozónu, intenzitě slunečního záření, rychlosti větru, teplotě vzduchu, a také měsíci a dnu měření:

```
> data(airquality)
> summary(airquality)
     Ozone           Solar.R          Wind             Temp      
 Min.   :  1.00   Min.   :  7.0   Min.   : 1.700   Min.   :56.00  
 1st Qu.: 18.00   1st Qu.:115.8   1st Qu.: 7.400   1st Qu.:72.00  
 Median : 31.50   Median :205.0   Median : 9.700   Median :79.00  
 Mean   : 42.13   Mean   :185.9   Mean   : 9.958   Mean   :77.88  
 3rd Qu.: 63.25   3rd Qu.:258.8   3rd Qu.:11.500   3rd Qu.:85.00  
 Max.   :168.00   Max.   :334.0   Max.   :20.700   Max.   :97.00  
 NA's   : 37.00   NA's   :  7.0                                   
     Month            Day      
 Min.   :5.000   Min.   : 1.00  
 1st Qu.:6.000   1st Qu.: 8.00  
 Median :7.000   Median :16.00  
 Mean   :6.993   Mean   :15.80  
 3rd Qu.:8.000   3rd Qu.:23.00  
 Max.   :9.000   Max.   :31.00  
```

Koncentrace ozónu má velmi nesymetrickou distribuci (viz Obr. 1). Jde o množství částic v jednotce objemu a proto je asi vhodnější frekvenci ozónových molekul vztáhnout na jeden rozměr použitím třetí odmocniny, případně logaritmickou transformací (Obr. 1):

```
> par(mfcol=c(1,2))
> hist(airquality$Ozone)
> hist(airquality$Ozone^0.333)
> par(mfcol=c(1,1))
```



Obr. 1

Koncentraci ozónu tedy transformujeme navrženým způsobem (pozor, vytvoříme tak nezávislou proměnnou, sloupec *Ozone* v datovém rámci *airquality* se tím nezmění):

```
> ozone <- airquality$Ozone^0.333
```

Fitování klasického lineárního modelu

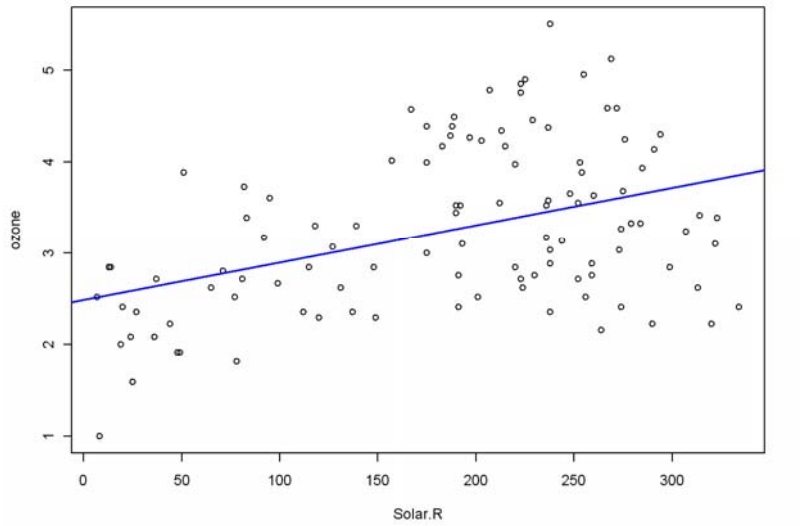
Nejvíce nás asi bude zajímat vztah mezi ozónem a radiací, která ke tvorbě ozónu funkčně přispívá. Model regresní přímky aplikujeme na naše data takto:

```
> lm.1<-lm(ozone~Solar.R,data=airquality)
```

Závislost (transformované) koncentrace ozónu na slunečním záření je vyjádřena vzorcem modelu, prvním parametrem funkce *lm*. Parametr *data* udává datový rámec, ve kterém R hledá proměnné, pokud je nenalezne v aktuálním prostředí (kde je např. proměnná *ozone*, kterou jsme vytvořili v předchozím kroku). Výsledek práce funkce *lm* (tj. údaje o nafitovaném lineárním modelu) jsme uložili do objektu, který jsme pojmenovali *lm.1*.

V případě přímkové regrese můžeme výsledný model znázornit velmi jednoduše, spolu s původními daty (při užití funkce *abline* by postačoval jen první parametr, zbylé dva mění vzhled vynášené přímky):

```
> plot(ozone~Solar.R,data=airquality)
> abline(lm.1,col="blue",lwd=2)
```



Obr. 2

Čtenář možná došel k názoru, že pro vztah mezi ozónem a radiací není lineární závislost nejlepším modelem: do hodnoty radiace kolem 230 koncentrace ozónu roste (strměji, než přímka naznačuje), pak spíše klesá (větším tempem, než rostla). Zatím budeme tento nesoulad tolerovat, s metodami popisujícími lépe takovéto závislosti se seznámíme později (viz loess smoother nebo zobecněné aditivní modely, GAM).

Na fitovaném regresním modelu nás asi nejprve budou zajímat jeho parametry, jejichž odhady se obvykle v odborných publikacích uvádějí. Tyto a další základní informace o modelu získáme použitím funkce *summary* na objekt představující regresní model:

```
> summary(lm.1)

Call:
lm(formula = ozone ~ Solar.R, data = airquality)

Residuals:
    Min      1Q  Median      3Q     Max
-1.5778 -0.5881 -0.1158  0.5972  2.0458

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 2.4836190  0.1742473   14.25  < 2e-16 ***
Solar.R     0.0041137  0.0008464    4.86 3.96e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.8091 on 109 degrees of freedom
Multiple R-Squared: 0.1781,     Adjusted R-squared: 0.1706
F-statistic: 23.62 on 1 and 109 DF,  p-value: 3.956e-06
```

Sekce nazvaná *Call* se nám může zdát nadbytečná (uvádí způsob, jakým byl objekt vytvořen), ale pokud budeme za několik týdnů či měsíců zkoumat, co je objekt s názvem *lm.1* zač, bude se hodit. Sekce *Residuals* shrnuje základní distribuční vlastnosti **residuálů**. Ty představují rozdíl mezi hodnotou, která byla pro dané pozorování **vysvětlované proměnné** (*ozone*) změřena, a hodnotou, kterou předpovídá nafitovaný

(odhadnutý) model regresní přímky, tedy vertikální pozice přímky nad hodnotou **vysvětlující proměnné** *Solar.R* pro dané pozorování. Tato předpovídaná (nebo také **fitovaná**) hodnota (predicted value, fitted value) se označuje písmenem *y*, nad kterým je stříška. Pokud tedy residuál označíme písmenkem *e*, můžeme vztah mezi těmito třemi hodnotami (pozorovaná hodnota, fitovaná hodnota a regresní residuál) popsat touto rovnicí:

$$y_i = \hat{y}_i + e_i$$

Fitovanou hodnotu vysvětlované proměnné získáme dosazením do rovnice, která má pro jednoduchou přímkovou regresi podobu

$$\hat{y}_i = b_0 + b_1 * x$$

kde *x* je hodnota vysvětlující proměnné, tj. změřená intenzita slunečního záření, zatímco b_0 a b_1 jsou **odhady regresních koeficientů** daného modelu. Ty jsou zobrazeny ve shrnutí modelu (výstup funkce *summary* výše) ve sloupci *Estimate* sekce *Coefficients*. Jinými slovy, fitovanou (předpovídanou) hodnotu transformované koncentrace ozónu získáme pro tento model tak, že známou hodnotu slunečního záření násobíme hodnotou *0.0041137* a k výsledku přičteme *2.4836190*. Hodnota b_0 je ve výstupu označena slovem *(Intercept)*, protože není spojena s žádnou konkrétní proměnnou a v diagramu představuje průsečík (intercept) přímky se svislou osou (tj. fitovanou hodnotu ozónu pro nulovou hodnotu radiace).

Jak již bylo naznačeno, koeficienty b_0 a b_1 jsou odhady parametrů pro základní populaci ("populace" ve statistickém smyslu), získané na základě pozorování, která jsme shromáždili a použili k fitování modelu. Pro skutečné (a neznámé) parametry se model jednoduché lineární regrese vyjadřuje takto:

$$Y_i = \beta \ + \beta_1 X_i + \varepsilon_i$$

k čemuž statistici často doplňují

$$\varepsilon_i \sim N(0, \sigma^2)$$

aby vyjádřili, že stochastická (neurčená) část modelu se dá popsat svými distribučními vlastnostmi: dá se charakterizovat normální (Gaussovou) distribucí s nulovým průměrem a konstantní variancí.

Hodnota vysvětlované proměnné, predikovaná modelem, se v rovnicích odkazujících na základní statistickou populaci neoznačuje písmenem *y* se stříškou, nýbrž jako *Ey*, případně *E(y)*, písmeno E zde odkazuje na slovo *expected* (očekávaná [hodnota]).

Rozdíl mezi regresními koeficienty β a jejich odhady *b* se stává podstatným, jakmile začneme testovat jednotlivé parametry regresního modelu. Test absolutního členu (β_0) nebývá obvyklý, takže v tomto jednoduchém modelu se zaměříme na test sklonu regresní přímky. Testovaná nulová hypotéza

$$H_0 : \beta_1 = 0$$

se týká vlastnosti základní populace (tj. nezajímá nás vztah mezi ozónem a radiací v rámci konkrétních 110 dnů pozorování, ale pro všechny možné dny – i ty bez pozorování). Ostatně testovat, že náš odhad b_1 je různý od nuly postrádá smysl: vidíme, že nemá nulovou hodnotu, ta však může být jen důsledkem toho, že pracujeme s výběrem omezené velikosti. Výše uvedenou hypotézu můžeme testovat buď pomocí T statistiky, uvedené ve výstupu funkce *summary* (tam je i odpovídající hladina signifikance), nebo pomocí F testu, který bude probrán níže.

Ještě se vrátíme k poslední části výstupu z funkce *summary*, aplikované na objekt obsahující nafitovaný lineární model. Pro přehlednost tuto část uvádím znovu:

```
Multiple R-Squared: 0.1781,     Adjusted R-squared: 0.1706
F-statistic: 23.62 on 1 and 109 DF,  p-value: 3.956e-06
```

Hodnota označená jako *Multiple R-Squared* představuje takzvaný **koeficient determinace** (coefficient of determination), který může nabývat hodnot v rozsahu od nuly do jedničky. Po vynásobení 100 jej můžeme považovat za procento z celkové variability hodnot vysvětlované proměnné, které nám nafitovaný regresní model objasňuje. V případě jednoduché regrese (pouze s jednou vysvětlující proměnnou) lze tedy nejen říct, že nám změřené hodnoty slunečního záření vysvětlují (při použití modelu lineární závislosti) asi 18% z variability v hodnotách koncentrace ozónu, ale také platí, že hodnota 0.1781 je druhou mocninou hodnoty lineární korelace mezi těmito dvěma proměnnými (*ozone* a *Solar.R*). Problém s koeficientem determinace (jehož výpočet

popíšu níže) je v tom, že není zrovna nejlepším odhadem pro základní statistickou populaci a je značně ovlivňován počtem vysvětlujících proměnných, velikostí výběru a také vztahem mezi těmito dvěma veličinami.[1] Proto je doporučován tzv. **upravený koeficient determinace** (adjusted coefficient of determination), obvykle označovaný jako R^2_{adj}. Jeho hodnota je uvedena ve stejném řádku a protože náš regresní model má jen jednu vysvětlující proměnnou a vychází z poměrně velkého počtu pozorování, tyto dva odhady se od sebe příliš neliší.

Analýza variance regresního modelu

Uváděná F statistika vyplývá z tzv. **analýzy variance regresního modelu** (analysis of variance of the regression model). Zde musím upozornit na terminologický problém, protože se označení analýza variance (obvykle s akronymem ANOVA) používá i pro samostatný statistický model, ve kterém jedna nebo více kvalitativních proměnných (faktorů) slouží k vysvětlování hodnot kvantitativní proměnné. Tento terminologický překryv ale není čistě náhodný, jde o obdobný postup. Vpodstatě jediný rozdíl je v tom, že v klasické analýze variance (s faktory) se o odhadnuté hodnoty regresních parametrů obvykle přímo nezajímáme a soustřeďujeme se právě na rozklad variancí.

V tomto rozkladu se snažíme určit, jak velkou část z celkové variability hodnot vysvětlované proměnné jsme vysvětlili pomocí našeho modelu a jak velká část zůstala neobjasněna. Rozklad provádíme s použitím tzv. **sumy čtverců** (sum of squares, SS), další výpočty včetně testu signifikance ale provádíme s variancemi, které se zde tradičně označují také jako střední čtverce (mean squares).

Celkovou sumu čtverců (total sum of squares, TSS) vypočteme jako součet druhých mocnin odchylek pozorovaných hodnot vysvětlované proměnné od jejího průměru:

$$TSS = \sum_{i=1}^{n} \left(y_i - \bar{y} \right)^2$$

Pokud bychom tuto hodnotu vydělili počtem pozorování zmenšeným o jednotku ($n-1$), dostaneme známý odhad variance vysvětlované proměnné y. Všimněme si ještě, že celková suma čtverců není nijak ovlivněna zvoleným regresním modelem a že tedy zůstává stejná pro různé modely, pokud vysvětlují hodnoty téže proměnné.

Celková suma čtverců nám tedy ukazuje míru rozptylu pozorovaných hodnot kolem průměru. Pokud teď nahradíme pozorované hodnoty těmi, které pro jednotlivá pozorování předpovídá náš regresní model (tj. fitovanými hodnotami), dostáváme **modelovou sumu čtverců** (model sum of squares, MSS), která ukazuje míru rozptylu předpovídaných hodnot kolem průměru, tedy variabilitu objasněnou modelem:

[1] To, že klasický koeficient determinace nefunguje dobře, pokud není počet pozorování (n) výrazně větší než počet parametrů modelu (tj. obvykle počet vysvětlujících proměnných, p) si můžeme dobře ukázat na extrémní situaci, ve které by počet parametrů byl roven počtu pozorování. Takový model by plně vysvětlil (přesně předpovídal) všechny hodnoty vysvětlované proměnné, aniž by to vypovídalo cokoliv o jeho kvalitě.

$$MSS = \sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2$$

Residuální sumu čtverců (residual sum of squares, RSS) můžeme určit dopočtem (TSS je rovna součtu MSS a RSS, a tedy RSS = TSS – MSS), ale je dobré vědět, že ji lze vypočíst i takto:

$$RSS = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

S obsahem závorky v předcházejícím vzorečku jsme se již potkali: rozdíl mezi pozorovanou (skutečnou) a předpovídanou (fitovanou) hodnotou vysvětlované proměnné jsme nazvali regresní residuál a residuální suma čtverců tedy představuje druhé mocniny těchto residuálů, sečtené přes všechna pozorování.

Než pokročíme dále, provedeme analýzu variance pro již nafitovaný regresní model, pomocí funkce *anova*:

```
> anova(lm.1)
Analysis of Variance Table

Response: ozone
           Df Sum Sq Mean Sq F value    Pr(>F)
Solar.R     1 15.467  15.467  23.624 3.956e-06 ***
Residuals 109 71.362   0.655
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Dvě sumy čtverců jsou zobrazeny ve sloupečku *Sum Sq*: modelová suma čtverců má hodnotu 15.467, zatímco residuální suma čtverců má hodnotu 71.362. Celková suma čtverců není zobrazena, ale je součtem těchto dvou hodnot.

Poměr modelové sumy čtverců ku celkové sumě čtverců představuje podíl variability objasněné modelem na celkové variabilitě vysvětlované proměnné a měl by tedy být roven neupravenému koeficientu determinace:

```
> 15.467/(15.467+71.362)
[1] 0.1781317
```

což se plně shoduje s hodnotou 0.1781 uváděnou ve výstupu z funkce *summary*.

Pro výpočet F statistiky, kterou používáme v testu regresního modelu, musíme (na rozdíl od koeficientu determinace) pracovat nikoliv se sumami čtverců, ale s variancemi (průměrné čtverce, mean squares), které jsou uváděny ve výstupu funkce *anova* ve sloupečku *Mean Sq*. Tyto variance získáme vydělením hodnot součtu čtverců odpovídajícími stupni volnosti (uvedeny ve sloupci *Df*). V případě modelové sumy čtverců odpovídá počet stupňů volnosti počtu koeficientů v modelu s vyloučením absolutního členu b_0, v našem případě tedy hodnotě 1. Celkový počet stupňů volnosti (DF, z degrees of freedom) je roven *n-1* a je součtem počtu stupňů volnosti modelu

a počtu DF residuální variability. Aby si čtenář mohl ověřit své pochopení, uvádím, že počet pozorování použitých pro odhad modelu byl roven 111.[2]

Můžeme nyní spočítat poměr modelového průměrného čtverce a residuálního průměrného čtverce (15.467/0.655 = 23.624, viz výstup funkce *anova* výše). Za platnosti nulové hypotézy, která tvrdí, že neexistuje lineární vztah mezi vysvětlovanou proměnnou (*ozone* v našem příkladě) a proměnnými vysvětlujícími (zde jen *Solar.R*), pochází (při splnění některých předpokladů, viz níže) tento poměr z F distribuce s parametry 1 a 109 (pro náš příklad). Pro hodnotu 23.624 je ale pravděpodobnost, že jsme takovou hodnotu "vytáhli" z této F distribuce, menší než 0.000004, a proto nulovou hypotézu zamítáme.

Pro model s více vysvětlujícími proměnnými rozkládá funkce *anova* modelovou sumu čtverců na příspěvky jednotlivých proměnných a zobrazuje odpovídající dílčí testy založené na F statistice. Naproti tomu funkce *summary* zobrazuje na konci svého výstupu vždy celkový F test, společně hodnotící všechny vysvětlující proměnné.

Dosud jsme na objekt vrácený funkcí *lm* používali funkce, které jej z toho či onoho pohledu charakterizovaly. Na objekty představující různé typy regresních modelů ale můžeme aplikovat i různé tzv. extrakční funkce, které (jak název napovídá) z daného objektu získávají (často výpočtem) konkrétní číselnou informaci, se kterou lze dále pracovat. V následujících příkladech použijeme funkce *fitted* (získávající fitované, tj. předpovídané hodnoty modelu) a *resid* (což je pro uživatelské pohodlí zkrácená forma názvu funkce *residuals*, extrahuje samozřejmě regresní residuály).

```
> lm.1.fit<-fitted(lm.1)
> lm.1.res<-resid(lm.1)
> summary(lm.1.fit)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  2.512   2.951   3.335   3.244   3.535   3.858
> summary(ozone)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
  1.000   2.618   3.154   3.246   3.979   5.508  37.000
> summary(lm.1.res)
      Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
-1.578e+00 -5.881e-01 -1.158e-01  4.911e-18  5.972e-01  2.046e+00
>
```

Porovnejme sumarizaci fitovaných hodnot s podobnou sumarizací původní vysvětlované proměnné. Není náhodou, že jsou jejich aritmetické průměry (*Mean*) tak podobné. Měly by být ve skutečnosti shodné, nicméně proměnná *ozone* obsahuje několik hodnot, které funkce *lm* nepoužila, protože neměla k dispozici odpovídající hodnoty proměnné *Solar.R*. Rozsah fitovaných hodnot je také trochu menší než rozsah skutečných koncentrací ozónu. To je pochopitelné, protože fitované hodnoty představují *průměrnou* hodnotu, kterou pro ozón regresní model předpovídá při určité intenzitě záření. Můžeme také říci, že obsah prvých dvou sumarizovaných proměnných (tj. *lm.1.fit* a *ozone*) je protějškem modelové sumy čtverců a residuální sumy čtverců, podobně jako proměnná *lm.1.res* (obsahující regresní residuály) odpovídá residuální sumě čtverců. Lze to ukázat ještě jinak:

```
> na.vals<-is.na(ozone)|is.na(airquality$Solar.R) #pozorov. s chyběj. hodnotami
```

[2] Zvídavý čtenář možná zjistí, že datový rámec *airquality* obsahuje 153 pozorování, ne 111. Nicméně, jsou zde i chybějící údaje (*NA*) a funkce *lm* automaticky vyloučí ta pozorování, u kterých pro jednu nebo více proměnných údaj chybí.

```
> sum(na.vals)   # kolik chybějících hodnot?
[1] 42
> cor(lm.1.fit,ozone[!na.vals])^2
[1] 0.1781271
> cor(lm.1.res,ozone[!na.vals])^2
[1] 0.821873
```

Druhá mocnina z lineární korelace mezi fitovanými a skutečnými hodnotami koncentrace ozónu je rovna koeficientu determinace (neupravenému), tak jak nám jej pro objekt *lm.1* zobrazuje funkce *summary*. Podobně druhá mocnina korelace mezi residuály a skutečnými hodnotami *ozone* vyjadřuje velikost té části variability, kterou jsme modelem nevysvětlili, a sčítá tudíž vždy do 1.0 s koeficientem determinace.

Důvodem ke "cvičení" s chybějícími hodnotami (konstrukce vektoru *na.vals* a jeho užití pro výběr hodnot *ozone*) je to, že funkce *lm* odstranila při fitování modelu *lm.1* všechna pozorování, ve kterých chyběla hodnota vysvětlované (*ozone*) **nebo** vysvětlující (*Solar.R*) proměnné. Proto jsou vektory s fitovanými hodnotami a residuály modelu kratší než výchozí proměnná *ozone* a bez těchto úprav je nelze spolu korelovat. Možná jednodušší by ale bylo odstranit z výchozí tabulky dat všechna pozorování s alespoň jednou chybějící hodnotou (*aq <- na.omit(airquality)*) a teprve pak provádět jakékoliv analýzy.

Parciální F test

K testu hypotézy, týkající se vlivu konkrétní vysvětlující proměnné, můžeme použít tzv. dílčí F test (partial F test), ve kterém se porovnávají dva modely, lišící se právě jen v přítomnosti či absenci testované proměnné. To si můžeme ukázat na následujícím příkladě.

Zajímá mne, jestli by koncentraci ozónu (přesněji řečeno její třetí odmocninu) nepředpovídal lépe lineární regresní model, ve kterém by byla použita jak informace o intenzitě slunečního záření, tak o teplotě vzduchu. Takový model vytvoříme v programu R následovně:

```
> lm.2<-lm(ozone~Solar.R+Temp,data=airquality)
```

Stejný model může ale vytvořit i rozšířením modelu *lm.1*, s pomocí funkce *update*:

```
> lm.2x<-update(lm.1,.~.+Temp)
```

Takový zápis říká "uprav existující model *lm.1* tak, že vysvětlovaná proměnná zůstane stejná (.) a k stávajícím vysvětlujícím proměnným (jen *Solar.R* pro *lm.1*) přidáš *Temp*. A nyní můžeme porovnat modely *lm.1* a *lm.2*, překvapivě opět pomocí funkce *anova*:

```
> anova(lm.1,lm.2)
Analysis of Variance Table

Model 1: ozone ~ Solar.R
Model 2: ozone ~ Solar.R + Temp
  Res.Df    RSS  Df Sum of Sq      F    Pr(>F)
1    109 71.362
2    108 33.758   1    37.605 120.31 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Tabulka (s řádky uvedenými čísly 1 a 2, odkazujícími na dva srovnávané modely) nám ukazuje, že přidáním proměnné *Temp* poklesla residuální suma čtverců (neobjasněná

variabilita v hodnotách proměnné *ozone*) ze 71.362 na 33.758 (tj. o 37.605, jak tabulka také ukazuje). O stejný objem se tudíž zvětšila modelová suma čtverců, protože celková suma čtverců je pro oba modely stejná (nezávisí na obsahu modelu). Změnu v modelové sumě čtverců můžeme (po převedení na mean square za pomoci počtu DF, o které se zvýšila složitost modelu) použít k výpočtu F statistiky, potřebujeme ji ale vydělit residuálním středním čtvercem (residuální variancí). Tu nám ale funkce *anova* při tomto použití (porovnání modelů) nezobrazuje. Je to residuální variance složitějšího z obou porovnávaných modelů:

```
> anova(lm.2)
Analysis of Variance Table

Response: ozone
           Df Sum Sq Mean Sq F value    Pr(>F)
Solar.R     1 15.467  15.467  49.482 1.914e-10 ***
Temp        1 37.605  37.605 120.307 < 2.2e-16 ***
Residuals 108 33.758   0.313
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> 37.605/0.313
[1] 120.1438
```

Chyba na čtvrté pozici (tj. v desetinách) je způsobena tím, že hodnoty užité v podílu jsou zaokrouhleny. Zobrazení výstupu funkce *anova* s jediným modelem, *lm.2*, také ukazuje, jak je modelová suma čtverců rozdělena mezi dvě vysvětlující proměnné (*Solar.R* a *Temp*), podobně jako klasická ANOVA se dvěma faktory (tzv. dvoucestná ANOVA) rozděluje objasněnou variabilitu mezi tyto faktory. Nicméně v našem příkladě (a také ve většině jiných dat pro mnohonásobnou regresi) toto rozdělení nefunguje tak dobře, jak bychom doufali. Důvodem je, že na rozdíl od dvoucestné ANOVA počítané obvykle pro experimentální data s balancovaným designem (tj. stejný počet opakování pro všechny kombinace hladin obou faktorů), kvantitativní vysvětlující proměnné jsou spolu obvykle korelovány. Variabilita vysvětlená jednotlivými proměnnými se proto zčásti překrývá, takže záleží na tom, "kdo si ukousne první". Podívejme se na výsledek u skoro stejného modelu, jen se dvěma vysvětlujícími proměnnými zadanými v opačném pořadí:

```
> lm.2b<-lm(ozone~Temp+Solar.R, data=airquality)
> anova(lm.2b)
Analysis of Variance Table

Response: ozone
           Df Sum Sq Mean Sq F value    Pr(>F)
Temp        1 49.247  49.247 157.556 < 2.2e-16 ***
Solar.R     1  3.824   3.824  12.233 0.0006828 ***
Residuals 108 33.758   0.313
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Po změně pořadí našich dvou vysvětlujících proměnných v modelu sice pořád platí, že model obsahující obě je lepší, než model jen s jednou z nich, ale význam proměnné *Solar.R* se zmenšil, jak pokud jde o absolutní objem vysvětlené sumy čtverců (propad z 15.467 na 3.824), tak v odpovídající F statistice.

Je důležité, abychom se s tímto omezením naučili žít. Jakmile totiž přejdeme ke složitějším modelům, u kterých se nepředpokládá normální distribuce nevysvětlené variability (tj. ke zobecněným lineárním modelům), přestane nezávislost (a tedy

jednoznačná oddělitelnost) příspěvků jednotlivých vysvětlujících proměnných platit i pro data pocházející z balancovaně uspořádaných experimentů. Při výběru vhodného a přitom úsporného modelu (parsimonious model) je proto vhodné brát zřetel i na pořadí volby vysvětlujících proměnných a začínat s těmi nejvlivnějšími. To je přístup užívaný například v metodě postupného výběru obsahu modelu (*stepwise selection* nebo též *forward selection*).

Postupný výběr

Metoda postupného výběru je sice založena na porovnání dvou modelů (jednoduššího s jiným, lišícím se pouze rozšířením o jednu vysvětlující proměnnou), které jsme si ukázali výše (při porovnání modelů *lm.1* a *lm.2* pomocí funkce *anova*), obvykle ale začíná od tzv. **nulového modelu**. Ten představuje nepříliš optimistickou hypotézu, že k vysvětlení hodnot vysvětlované proměnné nemůžeme udělat nic více, než je považovat za náhodně rozptýlené kolem konstantní hodnoty. Tuto hodnotu v základní statistické populaci představuje střední hodnota a pro náš omezený výběr pozorování ji obvykle odhadujeme aritmetickým průměrem (alespoň v případě klasické regrese či ANOVA modelů). Nulový model pro naše ukázková data nafitujeme takto (nepodstatné části výstupu funkce *summary* jsou vynechány):

```
> lm.0<-lm(ozone~+1,data=airquality)
> summary(lm.0)
[...]
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.24638    0.08228   39.45   <2e-16 ***

[...]
```

Výraz *+1* na pravé straně rovnice modelu představuje absolutní člen, který nemusíme uvádět, pokud se krom něj vyskytují v rovnici i nějaké vysvětlující proměnné. Předpovídaná střední hodnota koncentrace ozónu je tedy *3.24638*, měla by být tedy stejná jako ta, kterou nám vrátí funkce *mean*:

```
> mean(ozone)
[1] NA
```

Problém je opět v tom, že v datovém rámci *airquality* (a tím i v námi transformované samostatné proměnné *ozone*) jsou chybějící hodnoty. Musíme proto explicitně říci, že je chceme odstranit, než spočítáme aritmetický průměr:

```
> mean(na.omit(ozone))
[1] 3.246381
```

Nejjednodušší lineární model, vysvětlující koncentraci ozónu a přitom složitější než tento model nulový, bude samozřejmě obsahovat jednu vysvětlující proměnnou. Takovým modelem jsme se již zabývali (model *lm.1*), nicméně výsledky z předchozí sekce naznačují, že stejně jednoduchý model užívající proměnnou *Temp* místo proměnné *Solar.R* by asi vysvětlil více z variability v hodnotách proměnné *ozone*. Nezabývali jsme se ještě ale tím, kolik variability nám může objasnit třetí z proměnných, které máme k dispozici pro vysvětlování koncentrace ozónu – rychlost větru v proměnné *Wind*. Skutečnost je taková, že rychlost větru není pro vysvětlení koncentrace ozónu lepší než

teplota vzduchu, jak si čtenář může ověřit pomocí funkcí *lm* a *summary*. Model obsahující pouze proměnnou *Temp* nafitujeme takto:

```
> lm.1.Temp<-lm(ozone~Temp,data=airquality)
```

V dalším kroku bychom tento model chtěli porovnat s modelem se dvěma vysvětlujícími proměnnými (prediktory), tedy buď *Temp* + *Wind* nebo *Temp* + *Solar.R*. Druhý z nich již máme definován v objektu *lm.2b* (který má na rozdíl od *lm.2* správné pořadí prediktorů v modelu), první variantu získáme takto (pro připomenutí použijeme alternativní postup výpočtu - s funkcí *update*):

```
> lm.2c<-update(lm.1.Temp,.~.+Wind)
```

a množství variability objasněné tímto modelem odhadneme opět s pomocí funkce *anova*:

```
> anova(lm.2c)
Analysis of Variance Table

Response: ozone
           Df Sum Sq Mean Sq F value    Pr(>F)
Temp        1 50.770  50.770 165.540 < 2.2e-16 ***
Wind        1  4.895   4.895  15.960 0.0001156 ***
Residuals 113 34.656   0.307
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Vzhledem k tomu, že u modelu *lm.2b* byla residuální (nevysvětlená) suma čtverců rovna *33.758* (viz předchozí sekce) a zde je *34.656*, mohli bychom usoudit, že intenzita záření je pro doplnění teploty vzduchu lepším prediktorem než rychlost větru. Není to ale závěr správný, pokud si všimneme počtu residuálních stupňů volnosti. V modelu užívajícím radiaci byl roven 108, v modelu zahrnujícím rychlost větru je 113: modely, zdá se, pracovaly s odlišným počtem pozorování! Objasnění těchto obtíží je opět v chybějících hodnotách. Vrátíme-li se k výstupu z funkce *summary* použité na datový rámec *airquality*, vidíme, že proměnné *Temp* a *Wind* žádné chybějící hodnoty nemají, zatímco proměnná *Solar.R* jich má sedm. Pokud by těchto sedm případů bylo podmnožinou 37 pozorování, u kterých chybí údaj o koncentraci ozónu, problém by nevznikl, ale bohužel tomu tak není. Funkce *lm* vynechá vždy jen ta pozorování, pro které chybí nějaká hodnota, a to pouze v případě, že jde o hodnoty v proměnných použitých v daném modelu.

Správným řešením by zde bylo před výběrem obsahu modelu vytvořit kopii datového rámce, ve které by byla jen pozorování, pro které údaje nechybí pro žádnou z proměnných, tedy například takto:

```
> airqual<-na.omit(airquality)
```

a výsledný datový rámec nadále používat (viz níže, použití funkce *step*). Zde problém přibližně vyřešíme takovým porovnáním modelů, které není rozdílným počtem stupňů volnosti přímo ovlivněno, tj. například pomocí hodnoty *Mean Sq*[uare] v řádku *Residuals* nebo pomocí hodnoty F statistiky v řádku prediktoru, ve kterém se oba modely liší. Obě porovnání vedou k závěru, že volba rychlosti větru je lepší, ve srovnání s intenzitou záření.

V tomto okamžiku náš výběr modelu zastavíme, i když v praxi bychom se museli ještě zajímat o to, zda model obsahující všechny tři prediktory není lepší než model uložený v objektu *lm.2c*[3], a náš předchozí postup si shrneme ve funkci *anova*:

```
> anova(lm.0,lm.1.Temp,lm.2c)
Analysis of Variance Table

Model 1: ozone ~ +1
Model 2: ozone ~ Temp
Model 3: ozone ~ Temp + Wind
  Res.Df    RSS  Df Sum of Sq       F    Pr(>F)
1    115 90.320
2    114 39.551   1    50.770 165.540 < 2.2e-16 ***
3    113 34.656   1     4.895  15.960 0.0001156 ***
```

Náš "ruční" výběr obsahu modelu sice umožňuje důkladné porozumění studovaným vztahům (zejména pokud je vhodně doplněn grafickými výstupy), ale pro větší počet vysvětlujících proměnných, které často máme pro tvorbu modelu k dispozici, ale může být volba modelu zdlouhavá. Více automatický přístup poskytuje funkce *step*. Než se s ní seznámíme, je třeba uvést anglický termín **parsimony**. Ten by se při hledání obsahu statistických modelů dal nejspíše popsat českým slovem "úspornost" nebo spíše "účelná úspornost". Vyjadřuje totiž takovou hledanou vlastnost modelu, která váží jeho schopnost předpovídat hodnoty vysvětlované proměnné proti jeho složitosti. Balance mezi přesností a jednoduchostí modelu se odráží i v definici statistiky, která se pro vyjádření úspornosti modelu používá nejčastěji. Je to tzv. AIC statistika, zkratka odpovídá termínu *Akaike Information Criterion*. Přesný postup výpočtu není podstatný, ale je důležité vědět, že hodnota AIC je výsledkem součtu dvou členů, z nichž první je úměrný logaritmu residuální sumy čtverců[4], zatímco druhů je úměrný složitosti modelu (počtu jeho členů). Z toho vyplývá skutečnost, že nejúspornější modely mají nejnižší hodnotu AIC statistiky. AIC statistiku lze při volbě modelu použít buď tak, že ji spočteme pro všechny možné modely (tj. se všemi možnými kombinacemi potenciálních vysvětlujících proměnných) a vybereme model s nejnižší hodnotou, nebo ji použijeme v rámci metody postupného výběru.

Zde si ukážeme ten druhý způsob, i když alespoň v případě omezeného počtu možných prediktorů je metoda výčtu všech možných modelů spolehlivější. K postupnému výběru s použitím AIC statistiky slouží v programu R funkce *step*. Pro naše příkladová data bude vhodné pracovat s rámcem, ve kterém se již nevyskytují chybějící hodnoty (NA), jehož vytvoření jsme si ukázali výše. Nicméně musíme obdobně upravit i samostatně existující proměnnou *ozone*:

```
> ozone<-airqual$Ozone^0.333
> lm.0<-update(lm.0,data=airqual)
> lm.aic<-step(lm.0,scope=~Solar.R+Wind+Temp)
```

[3] V ukončení výběru v tomto kroku se odráží i autorova vychytralost, protože funkce *anova* by se bránila porovnat model *lm.2c* s modelem se všemi třemi prediktory, právě díky odlišnému počtu residuálních stupňů volnosti.

[4] V případě klasických lineárních modelů. AIC je definována primárně pro odhad parametrů modelů pomocí obecnější metody maximální věrohodnosti (maximum likelihood) a v této podobě se používá například u zobecněných lineární modelů (GLM).

```
Start:  AIC= -25.26
 ozone ~ +1
          Df Sum of Sq      RSS      AIC
+ Temp     1    49.247   37.581 -116.215
+ Wind     1    31.114   55.715  -72.511
+ Solar.R  1    15.467   71.362  -45.036
<none>                   86.829  -25.261

Step:  AIC= -116.22
 ozone ~ Temp

          Df Sum of Sq      RSS      AIC
+ Wind     1     5.796   31.785 -132.809
+ Solar.R  1     3.824   33.758 -126.126
<none>                   37.581 -116.215
- Temp     1    49.247   86.829  -25.261

Step:  AIC= -132.81
 ozone ~ Temp + Wind

          Df Sum of Sq      RSS      AIC
+ Solar.R  1     4.045   27.740 -145.919
<none>                   31.785 -132.809
- Wind     1     5.796   37.581 -116.215
- Temp     1    23.930   55.715  -72.511

Step:  AIC= -145.92
 ozone ~ Temp + Wind + Solar.R

          Df Sum of Sq      RSS      AIC
<none>                   27.740 -145.919
- Solar.R  1     4.045   31.785 -132.809
- Wind     1     6.018   33.758 -126.126
- Temp     1    17.414   45.154  -93.839
```

Každá z uvedených tabulek shrnuje důsledky změn, které bychom provedli v modelu, jehož obsah je stručně popsán na tabulkou. Všimněte si, že jednou z možných změn je i "<none>", tj. žádná změna, umožňující porovnání změny se současným stavem. Hned v prvním kroku je vidět, že přidání kterékoliv ze tří možných proměnných do nulového modelu sníží hodnotu jeho AIC, tj. zvýší jeho kvalitu, nicméně největší pokles je pro proměnnou *Temp* a ta je proto vybrána. Od dalšího kroku je vidět, že funkce *step* testuje i možnost, že by složitějšímu modelu prospělo vypuštění některé z dříve vybraných vysvětlujících proměnných. I když to na první pohled vypadá jako nesmysl, má tento krok svůj význam. Je to proto, že jednotlivé potenciální vysvětlující proměnné jsou spolu obvykle ve větší či menší míře korelovány, což vytváří složité nelineární interakce mezi jejich rolemi v modelu, a současně funkce *step* nezkouší všechny možné kombinace prediktorů, jen jejich podmnožinu, protože z možných proměnných vybírá v každém kroku jen tu "nejlepší".

Náš výsledný model (ke stejnému modelu bychom se v případě těchto dat dobrali i postupným výběrem užívajícím parciální F test) obsahuje všechny tři vysvětlující proměnné, *Temp*, *Wind* i *Solar.R*. Pokud si tento model shrneme nejjednodušším způsobem (tj. vypsáním jeho obsahu), jazyk S nám zobrazí především čtyři regresní koeficienty:

```
> lm.aic
Call:
lm(formula = ozone ~ Temp + Wind + Solar.R, data = airqual)
```

```
Coefficients:
(Intercept)         Temp         Wind      Solar.R
  -0.295442     0.049950    -0.075792     0.002202
```

Jak jejich zobrazené názvy naznačují, kombinují se tyto koeficienty v modelu s jednotlivými vysvětlujícími proměnnými. Pro naše konkrétní data bychom předpovídali hodnoty vysvětlované proměnné (koncentrace ozónu) pomocí rovnice:

-0.295442 + 0.049950*Temp – 0.075792*Wind + 0.002202*Solar.R

Tento způsob kombinace jednotlivých vysvětlujících proměnných v mnohonásobné regresi se nazývá **lineární kombinace vysvětlujících proměnných**. První koeficient (intercept) se s žádnou proměnnou nekombinuje, ale můžeme si představit, že mu odpovídá proměnná, která má hodnotu 1 pro všechna naše pozorování.

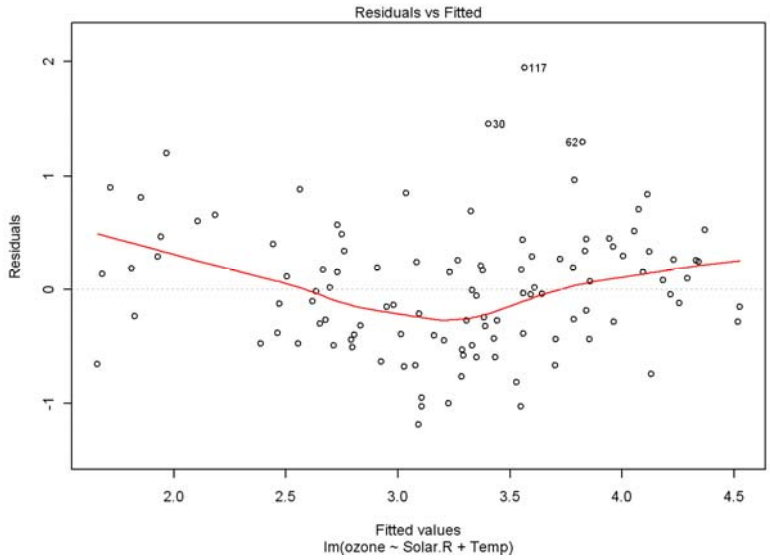
Na závěr této části si neodpustím varování před nadbytečným užíváním postupného výběru. Ať již vybíráme pomocí testování parciální F statistiky nebo pomocí AIC, metoda je citlivá k situaci, kdy máme velký počet vysvětlujících proměnných, ze kterých chceme vybrat, a to zejména pokud jsou mezi některými z nich velké korelace. Uvážlivost při volbě měřených charakteristik na základě znalosti studovaných jevů je dobrým počátečním krokem pro jejich popis pomocí statistických modelů.

Regresní diagnostika

Poslední částí povídání o klasických lineárních modelech (fitovaných pomocí funkce *lm*) bude stručná informace o tom, jak vytvořené modely kriticky posuzovat. Základním nástrojem může být použití funkce *plot* (zde se vracím k našemu původnímu jednoduchému modelu):

```
> plot(lm.1)
```

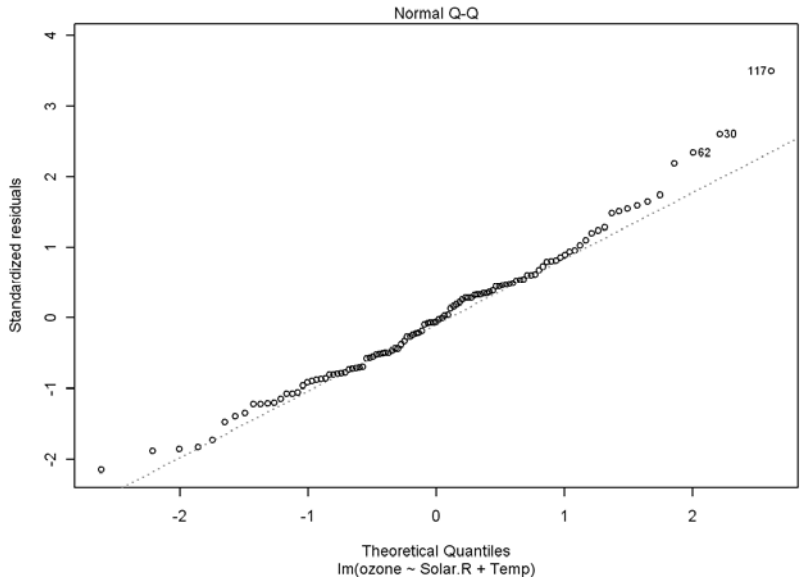
Pokud je takto volána, funkce *plot* zobrazí čtyři ze šesti možných diagramů pro tzv. regresní diagnostiku (posouzení předpokladů a kvality zvoleného lineárního modelu). Funkce zobrazuje diagramy jednotlivě, mezi nimi čeká na klávesu *Enter* nebo kliknutí myší do obrázku. Pro tvorbu všech šesti diagramů, které níže stručně zobrazuji jsem použil volání s parametrem *which* (hodnota od 1 do 6 pro jednotlivé typy).



Obr. 3

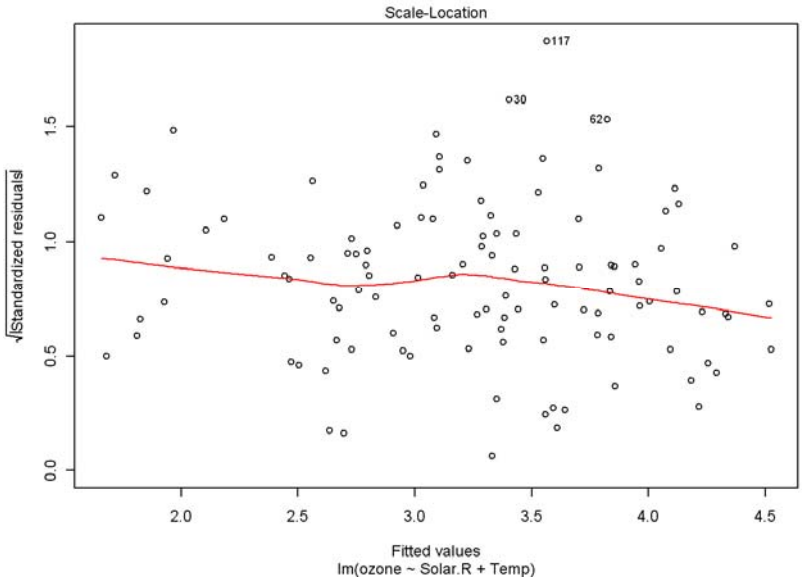
Diagram v Obr. 3 vynáší hodnoty regresních residuálů (tj. odchylek koncentrací ozónu, predikovaných lineárním modelem z hodnot radiace a teploty) proti fitovaným hodnotám.[5] Ve výsledném obláčku bodů by neměla být vidět již žádná tendence ani ve vertikální pozici ani v rozptylu. Případnou tendence v pozici residuálů ve svislém směru by měl pomoci odhalit neparametrický model znázorněný červenou křivkou (model je odhadován metodou loess, se kterou se seznámíme později). V tomto případě nám tato křivka naznačuje, že model lineární závislosti ozónu na radiaci a teplotě je asi příliš jednoduchý, skutečná závislost je více zakřivená než regresní rovina, kterou lineární model představuje.

[5] Ano, opravdu jsou tedy fitované (předpovídané) hodnoty použity jak na ose vodorovné (přímo) tak na ose svislé (při výpočtu hodnot residuálů).



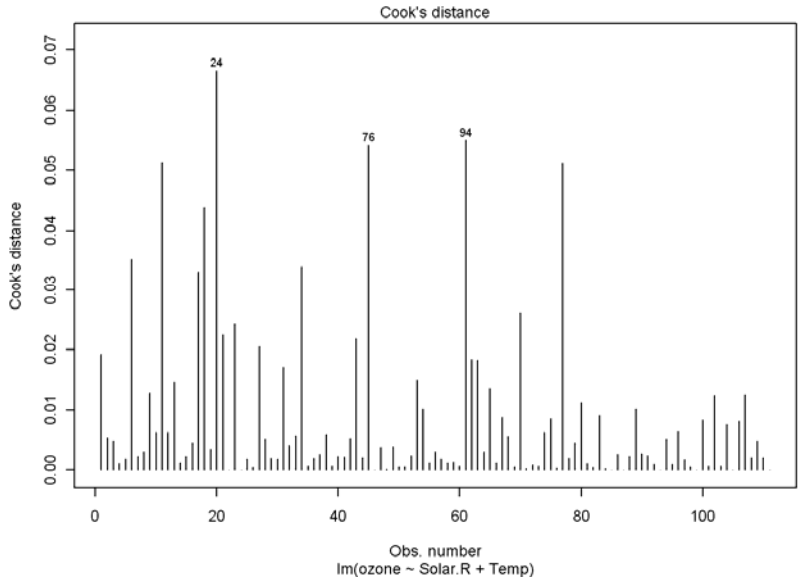
Obr. 4

Z diagramu regresní diagnostiky s hodnotou parametru *which* rovnou 2 (Obr. 4) lze zjistit, jak moc se distribuce residuálů v našem modelu podobá normální (Gaussově) distribuci. V případě plné shody by body (kolečka) ležely na tečkované referenční přímce. To, že se v našem příkladě řetězec bodů zdvihá nad referenční čáru na obou koncích, nám říká, že distribuce je sešikmená, s povlovnějším sestupem v pravém části. Pozorování s takovými velkými kladnými residuály (tj. předpovídaná hodnota koncentrace ozónu je nižší než skutečná) jsou identifikována svým pořadovým číslem (největší pro 117-té pozorování). Všimněme si ještě, že popiska svislé osy mluví o **standardizovaných** residuálech. Je to proto, že residuály odhadnuté odečtením předpovídané od skutečné hodnoty vysvětlované proměnné nemají stejné distribuční vlastnosti (např. proto, že jednotlivá pozorování nemají stejně velký vliv na výsledek fitovní lineárního modelu, viz níže). Nicméně odlišnosti mezi residuály lze odstranit právě tou standardizací (její detaily zde uvádět nebudeme).



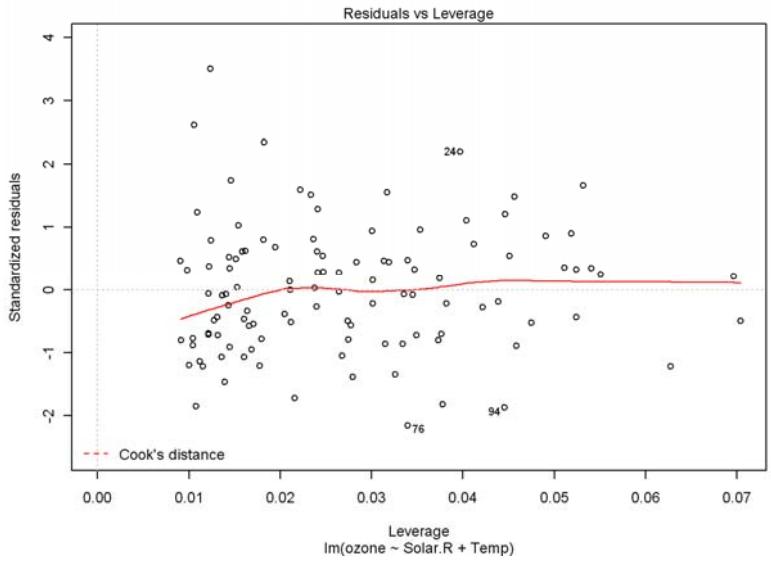
Obr. 5

Diagnostický diagram v Obr. 5 také používá (podobně jako ten v Obr. 3) fitované (předpovídané) hodnoty a residuály, velký rozdíl je ale v tom, že na svislé ose jsou odmocniny absolutních hodnot residuálů (standardizovaných). Diagram proto neukazuje směr odchylky předpovídané od skutečné hodnoty, ale ukazuje lépe její míru, a tím variabilitu residuálů. Jedním z předpokladů klasických lineárních modelů (podobně jako u ANOVA modelů) je tzv. homogenita variancí (homoscedasticity). V případě regrese lze tento požadavek popsat tak, že variabilita residuálů (tj. míra variance vysvětlované proměnné, neobjasněná vysvětlujícími proměnnými) se žádným jednoznačným způsobem nemění s očekávanou hodnotou vysvětlované proměnné (v grafu představovanou fitovanou hodnotou). Nejčastějším typem porušení tohoto požadavku je situace, kdy variabilita roste s očekávanou hodnotou a odpovídající podobou diagramu typu, jaký je v Obr. 5, by byla výrazně rostoucí červená křivka. V našem případě se dá nejvýše říci, že se variabilita postupně a lehce snižuje, za výrazné porušení požadavku homogenity to ale nelze považovat.



Obr. 6

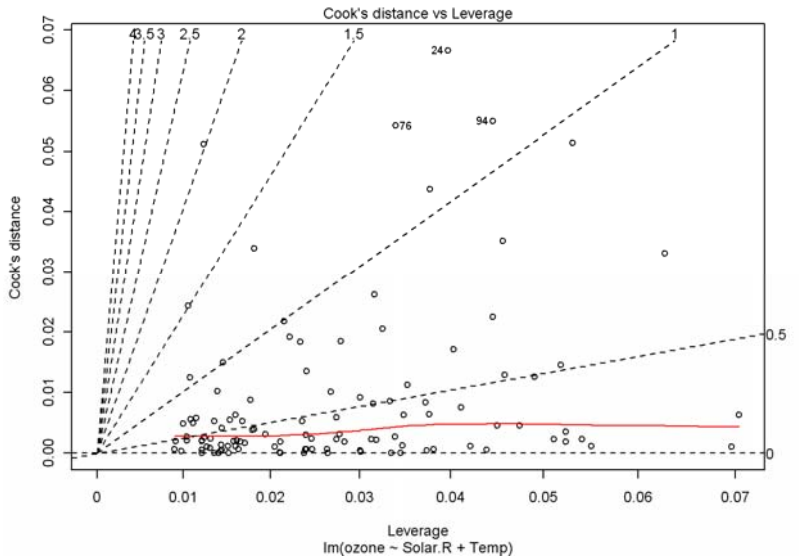
Cookovu distanci lze považovat za druhou základní informaci (po residuálech) o vztahu jednotlivých našich pozorování k regresnímu modelu. Vpodstatě nám tato distance říká, jak moc je výsledná podoba regresního modelu ovlivněna jedním konkrétním pozorováním. Jinými slovy, diagram v Obr. 6, ve kterém je Cookova distance vynášena proti pořadovému číslu pozorování, nám říká, že nejvíce by s odhady regresních koeficientů "zahýbalo" vyřazení pozorování č. 24. V ideálním světě bychom si přáli, aby všechna pozorování měla na výsledným model stejný vliv (ten by pak byl roven p/n, kde p je počet vysvětlujících proměnných v modelu a n je počet pozorování), v praxi to ale nemůžeme očekávat. Přesto je užitečné věnovat velkým hodnotám pozornost a ujistit se, zda nejde o důsledek chyb v zápisu hodnot vysvětlované a/nebo vysvětlujících proměnných. Pokud ne a nelze ani říci, že dané pozorování z jiných apriorních důvodů do vyhodnocování nepatří, nezbývá než se s jeho větším vlivem na výsledek smířit.



Obr. 7

V diagramu na Obr. 7 jsou residuály vynášeny proti statistice zvané *leverage*. Toto anglické slovo lze v této souvislosti přeložit asi nejlépe českým slovem "páka". Leverage nám ukazuje jak velký vliv na výsledný model má pozorování díky hodnotám vysvětlující proměnné (nebo více vysvětlujících proměnných). Pokud by například hodnoty naší (jediné) vysvětlující proměnné byly 9, 14, 10, 13, 125, bude mít největší hodnotu leverage poslední pozorování, bez ohledu na to, zda bude příslušný bod (když se tyto hodnoty zkombinují s hodnotami vysvětlované proměnné) odpovídat závislosti naznačované ostatními pozorováními nebo se od ní odchylovat. Jedním způsobem, jak tyto dva případy oddělit, je právě diagram v Obr. 7. Vidíme, že pozorování s nejvyšší pákou (na pravé straně) nemají regresní residuály tak velké. Ale pozor, to může být také proto, že takové odlehlejší body k sobě nafitovanou regresní přímku účinně "přitáhly".

Jiný způsob, jak se na stejný problém podívat, nabízí diagram vynášející Cookovy distance (zahrnují v sobě jak extrémnost hodnot vysvětlujících proměnných, tak nesoulad hodnot vysvětlované proměnné s modelem) proti hodnotě leverage (Obr. 8).



Obr. 8

Všimněme si, že v Obr. 8 je čárkovanými isočárami vynesen rozsah hodnot standardizovaných regresních residuálů, tj. hodnot, které v předchozím diagramu v Obr. 7 byly přímo vynášeny na svislou osu. Legenda v levém dolním rohu naznačuje, že naopak v něm jsou obdobně vynášeny hodnoty Cookovy distance, nicméně ve vlastním diagramu tyto isočáry nevidíme, protože odpovídají hodnotám, které jsou příliš velké (pracujeme zde totiž s velmi ukázněnými daty, kde žádná opravdu vlivná pozorování nejsou).

ANOVA

Analýzu variance zde zmíníme více okrajově, spíše pro ilustraci jejího vztahu ke klasickému regresnímu modelu. Jak již bylo naznačeno v Úvodu, rozdíl mezi jednoduššími typy ANOVA a regresními modely je jen v tom, že v případě ANOVA jsou vysvětlujícími proměnnými faktory. Krom toho existuje tzv. **analýza kovariance** (ANOCOV či ANCOVA), ve které se oba typy vysvětlujících proměnných kombinují dohromady. Jednoduchý ANOVA model (s jedním faktorem) lze spočítat stejně dobře jako lineární regresní model, ve kterém je faktor nahraze sérií tzv. dummy variables. Každá z těchto proměnných odpovídá jedné hladině faktoru[6] a obsahuje pro dané pozorování hodnotu *1.0* jen tehdy, pokud dané pozorování nabývá danou hladinu faktoru, jinak je hodnota nulová. Z toho vyplývá, že v sérii dummy variables kódujících jeden faktor má každé pozorování vždy jen jednu jedničku, ostatní hodnoty jsou nuly. Pro úplnost je třeba říci, že některé modely analýzy variance nelze tímto způsobem reprodukovat (alespoň ne v rámci jednoho modelu), protože mají více než jednu složku náhodné variability (tzv. nested design, včetně analýzy opakovaných měření – repeated measures ANOVA). Jde ale o omezenost klasického pojetí lineárních modelů, kterou

[6] Není to úplně přesné, pokud v regresním modelu ponecháme absolutní člen (intercept), jedna z dummy variables (odpovídající hladině faktoru, kterou považujeme za referenční, v experimentálních datech typicky "kontrola") musí být vynechána. Odpovídá to logicky i tomu, kolik stupňů volnosti mají jednotlivé faktory v klasické ANOVA (počet hladin faktoru mínus jedna).

odstraňují modely smíšených efektů (linear mixed effect models, LME), popisované v kapitole 8.

Jednoduché i složitější modely ANOVA lze v jazyce S vytvářet pomocí funkce *aov*, která svou rolí odpovídá funkci *lm* u klasických lineárních modelů. Nejjednodušší příklad ANOVA s jedním faktorem (one-way ANOVA) si můžeme ukázat na datech, která jsou distribuována s programem R:

```
> summary(InsectSprays)
     count       spray
 Min.   : 0.00   A:12
 1st Qu.: 3.00   B:12
 Median : 7.00   C:12
 Mean   : 9.50   D:12
 3rd Qu.:14.25   E:12
 Max.   :26.00   F:12
```

Proměnná *count* udává počet jedinců hmyzu, který přežil na rostlinách po použití insekticidu a rozdíly mezi šesti skupinami A až F nám ukazují rozdílnou účinnost jednotlivých druhů insekticidu. Vidíme, že pro každý druh insekticidu máme 12 nezávislých pozorování. To je předpokladem tzv. vyrovnaného uspořádání (balanced design), které zvyšuje sílu našeho testu a v některých složitějších modelech je téměř nezbytné pro jejich efektivní použití. Model ANOVA pro tato data popisuje rozdíly v průměrném počtu jedinců pro jednotlivé typy insekticidů a jeho test odpovídá nulové hypotéze, že se průměry těchto šesti skupin mezi sebou neliší (tj. insekticidy mají stejný účinek):

```
> aov.1<-aov(count~spray,data=InsectSprays)
> summary(aov.1)
            Df  Sum Sq Mean Sq F value    Pr(>F)
spray        5 2668.83  533.77  34.702 < 2.2e-16 ***
Residuals   66 1015.17   15.38
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Všimněme si, že pro objekty vracené funkcí *aov* zobrazuje funkce *summary* výstup obdobný tomu, který jsme pro objekty vracené funkcí *lm* viděli po použití funkce *anova*.[7] Nic nám ve skutečnosti nebrání v tom, abychom na pravé straně vzorečku, který je prvním parametrem funkce *aov*, použili i kvantitativni (nebo jen kvantitativní) proměnné. Rozdíl proti *lm* je pak jen v tom, jak s výsledným objektem nakládají další funkce.

Ačkoliv nám funkce *summary* cudně tají odhadnuté parametry modelu, měli bychom se na ně podívat, abychom pochopili jejich význam. Můžeme si představit (a u jednoduchých ANOVA modelů tomu opravdu tak je), že funkce *aov* provádí svou práci nahrazením faktorů odpovídajícími dummy variables a následným fitováním odpovídajícího lineárního modelu. Parametry ANOVA modelu tedy odpovídají regresním koeficientům stojícím u jednotlivých dummy variables v takovém modelu a jejich hodnoty můžeme zjistit následujícím způsobem:

```
> coef(aov.1)
(Intercept)      sprayB      sprayC      sprayD      sprayE      sprayF
 14.5000000   0.8333333 -12.4166667  -9.5833333 -11.0000000   2.1666667
> summary.lm(aov.1)
```

[7] Pokud bychom použili příkaz *anova(aov.1)*, dostaneme stejný výstup jako s funkcí *summary*.

```
Call:
aov(formula = count ~ spray, data = InsectSprays)

Residuals:
   Min     1Q Median     3Q    Max
-8.333 -1.958 -0.500  1.667  9.333

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  14.5000     1.1322  12.807  < 2e-16 ***
sprayB        0.8333     1.6011   0.520    0.604
sprayC      -12.4167     1.6011  -7.755 7.27e-11 ***
sprayD       -9.5833     1.6011  -5.985 9.82e-08 ***
sprayE      -11.0000     1.6011  -6.870 2.75e-09 ***
sprayF        2.1667     1.6011   1.353    0.181
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.922 on 66 degrees of freedom
Multiple R-Squared: 0.7244,     Adjusted R-squared: 0.7036
F-statistic:  34.7 on 5 and 66 DF,  p-value: < 2.2e-16
```

Funkce *coef* extrahuje ze statistického modelu jeho odhadnuté parametry (regresní koeficienty) a lze ji použít i u klasických lineárních modelů. Zajímavý je ale druhý příkaz, používající funkce *summary.lm*. Viděli jsme již dříve, že funkce *summary* se chová odlišně, pokud jí předáme objekt vrácený funkcí *aov* resp. funkcí *lm*. Jde o tzv. generickou funkci, která "se přizpůsobuje" povaze objektu, se kterým má pracovat. Jazyk S řeší problém generických funkcí tak, že pro každý typ (třídu) objektů má samostatnou verzi generické funkce, jejíž název získáme přidáním tečky a názvu třídy[8]. Objekt vrácený funkcí *lm* patří do třídy s označením *lm* a pokud na něj použijeme funkci *summary*, je automaticky použita verze *summary.lm*. Podobně dříve použitý příkaz *summary(aov.1)* automaticky vybere funkci s názvem *summary.aov*. Pokud ale chceme, aby se program R choval k objektu typu *aov* tak, jako by to byl objekt vrácený z funkce *lm*, musíme název požadované verze použít v příkazu (tedy *summary.lm*).

Aby čtenář nezačal panikařit, vrátím se k našim výsledkům. Co zobrazené koeficienty znamenají? Jde o průměry jednotlivých skupin, ale nejsou jednoduše kódovány. Průměrné počty hmyzu můžeme pro jednotlivé typy insekticidu spočítat přímo a porovnat je s našimi koeficienty:

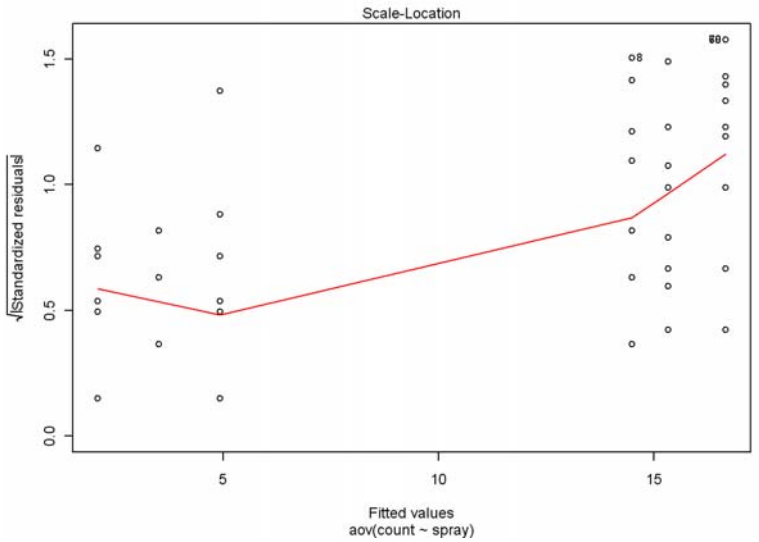
```
> tapply(InsectSprays$count,InsectSprays$spray,mean)
       A        B        C        D        E        F
14.500000 15.333333  2.083333  4.916667  3.500000 16.666667
```

Průměrná hodnota pro insekticid A se shoduje s hodnotou koeficientu pro intercept. Insekticid A byl automaticky zvolen (protože jde o první hladinu faktoru) jako referenční, takže ostatní regresní koeficienty nám ukazují, o kolik se průměr dané skupiny liší od průměru referenčního insekticidu A.

[8] Skutečnost je (jako obvykle) složitější, generické funkce obvykle mají nějakou obecnou verzi, která se používá pro všechny typy objektů, plus speciální verze, které mohou ale nemusí pro jednotlivé třídy existovat.

Podobně jako u klasických regresních modelů, ani tady bychom neměli přijmout nafitovaný model bez prověření, jak dobře jsou splněny jeho předpoklady. Ke grafickému posouzení opět použijeme funkci *plot*. Některé z šesti typů diagramů, které tato funkce nabízí jako pro lineární, tak pro ANOVA modely, postrádají v případě analýzy variance půvab, zde ale uvádím tvorbu jednoho podstatného diagramu[9]:

```
> plot(aov.1,which=3)
```



Obr. 9

Výsledný obrázek (Obr. 9), ve kterém jsou vynášeny odmocniny z absolutních hodnot regresních residuálů proti fitovaným hodnotám, nám ukazuje, že skupiny s většími průměrnými hodnotami (tři svislé pruhy bodů na pravé straně diagramu) mají větší variabilitu, než ostatní tři skupiny. Jde o klasické porušení předpokladu homogenity variancí, jeho závažnost nám pomůže posoudit Bartlettův test:

```
> bartlett.test(count~spray,data=InsectSprays)
        Bartlett test of homogeneity of variances
data:  count by spray
Bartlett's K-squared = 25.9598, df = 5, p-value = 9.085e-05
```

Vzhledem k tomu, že hypotézu o shodě variance mezi našimi šesti skupinami můžeme zamítnout na hladině menší než 0.0001, je jasné, že bude třeba počty jedinců hmyzu transformovat. Zůstaneme zde u klasické logaritmické transformace, i když pro počty jedinců může varianci stabilizovat podobně dobře i odmocninná transformace. Výhodou odmocninné transformace by v našem případě bylo, že by si dobře poradila se dvěma nulovými hodnotami, které v proměnné *count* máme, i bez přičtení jedničky. Z opatrnosti začneme nejprve Bartlettovým testem:

```
> bartlett.test(log(count+1)~spray,data=InsectSprays)
        Bartlett test of homogeneity of variances
data:  log(count + 1) by spray
Bartlett's K-squared = 8.7705, df = 5, p-value = 0.1186
```

A zde je opravený model ANOVA:

[9] Obdobný závěr by ale šel učinit i z QQ diagramu.

```
> aov.1<-aov(log(count+1)~spray,data=InsectSprays)
> summary(aov.1)
            Df Sum Sq Mean Sq F value    Pr(>F)
spray        5 38.518   7.704  46.007 < 2.2e-16 ***
Residuals   66 11.051   0.167
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Interakce mezi faktory a vnoření

Nakonec se zastavíme u trochy složitějšího ANOVA modelu, tzv. dvoucestné analýzy variance (two-way ANOVA). Úplně vhodná data program R nenabízí, zadáme tedy příkladová data ručně (nejsou dlouhá):

```
> y<-c(0.69,0.22,0.19,0.13,0.12,0.15,0.74,0.64,
+ 0.18,0.74,0.21,0.68,0.02,0.49,0.07,0.43)
> x<-c("y","n","y","n","n","n","y","y",
+ "y","y","n","n","n","y","n","y")
> x2<-c("y","y","n","n","y","n","y","n",
+ "n","y","n","y", "n","n","y","y")
> benl<-data.frame(Rhiz=y,B=as.factor(x),P=as.factor(x2))
> summary(benl)
      Rhiz        B     P
 Min.   :0.0200   n:8   n:8
 1st Qu.:0.1450   y:8   y:8
 Median :0.2150
 Mean   :0.3563
 3rd Qu.:0.6500
 Max.   :0.7400
```

Proměnná *Rhiz* představuje údaje o sušině oddenků (většinou patřících travám a ostřicím) získaných odběrem půdních sond konstantního objemu. Každá sonda je odebrána v jiné experimentální ploše terénního experimentu, ve kterém byl na plochy o velikost 1x1 m dlouhodobě aplikován fungicid benomyl (B) a/nebo anorganický fosfát (P), ve faktoriálním uspořádání (4 replikace od každé ze 4 kombinací, prostorově byly čtverce uspořádány jako tzv. Latinský čtverec, toto uspořádání ale v následujících analýzách ignorujeme).

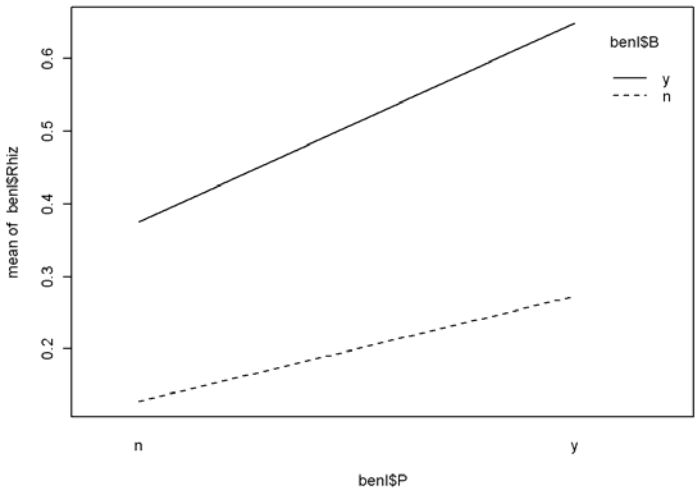
Dvoucestná ANOVA nám umožní zodpovědět otázku, zda má na množství oddenků vliv přidávání fosfátu, zda má vliv fungicid, a případně (pokud zahrneme do modelu interakci mezi oběma faktory), zda se tyto vliv jen sčítají nebo spolu interagují:

```
> aov.2<-aov(Rhiz~B+P,data=benl)
> summary(aov.2)
            Df  Sum Sq Mean Sq F value   Pr(>F)
B            1 0.39062 0.39062 10.3477 0.006746 **
P            1 0.17640 0.17640  4.6728 0.049881 *
Residuals   13 0.49075 0.03775
---
```

Začali jsme modelem jen s hlavními efekty, bez interakce. Výsledek nám ukazuje, že jak benlate, tak fosfáty mají vliv na oddenky, nelze ale zatím říct jaký. Musíme se podívat na odpovídající regresní koeficienty nebo vynést obrázek:

```
> coef(aov.2)
(Intercept)          By          Py
     0.0950      0.3125      0.2100
> interaction.plot(benl$P, benl$B,benl$Rhiz)
```

Obr.10 ukazuje, co funkce *interaction.plot* vytvoří (všimněte si, že prvý a druhý faktor se v diagramu použijí odlišně, je to určeno právě jejich pořadím).



Obr. 10

Jak graf, tak číselné hodnoty nám říkají, že vliv fungicidu i fosfátu na množství oddenků je pozitivní, ale vliv fungicidu je výraznější a máme o něm větší jistotu (ve smyslu dosažené signifikance). Zopakujme si na tomto složitějším příkladě, jak zobrazené koeficienty interpretovat. Průměrná hodnota biomasy oddenků v plochách bez aplikace fungicidu i fosfátů je 0.095. Průměrná hodnota pro plochy s aplikací pouze fungicidu je o 0.3125 vyšší. Předpovídaná průměrná biomasa v plochách, na které byl aplikován fungicid i fosfáty, je tedy 0.095+0.3125+0.2100.

Zatím jsme ale nevzali v úvahu možnou interakci. Přitom ale diagram v Obr. 10 se nazývá interakční (interaction plot) a nestejný sklon úseček také naznačuje, že zde interakce může být. Přesvědčíme se o tom následovně.

```
> aov.2int<-update(aov.2,.~.+B:P)
> anova(aov.2,aov.2int)
Analysis of Variance Table

Model 1: Rhiz ~ B + P
Model 2: Rhiz ~ B + P + B:P
  Res.Df     RSS Df Sum of Sq     F Pr(>F)
1     13 0.49075
2     12 0.47385  1   0.01690 0.428 0.5253
```

Přidání interakce tedy náš model průkazně nezlepší. Ale podívejme se ještě na shrnutí složitějšího modelu:

```
> summary(aov.2int)
            Df  Sum Sq Mean Sq F value  Pr(>F)
B            1 0.39062 0.39062  9.8924 0.00845 **
P            1 0.17640 0.17640  4.4672 0.05617 .
B:P          1 0.01690 0.01690  0.4280 0.52532
Residuals   12 0.47385 0.03949
---
```

Všimněte si, že pokud bychom rovnou začali s modelem zahrnujícím interakce (tak, jak to mnohé statistické programy automaticky nabízejí u analýzy se dvěma faktory), náš

závěr by byl, že průkazný vliv má jen aplikace fungicidu. Posun v signifikanci faktoru P není velký a je především důsledkem poklesu síly testu (připravili jsme se o další stupeň volnosti, což je při celkovém počtu 15 – pro nulový model – dosti závažné).

Interakci jsme do modelu zadávali pomocí operátoru dvojtečka. Nicméně výraz *B+P+B:P* lze zkráceně zapsat jako *B*P*. Pokud bychom ale hvězdičkou propojili tři faktory (např. *A*B*C*), odpovídá takový model nejen třem hlavním efektům, ale také třem interakcím prvního řádu (*A:B*, *A:C* a *B:C*) a také interakci druhého řádu (*A:B:C*).

A úplně nakonec, bez praktického příkladu, zmíním, že máme k dispozici ještě operátor lomítko (*/*), který v případě vzorečku regresního modelu neznamená dělení (podobně jako hvězdička zde nepopisuje násobení), nýbrž situaci, kdy je jeden faktor vnořen ve faktoru druhém. Například zápis *A/B* odpovídá vnoření hodnot faktoru *B* v jednotlivých hladinách faktoru *A*. Jinými slovy, porovnání průměrných hodnot hladin faktoru *B* má smysl jen v kontextu určité hodnoty faktoru *A*, nelze je porovnávat nezávisle na tomto nadřazeném faktoru. Tomu odpovídá i skutečnost, že zápis *A/B* je opět jen "zkratkou" pro *A+A:B* (tj. chybí zde hlavní efekt faktoru *B*).

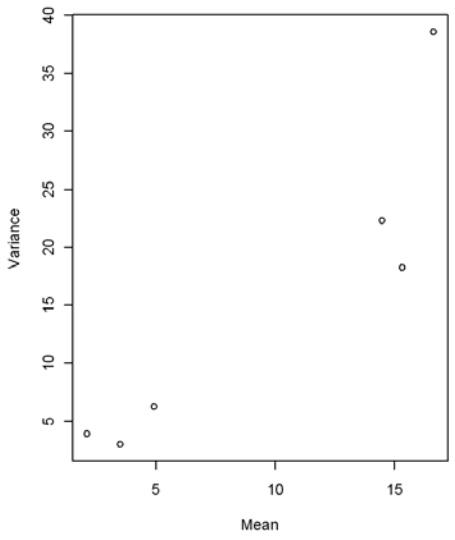
2 Zobecněné lineární modely pro počty a rozměry

Dříve, než si vytvoříme první **zobecněný** lineární model (generalized linear model, GLM), bych rád čtenáři připomenul, že jde o metodu, která staví na tzv. **obecných** lineárních modelech. Ty, mimo klasické lineární regrese, zahrnují i modely, ve kterých jsou vysvětlujícími proměnnými buď *jen* faktory (ANOVA modely), nebo *také* faktory (ANCOVA modely). Pro zdůraznění tohoto rozšíření klasické lineární regrese bude i náš první příklad pracovat s faktoriální vysvětlující proměnnou.

Motivační příklad

Nejprve se vraťme k datům, která jsme používali již dříve, a sice k datovému rámci *InsectSprays*. V tomto příkladu vysvětlujeme nalezené počty hmyzu na rostlinách ošetřených jedním ze šesti druhů insekticidu (označených A až F). Připomínám, že když jsme v předchozí kapitole zkoumali ANOVA model, který jsme pro tato jednoduchá data vytvořili (*anova(count~spray,data=InsectSprays)*), zjistili jsme pomocí obrázku regresní diagnostiky (a potvrdili si Bartlettovým testem), že variabilita reziduálů roste s průměrnou hodnotou počtů ve skupinách, odpovídajících jednotlivým druhům insekticidu. Tento vztah si můžeme graficky ukázat následujícím postupem:

```
> aov.is<-aov(count~spray, data=InsectSprays)
> y<-tapply(resid(aov.is),InsectSprays$spray,var)
> x<-tapply(fitted(aov.is),InsectSprays$spray, mean)
> plot(y~x,xlab="Mean",ylab="Variance")
```



Obr. 11

Funkce *tapply* (kterou jsme potkali již na konci předcházející kapitoly) nám umožňuje použít zvolenou funkci (třetí parametr funkce *tapply*) na hodnoty proměnné zadané jako první parametr a roztříděné do skupin pomocí faktoru, který je druhým parametrem *tapply*. V tomto případě jsme do proměnné *y* uložili varianci residuálů pro jednotlivé

insekticidy, zatímco v *x* jsou průměry hodnot fitovaných pro jednotlivé skupiny[10]. Vztah vynesený v Obr. 11 ukazuje lineární růst variance se střední hodnotou. Jak je z Obr. 11 vidět, hodnota variance roste úměrně hodnotě průměru, a to nás spolu se skutečností, že hodnoty proměnné *count* představují počty jedinců, vede k závěru, že residuální variabilitu by lépe než Gaussova (normální) distribuce popisovala tzv. Poissonova distribuce. Ta je jednou z více distribucí, které můžeme zvolit pro popis náhodné (nevysvětlené) variability ve zobecněném lineárním modelu.

Fitování zobecněného lineárního modelu

Abychom si takový model ukázali na našich datech, postačí volání funkce *lm* nahradit funkcí *glm* a v té přidat jeden nový parametr, s názvem *family*, který určuje typ distribuce.

```
> glm.1<-glm(count~spray,data=InsectSprays,family=poisson)
```

Protože jsme problém nestálé variance v proměnné *count* řešili v předchozí kapitole logaritmickou transformací (po přičtení jedničky), mohlo by být zajímavé výsledky obou modelů porovnat[11], srovnáním jak regresních koeficientů, tak předpovídaných hodnot:

```
> coef(glm.1)
(Intercept)      sprayB      sprayC      sprayD      sprayE
 2.67414865  0.05588046 -1.94017947 -1.08151786 -1.42138568
     sprayF
 0.13926207
> coef(aov.1)
(Intercept)      sprayB      sprayC      sprayD      sprayE
 2.69673832  0.05980403 -1.74413153 -0.98342718 -1.27049988
     sprayF
 0.11892303
>
> summary(InsectSprays$count)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.00    3.00    7.00    9.50   14.25   26.00
> summary(fitted(glm.1))
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  2.083   3.500   9.708   9.500  15.330  16.670
> summary(fitted(aov.1))
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.9526  1.4260  2.2050  2.0600  2.7570  2.8160
```

Pokud se čtenář nad porovnáním regresních koeficentů a následným porovnáním skutečných a oběma modely fitovaných hodnot zamyslí, asi mu přijde něco divného. Zatímco počty předpovídané oběma modely dobře odpovídají tomu, že v modelu *aov.1* byly hodnoty logaritmovány, zatímco v modelu *glm.1* ne, hodnoty regresních koeficientů jsou si překvapivě podobné, i když ne shodné. Jak je to možné?

Odpověď je docela jednoduchá. V použitém zobecněném lineárním modelu se také skrývá logaritmická transformace, díky které funkce *glm* pracuje s počty jedinců hmyzu na podobné (transformované) škále jako funkce *aov*.

---

[10] Mimochodem, v tomto případě se průměrovaly v každé skupině zcela stejné hodnoty (fitovanou hodnotou je zde právě průměr), takže použití funkce jen zajistilo, že máme pro každou skupinu pouze jeden údaj, nikoliv dvanáct stejných.

[11] Model *aov.1* byl získán příkazem *aov.1<-aov(log(count+1)~spray,data=InsectSprays)*

Součásti GLM

Zobecněné lineární modely používají tzv. **link funkci**. Jde o jednoduchou funkci[12], která převádí hodnoty vysvětlované proměnné na škálu slučitelnou s hodnotami, které nám definuje pravá strana regresní rovnice, obsahující vysvětlující proměnné. Například počty jedinců, které nikdy nemohou být záporné, se po transformaci vhodnou link funkcí (zde logaritmus) "roztáhnou" přes všechna možná kladná a záporná čísla, která mohou hodnoty vysvětujících proměnných násobené regresními koeficienty a sečtené dohromady vytvořit. Čtenář teď možná o předchozí větě pochybuje: nulu logaritmovat nelze, nejmenší nenulový počet je 1 a po zlogaritmování se změní na nulu, zatímco všechny ostatní větší počty se změní na čísla větší než nula. Připomeňme si ale, že regresní model nepředpovídá hodnotu konkrétních pozorování, nýbrž *průměrnou hodnotu* vysvětlované proměnné, kterou očekáváme pro danou kombinaci hodnot proměnných vysvětlujících. V případě počtu jedinců a jeho popisu Poissonovou distribucí je tato střední hodnota Poissonovy distribuce vždy kladná, ale může být například 0.35. Pokud by Poissonova distribuce měla nulový průměr, byla by také pravděpodobnost, že se při libovolně velkém počtu opakování setkáme alespoň s jedním kusem hmyzu, nulová, což není rozumný předpoklad.

Nyní si uvedený popis GLM shrneme ve více matematické podobě. U klasického lineárního modelu s více vysvětlujícími proměnnými jsme v předchozí kapitole použili termín "lineární kombinace vysvětlujících proměnných", který popisuje způsob, jakým se v modelu kombinují regresní koeficienty s hodnotami proměnných. U zobecněných lineárních modelů budeme tuto kombinaci nazývat **lineární prediktor** a můžeme ji označit řeckým písmenem *eta*:

$$\eta_i = b_0 + \sum_{j=1}^{p} b_j * x_{ji}$$

Odtud je již jen kousek k obecnému zapsání GLM. Podobně jako u klasického lineárního modelu vyjádříme hodnotu vysvětlované proměnné pomocí dvou složek: očekávané (čili – pro reálná data – fitované) hodnoty a náhodné složky, představující regresní residuál:

$$y_i = \hat{y}_i + e_i, \text{ kde } g(\hat{y}_i) = \eta_i$$

Funkce g je link funkce a residuály e_i mohou pocházet z **různých distribucí**. V našem příkladě jsme při užití funkce *glm* předpokládali, že vhodnou distribucí je Poissonova, ale v rámci GLM máme k dispozici i další distribuce, které probereme dále. Kde jsme ale ve funkci *glm* zvolili, že link funkcí bude logaritmická funkce? Ke každému ze základních typů distribucí, které GLM podporují, patří jedna tzv. **kanonická link funkce**, a v případě Poissonovy distribuce je to právě logaritmická link funkce, která se proto implicitně zvolí.

Na tomto místě již můžeme porovnat zobecněný lineární model (GLM) s normálním lineárním modelem. U obou typů řešíme stejné problémy při rozhodování, které vysvětlující proměnné v modelu použít, u GLM ale máme větší možnosti ve volbě

[12] která navíc musí mít komplementární inverzní funkci, která provádí transformaci opačným směrem, u logaritmu je to např. exponenciální funkce

distribuce nevysvětlené variability a také ve volbě link funkce, která "překládá" rozsah hodnot lineárního prediktoru (prakticky libovolné reálné číslo) do více omezeného rozsahu hodnot vysvětlované proměnné (kladná čísla u počtů ale také u rozměrů či většiny poměrů, hodnoty mezi 0 a 1 pro pravděpodobnosti, atd.). Pro každý běžný typ distribuce, užívaný u GLM, existuje kanonická link funkce, ale pro některé distribuce jsou k dispozici i jiné link funkce, které v konkrétních případech zmíníme.

Na konec tohoto úvodu ke GLM vysvětlím, proč funkce *fitted* vracela v případě *glm.1* modelu hodnoty na původní škále počtu jedinců. Je to proto, že tato funkce v případě GLM (ale také v případě zobecněných *aditivních* modelů, které budeme probírat později) spočítá fitované hodnoty dosazením hodnot vysvětlujících proměnných do lineárního prediktoru s odhadnutými koeficienty, ale pak ještě výsledek transformuje funkcí inverzní k link funkci. Proto například předpovídaná střední hodnota počtu jedinců hmyzu po použití insekticidu C je rovna (s drobnou zaokrouhlovací chybou):

```
> exp(2.67415865-1.94017947)
[1] 2.083354
```

tak, jak to ukazuje i výsledek funkce *fitted*.

```
> fitted(glm.1)[25]
      25
2.083333
```

Základní typy distribucí v GLM

Následující tabulka podává přehled hlavních typů distribucí, názvů užívaných link funkcí a typů a vlastností vysvětlovaných proměnných, pro které se jednotlivé možnosti hodí. Pořadí odpovídá předpokládané četnosti jejich využití v biologických vědách.

Statistická distribuce		**Kanonická link funkce**		**Vysvětlovaná proměnná**	
Název	**v S**	**Název**	**Definice**	**Hodnoty**	**Typy údajů**
binomická	binomial	logit	$\eta = \ln(\frac{\hat{y}}{1-\hat{y}})$	podíly z počtů	pravděpodobnosti jevů či výsledků
Poissonova	poisson	log	$\eta = \ln(\hat{y})$	celé nezáporné	počty případů, též konting. tabulky
negativně binomická	negative. binomial	log	$\eta = \ln(\hat{y})$	celé nezáporné	počty jedinců při shlukovité distrib.
Gamma	Gamma	reciprocal	$\eta = \frac{1}{\hat{y}}$	kladné reálné	velikosti, hmotnosti, jejich podíly
Gaussova	gaussian	identity	$\eta = \hat{y}$	jakékoliv reálné	s ohledem na ostatní možnosti skoro žádné

K tabulkovému přehledu je třeba doplnit, že možnost používat negativně binomickou distribuci není základní schopností programu R, přidává ji balíček (package) s názvem MASS. Naopak v přehledu chybí inverzní Gaussovská distribuce a také volby označované jako *quasi*, *quasipoisson* a *quasibinomial*. Alespoň poslední dvě ale zmíním v dalším textu.

Koeficienty GLM

Vraťme se k nafitovanému GLM, který je uložen v objektu *glm.1*, a podívejme se na jeho shrnutí (prvých osm řádků vypsaných funkcí *summary* bylo smazáno):

```
> summary(glm.1)
...
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  2.67415    0.07581  35.274  < 2e-16 ***
sprayB       0.05588    0.10574   0.528    0.597
sprayC      -1.94018    0.21389  -9.071  < 2e-16 ***
sprayD      -1.08152    0.15065  -7.179 7.03e-13 ***
sprayE      -1.42139    0.17192  -8.268  < 2e-16 ***
sprayF       0.13926    0.10367   1.343    0.179
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 409.041  on 71  degrees of freedom
Residual deviance:  98.329  on 66  degrees of freedom
AIC: 376.59

Number of Fisher Scoring iterations: 5
```

Tabulka s přehledem odhadnutých koeficientů nás svým obsahem nepřekvapí. Opět zde máme absolutní člen, představující hodnotu lineárního prediktoru pro insekticid A, zatímco ostatní koeficienty představují změnu hodnoty tohoto prediktoru při záměně insekticidu A jiným. Protože model *glm.1* byl nafitován s implicitní volbou logaritmické link funkce, můžeme údaje, které nám regresní koeficienty představují, převést na škálu, ve které jsme vysvětlovanou proměnnou měřili (tj. počty jedinců) pomocí exponenciální funkce. Podívejte se na následující výpočty:

```
> exp(2.67415)
[1] 14.50002
> exp(2.67415-1.94018)
[1] 2.083335
> exp(-1.94018)
[1] 0.1436781
> 2.083335/14.5
[1] 0.1436783
```

Výsledek prvního příkazu nám říká, že průměrný počet jedinců hmyzu na rostlinách ošetřených insekticidem A je 14.5 (stejný odhad jako z funkce *aov* na konci prvé kapitoly). Podobně druhý výsledek (tj. exponenciela ze součtu absolutního členu a koeficientu odpovídajícího insekticidu C) nám říká, že očekávaný průměrný počet

hmyzů na rostlinách ošetřených nejúčinnějším insekticidem C bude 2.08. Třetí příkaz je asi nejdůležitější, protože nám ukazuje – v kombinaci s příkazem následujícím – jak interpretovat jednotlivé koeficienty z modelů používajících logaritmickou link funkci (log se často používá i u negativně binomické a u Gamma distribuce, u té druhé hlavně pro hmotnosti a rozměry). Výsledek 0.1436781 (tj. exponenciela z regresního koeficientu) nám říká, že průměrný počet jedinců hmyzu bude na rostlinách ošetřených insekticidem C o 86% (tj. 100*(1-0.14) po zaokrouhlení) menší v porovnání s "referenčním" přípravkem A. Porovnávané vlivy jsou tedy na škále vysvětlované proměnné násobné, nikoliv "posuvné" (statistik by řekl "nikoliv aditivní"). Jinými slovy, hodnota pro C je 0.14-krát "větší" než pro referenční A. Tuto interpretaci hodnoty koeficientu bychom si mohli ukázat i více matematicky, úpravou rovnice předpovídající počet hmyzů pro insekticid C:

$$\hat{y}_C = e^{2.67415-1.94018} = e^{2.67415} \cdot e^{-1.94018}$$

Podobnou interpretaci můžeme použít i pro regresní koeficienty patřící ke kvantitativním vysvětlujícím proměnným. V takových případech ovšem neuvažujeme o změně kategorie u kategoriální proměnné, nýbrž o posunu (změně) hodnoty vysvětlující proměnné o jednotku. Pokud bychom tedy např. pro vysvětlování počtu hmyzů používali také dobu od aplikace insekticidu (měřenou v hodinách) a koeficient odpovídající době od aplikace měl hodnotu 1.098, pak bychom mohli říci, že očekávaný počet se s každou hodinou uplynulou od aplikace v průměru ztrojnásobí (protože exp(1.098) = 2.998).

Tabulka regresních koeficientů, kterou funkce *summary* zobrazuje pro GLM, se ale přeci jen trochu liší od tabulky pro klasický lineární model. Místo *t* statistik vidíme tzv. *z* statistiky, které se nabízejí pro srovnání s normovanou normální distribucí (N(0,1)), pro test dílčí hypotézy o hodnotě určitého koeficientu. Jde o tzv. **Waldovy statistiky**. Pro čtenáře mám ohledně jejich použití jedinou radu: nepoužívejte je. Předpoklady, které jsou s jejich oprávněným použitím spojeny, nejsou obvykle splnitelné a závěry z nich udělané jsou proto často nesprávné.

Analýza deviance

Na tomto místě zopakuji závěrečnou část výstupu funkce *summary*, abych čtenáře nemusel odkazovat příliš dozadu (v rámci kapitoly, myslím ;-)

```
(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 409.041  on 71  degrees of freedom
Residual deviance:  98.329  on 66  degrees of freedom
AIC: 376.59

Number of Fisher Scoring iterations: 5
```

Funkce nám nejprve sděluje, že o hodnotě dispersního parametru se předpokládá, že je rovna jedné. Vysvětlování, co tento parametr znamená obecně v GLM, by neprospělo čtenářovu přátelskému vztahu ke statistice, v případě Poissonovy distribuce to ale není zas tak složité. Pro ni totiž platí, že variance je rovna průměru. To můžeme zapsat například takovouto rovnicí: V(y) = Ey. V mnoha typech biologických dat představujících počty nebo i podíly (pravděpodobnosti) ale není variance *rovna* průměru, s průměrem ale lineárně roste (rychleji či pomaleji než průměr). To lze zapsat rozšířením

rovnice o parametr popisující rychlost nárůstu variance: V(y) = ϕ*Ey. A právě tento koeficient ϕ odpovídá dispersnímu parametru. Pokud používáme GLM a předpokládáme Poissonovu distribuci pro nevysvětlenou variabilitu, funkce *summary* a *anova* odhadují signifikance pro testy parametrů s předpokladem, že je ϕ rovno jedné (jak nám hláška ve výstupu sděluje). Pokud je ale ve skutečnosti ϕ významně větší než jedna (situace označovaná jako **overdispersion**), správné pravděpodobnosti chyb I. druhu (tj. hodnoty *p*) jsou vyšší, tj. naše závěry o hypotézách budou příliš optimistické. K možnému způsobu řešení se dostaneme v následující kapitole, v části pojednávající o nadměrné variabilitě.

V dalších dvou řádkách výstupu z funkce *summary* najdeme pojem **deviance**. Ve smyslu, v jakém je zde používána, představuje deviance zobecnění residuální sumy čtverců, které známe z regresního modelu. Residuální suma čtverců nafitovaného modelu je uvedena pod názvem *Residual deviance*. Deviance v předcházející řádce také představuje residuální devianci, ale pro tzv. **nulový model**. V nulovém modelu je v lineárním prediktoru přítomen jen absolutní člen, žádná proměnná vysvětlující. Můžeme proto říct, že deviance nulového modelu odpovídá celkové sumě čtverců (TSS) v klasickém lineárním či v ANOVA modelu. Podobně jako u těchto modelů můžeme celkovou devianci (devianci nulového modelu) rozložit na část naším modelem vysvětlenou a část nevysvětlenou. Pro náš příklad platí, že je naším modelem (tj. typem insekticidu) vysvětlena deviance o velikosti 409.041-98.329, a to za použití 5 (71-66) stupňů volnosti. Tento rozklad nám názorněji ukáže tabulka **analýzy deviance**, kterou počítá (to je trochu matoucí) funkce *anova*, pokud jí použijeme na GLM:

```
> anova(glm.1)
Analysis of Deviance Table
Model: poisson, link: log
Response: count
Terms added sequentially (first to last)

      Df Deviance Resid. Df Resid. Dev
NULL                     71     409.04
spray  5   310.71        66      98.33
```

Překvapující je, že na rozdíl od klasického rozkladu variance u regresních či ANOVA modelů nám zde funkce *anova* nenabízí žádný test signifikance. Je to proto, že v úvahu přicházejí u zobecněných lineárních modelů dva možné druhy testů a volba alespoň zčásti závisí na typu GLM a vlastnostech vysvětlované proměnné (např. přítomnost nadměrné variability). Pokud bychom předpokládali, že použití standardní Poissonovy distribuce je v našem případě správná volba, měl by test být založen na srovnání modelem snížené deviance (310.71) s χ^2 distribucí s 5 stupni volnosti. Použití tohoto testu uložíme funkci *anova* zadáním hodnoty parametru *test*:

```
> anova(glm.1,test="Chisq")
Analysis of Deviance Table
Model: poisson, link: log
Response: count
Terms added sequentially (first to last)
      Df Deviance Resid. Df Resid. Dev P(>|Chi|)
NULL                     71     409.04
spray  5   310.71        66      98.33 4.979e-65
```

Pokud by pro jiný typ analýzy deviance GLM byl vhodnější F test (obdobný F testu při rozkladu variance klasických modelů), parametr *test* by měl hodnotu *"F"*.

Residuály pro GLM

Pro obecný lineární model (tj. ANOVA a klasickou lineární regresi) jsme si ukázali, že residuální suma čtverců (RSS) je rovna součtu druhých mocnin regresních residuálů a současně jsou tyto residuály definovány jako rozdíly mezi skutečnou a modelem předpovídanou hodnotou vysvětlované proměnné. U GLM není situace tak jednoduchá, residuály definované podle prvé nebo druhé definice se obvykle liší - s výjimkou GLM se zvolenou Gaussovou distribucí a link funkcí "identity" (která ponechává hodnoty beze změny). Residuály, které po sečtení jejich druhých mocnin představují hodnotu residuální deviance, se nazývají **devianční residuály** (deviance residuals). Ty jsou implicitním typem, který funkce *resid* vrací. Residuály, které vzniknou odečtením fitované hodnoty vysvětlované proměnné (po transformací inverzní link funkcí) od hodnoty pozorované, lze spočítat použitím funkce *resid* s parametrem *type="response"*. Nicméně, tento typ není příliš často používán, vzhledem k tomu, že se pro některé distribuce variabilita těchto residuálů mění podle očekávané (fitované) hodnoty. Místo toho se doporučují tzv. **Pearsonovy residuály** (vypočteme použitím parametru *type="pearson"*), které jsou standardizovány tak, že případná změna variability ("rozptylu") hodnot těchto residuálů je nad rámec očekáváné proměnlivosti a naznačuje možné problémy.

Loglineární modely: analýza kontingenčních tabulek

Loglineární modely se užívaly již nějakou dobu před popularizací zobecněných lineárních modelů, ale představují jen jejich zvláštní případ. Slouží ke hledání vztahu mezi faktory a k testování hypotéz, týkajících se jednotlivých stran ("rozměrů") kontingenčních tabulek[13]. I v těchto modelech obvykle hraje jeden z faktorů roli "vysvětlované proměnné", ale v tomto speciálním případě nestojí v rovnici regresního modelu na levé straně – tam stojí hodnoty počtu případů z jednotlivých buněk kontingenční tabulky. Proto se také pro "závislý" faktor používá spíše termín **faktor odezvy** (response factor) a pro vysvětlující faktory termín stimulus factors (anglicky, nemám odvahu to překládat, smysl je celkem zjevný i tak).

Postup analýzy složitějších kontingenčních tabulek si ukážeme na příkladu, který sice není z oboru přírodních věd, ale bude snad pro čtenáře atraktivní. Příkladová data *Titanic* obsahují údaje o osudech 2201 pasažérů této známé lodi. Data byla sebrána vyšetřující komisí, a tak nás bude – podobně jako tuto komisi - nejvíce zajímat, zda přežití či nepřežití (faktor *Survived*) bylo ovlivněno postavením pasažéra: věkem (*Age* – dítě či dospělý), pohlavím (*Sex* – muž či žena) a třídou, ve které cestoval, tedy vpodstatě ekonomickým stavem (*Class* – první, druhá, třetí a posádka). Objekt *Titanic* není ale

[13] Každá strana dvou- i vícerozměrné kontingenční tabulky odpovídá jednomu z faktorů a tabulka je podél takové strany rozdělena na tolik částí ("řádků" / "sloupců"), kolik hladin faktor má. To mimochodem znamená, že v takovýchto analýzách můžeme pracovat i s vysvětlovanou proměnnou, která má větší počet samostatných kategorií, např. když pokusná zvířata volí z více možností (např. typ stravy) a my chceme zjistit, které další faktory jejich rozhodnutí ovlivňují.

datovým rámcem, tedy typem dat, která jsme dosud používali, nýbrž skutečnou čtyř-rozměrnou tabulkou:

```
> dim(Titanic)
[1] 4 2 2 2
```

Vlastnosti tabulky dobře shrnuje (a také provádí základní klasický χ^2 test hypotézy o úplné nezávislosti mezi všemi čtyřmi faktory) funkce *summary*:

```
> summary(Titanic)
Number of cases in table: 2201
Number of factors: 4
Test for independence of all factors:
        Chisq = 1637.4, df = 25, p-value = 0
        Chi-squared approximation may be incorrect
```

Tuto tabulku nelze úplně jednoduše zobrazit (na rozdíl od tabulky dvourozměrné), jazyk S to řeší jejím "rozřezáváním" podél všech faktorů mimo prvního a druhého. Výstup je docela dlouhý, proto z něj uvádím jen část:

```
> Titanic
, , Age = Child, Survived = No

      Sex
Class  Male Female
  1st     0      0
  2nd     0      0
  3rd    35     17
  Crew    0      0

, , Age = Adult, Survived = No

      Sex
Class  Male Female
  1st   118      4
  2nd   154     13
  3rd   387     89
  Crew  670      3
```

...

V této podobě ale nemůžeme data použít pro nafitování zobecněného lineárního modelu, počty osob z jednotlivých buněk tabulky potřebujeme "natáhnout" do jednoho sloupečku (proměnné), přičemž další proměnné budou odpovídat jednotlivým rozměrům (stranám) původní čtyřrozměrné tabulky. Způsob změny je překvapivě jednoduchý:

```
> tit<-as.data.frame(Titanic)
> summary(tit)
  Class       Sex        Age     Survived      Freq
 1st :8   Male  :16   Child:16   No :16   Min.   :  0.00
 2nd :8   Female:16   Adult:16   Yes:16   1st Qu.:  0.75
 3rd :8                                  Median : 13.50
 Crew:8                                  Mean   : 68.78
                                         3rd Qu.: 77.00
                                         Max.   :670.00
```

Čtenáře upozorňuji, že všechny případy, které mají stejnou hodnotu pro všechny čtyři faktory (*Class*, *Sex*, *Age*, *Survived*) jsou sdruženy do jednoho řádku, celkový počet případů v našem datovém rámci je proto 4*2*2*2 = 32, nikoliv 2201. Počty jsou uloženy v proměnné *Freq*.

Nyní můžeme začít pracovat s log-lineárním modelem. Připomínám, že log-lineární modely jsou tak trochu exotikou ve srovnání s jinými, u kterých není vysvětlovaná proměnná faktorem. Tomu odpovídá už to, jak vypadá nulový model:

```
> glm.tit.0<-glm(Freq~Class*Sex*Age+Survived,data=tit,family=poisson)
```

Definici tohoto modelu doporučuji důkladně prostudovat. Nulový model, tj. model odpovídající hypotéze, že pravděpodobnost přežití (tj. obecně pravděpodobnosti pro jednotlivé kategorie odezvového faktoru, zde *Survived*) nezávisí na žádném z ostatních faktorů, obsahuje hlavní efekt odezvového faktoru, ale také hlavní efekty a všechny možné interakce mezi vysvětlujícími faktory. Žádná z těchto částí se nesmí při dalších úpravách modelu odstranit. Naše alternativní hypotézy jsou tedy představovány jen interakcemi, obsahujícími odezvový faktor. Je vhodné začít u interakcí prvního řádu. Ty v případě našeho příkladu budou testovat hypotézy, ze přežití záviselo na třídě, ve které osoba cestovala, na jejím pohlaví, nebo na jejím věku (dítě vs. dospělý). Pro vysvětlující faktory, u kterých se takové interakce prvního řádu ukázaly průkazné, můžeme testovat i složitější hypotézy – např. závisela (případná) vyšší pravděpodobnost přežití dětí na tom, ve které třídě cestovaly?

Začneme tedy nejprve interakcemi prvního řádu: funkce *add1* nám umožní ozkoušet důsledky rozšíření modelu vždy o jeden člen, v tomto případě interakce prvého řádu mezi proměnnou *Survived* a jedním ze tří vysvětlujících faktorů. Všimněte si, jakým způsobem možná rozšíření modelu zadáváme: tečky ve vzorci znamenají "vše, co ve stávajícím modelu již je":

```
> add1(glm.tit.0,.~.+Survived:(Class+Sex+Age),test="Chisq")
Single term additions

Model:
Freq ~ Class * Sex * Age + Survived
              Df Deviance    AIC    LRT   Pr(Chi)
<none>             671.96 833.36
Class:Survived  3   491.06 658.46 180.90 < 2.2e-16 ***
Sex:Survived    1   237.49 400.90 434.47 < 2.2e-16 ***
Age:Survived    1   652.40 815.80  19.56 9.746e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Řádka začínající <*none*> představuje stávající, tj. náš nulový model. Následující tři řádky pak představují **jednotlivě** přidané členy a důsledky jejich přidání do nulového modelu. Je vidět (podle hodnoty poklesu deviance, ale i podle nejnižší hodnoty AIC), že kvalitu model nejvíce zvýší interakce, odpovídající přednostní záchraně žen. Protože uvedené testy se týkají vždy jednotlivých členů (jde o jejich tzv. "marginální", nezávislé efekty), neměli bychom si dovolit přidat všechny tři najednou, je třeba je testovat postupně:

```
> glm.tit.1<-update(glm.tit.0,.~.+Sex:Survived)

> add1(glm.tit.1,.~.+Survived:(Class+Age),test="Chisq")

Single term additions

Model:
Freq ~ Class + Sex + Age + Survived + Class:Sex + Class:Age +
    Sex:Age + Sex:Survived + Class:Sex:Age
```

```
               Df Deviance    AIC    LRT Pr(Chi)
<none>              237.49 400.90
Class:Survived  3   131.42 300.82 106.08 < 2e-16 ***
Age:Survived    1   231.60 397.00   5.89 0.01520 *
```

Je vidět, že i po zohlednění odlišného přístupu k ženám zůstává rozdíl mezi cestovními třídami výraznější než rozdíl mezi dospělými a dětmi. Je zde ale třeba poznamenat, že tento efekt může být způsoben čistě výrazně menší šancí posádky na záchranu, nemusí implikovat rozdílný přístup k nejnižší, třetí třídě cestujících. Po přidání tohoto členu můžeme velikost rozdílů samozřejmě ověřit:

```
> glm.tit.2<-update(glm.tit.1,.~.+Class:Survived)
> coef(glm.tit.2)
                  (Intercept)                      Class2nd
                    1.0773443                     1.0807642
                     Class3rd                     ClassCrew
                    2.6682755                   -17.7633987
                     SexFemale                      AgeAdult
                   -3.2645902                     3.5553481
                  SurvivedYes            Class2nd:SexFemale
                   -0.3531200                     2.2723605
          Class3rd:SexFemale          ClassCrew:SexFemale
                    2.0339597                     1.9973715
           Class2nd:AgeAdult            Class3rd:AgeAdult
                   -0.8292794                    -1.2909842
          ClassCrew:AgeAdult           SexFemale:AgeAdult
                   19.6344319                     1.4144652
       SexFemale:SurvivedYes         Class2nd:SurvivedYes
                    2.4213285                    -0.9525972
        Class3rd:SurvivedYes        ClassCrew:SurvivedYes
                   -1.6582356                    -0.8808128
 Class2nd:SexFemale:AgeAdult  Class3rd:SexFemale:AgeAdult
                   -2.1728838                    -2.0068708
ClassCrew:SexFemale:AgeAdult
                   -4.9690878
```

Ve výstupu jsou tučně znázorněny jediné z koeficientů, které má smysl interpretovat. Ostatní jen odpovídají rozdílům v relativní četnosti případů pro jednotlivé kategorie faktorů, nebo pro jejich kombinace (proto jsou také nezbytnou součástí nulového loglineárního modelu). Vidíme, že ženy měly výrazně vyšší šanci přežít než muži, nejvyšší šanci přežít měly osoby z 1. třídy (pro tu koeficient v modelu není, nicméně další tři implikují rozdíly právě oproti této referenční třídě), a že šance na přežití byla nižší pro 2. třídu a ještě více pro 3. třídu, ve srovnání s posádkou. Vrátíme se ještě k rozdílu mezi dětmi a dospělými, který skóroval až na třetím místě:

```
> add1(glm.tit.2,.~.+Survived:Age,test="Chisq")
Single term additions

Model:
Freq ~ Class + Sex + Age + Survived + Class:Sex + Class:Age +
    Sex:Age + Sex:Survived + Class:Survived + Class:Sex:Age
             Df Deviance     AIC     LRT   Pr(Chi)
<none>           131.418 300.820
Age:Survived  1  112.567 283.969  18.852 1.413e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
```

I tento termín je tedy nakonec průkazný, preference dětí při záchraně pasažérů vynikla až po zahrnutí předchozích dvou interakčních členů.

```
> glm.tit.3<-update(glm.tit.2,.~.+Survived:Age)
```

Užitečnost interakcí druhého řádu posoudíme takto:

```
> add1(glm.tit.3,.~.+Survived:(Sex*Age*Class),test="Chisq")
Single term additions

Model:
Freq ~ Class + Sex + Age + Survived + Class:Sex + Class:Age +
    Sex:Age + Sex:Survived + Class:Survived + Age:Survived +
    Class:Sex:Age
                  Df Deviance     AIC     LRT   Pr(Chi)
<none>               112.567 283.969
Sex:Age:Survived    1   94.548 267.950  18.018 2.188e-05 ***
Class:Sex:Survived  3   45.899 223.301  66.667 2.206e-14 ***
Class:Age:Survived  3   76.904 254.306  35.663 8.825e-08 ***
```

Opět je vidět, že pokud uvažujeme o těchto interakcích jednotlivě, přidání každé z nich by bylo vhodné (posuzováno jak testem ve sloupci *Pr(Chi)*, tak hodnotou AIC statistiky). Ale opět musíme vzít v úvahu jejich kombinované efekty a vybírat je postupně. Určitě začneme členem *Class:Sex:Survived*, který odpovídá hypotéze, že vliv pohlaví pasažéra na jeho přežití závisel zčásti na tom, v jaké třídě cestoval.

```
> glm.tit.4<-update(glm.tit.3,.~.+Class:Sex:Survived)
> add1(glm.tit.4,.~.+Survived:(Sex*Age*Class),test="Chisq")
Single term additions

Model:
Freq ~ Class + Sex + Age + Survived + Class:Sex + Class:Age +
    Sex:Age + Sex:Survived + Class:Survived + Age:Survived +
    Class:Sex:Age + Class:Sex:Survived
                  Df Deviance     AIC     LRT   Pr(Chi)
<none>                45.899 223.301
Sex:Age:Survived    1   37.263 216.665   8.637  0.003295 **
Class:Age:Survived  3    1.685 185.088  44.214 1.359e-09 ***
```

a pokračujeme dále ...

```
> glm.tit.5<-update(glm.tit.4,.~.+Class:Age:Survived)
> add1(glm.tit.5,.~.+Survived:(Sex*Age*Class),test="Chisq")
Single term additions

Model:
Freq ~ Class + Sex + Age + Survived + Class:Sex + Class:Age +
    Sex:Age + Sex:Survived + Class:Survived + Age:Survived +
    Class:Sex:Age + Class:Sex:Survived + Class:Age:Survived
                Df  Deviance     AIC     LRT Pr(Chi)
<none>               1.685 185.088
Sex:Age:Survived  1 4.239e-10 185.402   1.685  0.1942
```

Interakce mezi věkem, pohlavím a přežitím (kterou by bylo např. možné interpretovat tak, že větší šanci na přežití měla děvčata ve srovnání s chlapci) již průkazná není!

Z toho mimo jiné vyplývá[14], že nemá smysl se zabývat nejsložitější možnou interakcí, tj. *Sex:Age:Class:Survived*.

[14] Tedy alespoň pro některé lidi, zabývající se statistikou, protože přístup k interakcím ve vztahu k hlavním efektům a k interakcím nižšího řádu se mezi nimi často liší

Budeme proto model *glm.tit.5* považovat za konečný. Nicméně, dosud jsme si udělali jen hrubou představu, které faktory - případně jejich interakce - souvisejí s pravděpodobností přežití, nicméně k využití těchto výsledků si musíme udělat také představu, v jaké míře a v jakém směru je tato pravděpodobnost měněna. Následující kuchařka ukazuje postup, kterým si vytvoříme tabulku pravděpodobností přežití

```
> tit.names<-lapply(tit[,-5],levels)
> tit.names
$Class
[1] "1st"  "2nd"  "3rd"  "Crew"
$Sex
[1] "Male"   "Female"
$Age
[1] "Child" "Adult"
$Survived
[1] "No"  "Yes"

> tit.pm<-predict(glm.tit.5, expand.grid(tit.names),type="response")
```

Funkce *expand.grid* nám zajistila vytvoření datového rámce s hodnotami vysvětlujících proměnných odpovídajícími každé možné kombinaci hladin faktorů, které jsou definovány seznamem v proměnné *tit.names*. Parametr *type="response"* byl zase nutný, aby byly s použitím GLM předpovídány počty, nikoliv hodnoty na škále lineárního prediktoru (zadali jsme typ distribuce *poisson*, kterému odpovídá implicitní *log* link funkce).

```
> tit.pm<-matrix(tit.pm,ncol=2,byrow=F, dimnames=list(NULL,tit.names[[4]]))
```

Předpovídané hodnoty jsme v předcházejícím příkazu pouze "přetvarovali" do matice se dvěma sloupci, a to tak, že předpovídané počty osob pro hodnoty *No* a *Yes* faktoru *Survived* jsou v matici (pro každou kombinaci zbývajících faktorů) umístěny vedle sebe.

Poslední výpočetní krok, který nám zbývá, je změnit absolutní počty na relativní podíly, tj. předpovídané pravděpodobnosti. To zajistíme vydělení obou hodnot v každém řádku matice řádkovým součtem (operátor %*% představuje násobení dvou matic, resp. v našem případě matice *tit.pm* a vektoru *[1 1]*).

```
> tit.pr<-tit.pm/drop(tit.pm %*% rep(1,2))
> tit.pr
              No        Yes
 [1,] 0.00000000 1.00000000
 [2,] 0.00000000 1.00000000
 [3,] 0.77916667 0.22083333
 [4,]        NaN        NaN
 [5,] 0.00000000 1.00000000
 [6,] 0.00000000 1.00000000
 [7,] 0.47096774 0.52903226
 [8,]        NaN        NaN
 [9,] 0.67428571 0.32571429
[10,] 0.91666667 0.08333333
[11,] 0.83246753 0.16753247
[12,] 0.77726218 0.22273782
```

```
[13,] 0.02777778 0.97222222
[14,] 0.13978495 0.86021505
[15,] 0.55393939 0.44606061
[16,] 0.13043478 0.86956522
```

Je docela přirozené, že výsledek dělení je *NaN* v řádcích 4 a 8, protože tyto řádky odpovídají dětem (obou pohlaví), které by cestovaly jako posádka. Nakonec již jen přidáme popisky řádků a předpovídané pravděpodobnosti zpřehledníme zaokrouhlením:

```
> tit.pr<-cbind(expand.grid(tit.names[-4]),prob=round(tit.pr,2))
> tit.pr
   Class    Sex   Age prob.No prob.Yes
1    1st   Male Child    0.00     1.00
2    2nd   Male Child    0.00     1.00
3    3rd   Male Child    0.78     0.22
4   Crew   Male Child     NaN      NaN
5    1st Female Child    0.00     1.00
6    2nd Female Child    0.00     1.00
7    3rd Female Child    0.47     0.53
8   Crew Female Child     NaN      NaN
9    1st   Male Adult    0.67     0.33
10   2nd   Male Adult    0.92     0.08
11   3rd   Male Adult    0.83     0.17
12  Crew   Male Adult    0.78     0.22
13   1st Female Adult    0.03     0.97
14   2nd Female Adult    0.14     0.86
15   3rd Female Adult    0.55     0.45
16  Crew Female Adult    0.13     0.87
```

Je vidět, že na pasažéry ve třetí třídě nemělo pravidlo preference dětí a žen tak výrazný dopad…

Analýza velikostí a hmotností

Gamma distribuci pro nevysvětlenou variabilitu předpokládáme hlavně pro vysvětlované proměnné představující buď různá měření délky, šířky, objemu či hmotností, nebo také poměry takovýchto veličin. V této kapitole se zaměříme na prvý případ, analýze poměrů se budeme věnovat v kapitole následující. Obecnou vlastností vysvětlovaných proměnných, pro které gamma distribuci volíme, je to, že hodnoty jsou striktně pozitivní (tj. např. organismy nemohou mít zápornou ani nulovou velikost či hmotnost, nula by znamenala chybějící hodnotu, kterou v jazyce S zaznamenáváme odlišně), a také to, že variabilita kolem očekávané hodnoty s touto hodnotou roste, a to rychleji než v případě Poissonovy distribuce - s druhou mocninou očekávané hodnoty. V případě velikostí a hmotností je vhodnou link funkcí logaritmická (*log*) funkce, ale je potřeba vědět, že implicitním typem link funkce je zde tzv. inverzní funkce (tj. převrácená hodnota

lineárního prediktoru), která je vhodná v případě užití gamma distribuce pro vysvětlování poměrů (podílů) pozitivních veličin. Proto v případech, které probíráme v této kapitole, musíme u gamma distribuce link funkci explicitně zadat (tj. jako *family=Gamma(link=log)*).

Příkladová data budeme importovat z listu *Leaves* v XLS souboru s daty pro cvičení:

```
> leaves<-read.delim("clipboard")
> summary(leaves)
      area            length          width            distMW
 Min.   : 10.70   Min.   : 5.00   Min.   : 2.000   Min.   : 2.00
 1st Qu.: 56.50   1st Qu.:17.00   1st Qu.: 6.000   1st Qu.: 8.50
 Median : 89.10   Median :22.00   Median : 7.000   Median :11.50
 Mean   : 91.03   Mean   :21.17   Mean   : 7.167   Mean   :11.95
 3rd Qu.:112.15   3rd Qu.:25.00   3rd Qu.: 8.750   3rd Qu.:15.00
 Max.   :260.40   Max.   :40.00   Max.   :12.000   Max.   :24.00
```

Kdyby byly listy obdélníkového tvaru, předpokládali bychom vztah typu:

$$area = length * width$$

Nicméně, zde měřené listy jsou vejčitě-eliptické s různě protaženou špičkou, tj. s různou pozicí místa největší šířky na ose mezi bazí a špičkou listu (proměnná *distMW* udává tuto charakteristiku, nicméně jako absolutní vzdálenost místa největší šířky od báze, v mm).

Obecnějším způsobem může velikost listové plochy záviset na dalších parametrech (které lze snadno změřit bez zničení listu) např. takto:

$$area = b0*length^{b1}*width^{b2}*(distMW/length)^{b3}$$

a po zlogaritmování obou stran dostáváme:

$$log(area)=log(b0)+b1*log(length)+b2*log(width)+b3*log(distMW/length)$$

V tomto okamžiku je možná potřebné zdůraznit, že nám jde o efektivní předpověď plochy, která nemusí odpovídat nějaké geometrické teorii. V té bychom museli k výpočtu plochy určitě použít jak délku, tak šířku, nicméně pokud je mezi oběma veličinami výraznější korelace, můžeme preferovat model, který například pracuje jen s délkou. Je proto vhodné zvolit složitost modelu metodou postupného výběru. V případě loglineárního modelu v předchozí sekci jsme model vybírali ručně, zde ale můžeme použít částečně automatizovanou funkci *step*. Nejprve ale nafitujeme nulový model:

```
> glm.leaves.0<-glm(area~+1,family=Gamma(link=log),data=leaves)
```

Za povšimnutí stojí, že na rozdíl od výše uvedeného logaritmizovaného vztahu zde funkci *log* pro proměnnou *area* nepoužíváme – její užití vyplývá ze zvolené link funkce. Postupný výběr provedeme tak, že funkci *step* zadáme jednak výchozí (nulový) model, jednak rozsah složitosti modelů, ze kterých může vybírat (tj. seznam všech možných členů, které může model obsahovat), pomocí parametru *scope*:

```
> glm.leaves<-step(glm.leaves.0,scope=~log(length)+log(width)+
+ log(distMW/length))
Start:  AIC= 448.39
 area ~ +1

                    Df Deviance    AIC
+ log(length)        1     3.48 413.96
+ log(width)         1     4.39 416.59
<none>                    16.08 448.39
```

```
+ log(distMW/length)  1    15.90 449.88

Step:  AIC= 384.02
 area ~ log(length)

                     Df Deviance    AIC
+ log(width)          1     1.25 362.86
<none>                      3.48 384.02
+ log(distMW/length)  1     3.47 385.89
- log(length)         1    16.08 513.18

Step:  AIC= 342.81
 area ~ log(length) + log(width)

                     Df Deviance    AIC
<none>                      1.25 342.81
+ log(distMW/length)  1     1.22 343.70
- log(width)          1     3.48 411.76
- log(length)         1     4.39 440.77
```

Začátek každé části výstupu funkce *step* jsem zvýraznil tučným písmem. V části začínající slovem *Start* testuje *step* možné jednoduché změny stávajícího modelu (přidání jedné z vysvětlujících proměnných, resp. ponechání beze změny). Tyto čtyři varianty jsou seřazeny podle rostoucí hodnoty AIC, a tedy nejlepší možnost (kterou funkce *step* následně také zvolí) je první v pořadí. V prvém kroku je proto nejprve přidán logaritmus délky listu jako vysvětlující proměnná, v druhém pak logaritmus šířky, a v posledních kroku dospěje *step* k závěru, že žádné další vylepšení modelu není možné (měřeno pomocí AIC), tedy že pokles residuální deviance po přidání členu *log(distMW/length)* není dostatečně velký vzhledem ke zvýšené složitosti modelu. Výsledný model jsme uložili do objektu s názvem *glm.leaves* a z něj můžeme dostat i informaci o hodnotách regresních koeficientů:

```
> summary(glm.leaves)
Call:
glm(formula = area ~ log(length) + log(width), family = Gamma(link = log),
    data = leaves)
Deviance Residuals:
     Min        1Q    Median        3Q       Max
-0.50565  -0.10420  -0.01785   0.10152   0.38329

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.13210    0.17979   0.735    0.467
log(length)  0.83109    0.08352   9.951 2.94e-12 ***
log(width)   0.90945    0.10894   8.348 3.27e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
(Dispersion parameter for Gamma family taken to be 0.03135264)

    Null deviance: 16.0754  on 41  degrees of freedom
Residual deviance:  1.2544  on 39  degrees of freedom
AIC: 342.81
Number of Fisher Scoring iterations: 4
```

Rovnice použitelná pro odhad plochy listu z jeho délky a maximální šířky má tedy tuto podobu:

$$area = 1.14122 * length^{0.83109} * width^{0.90945}$$

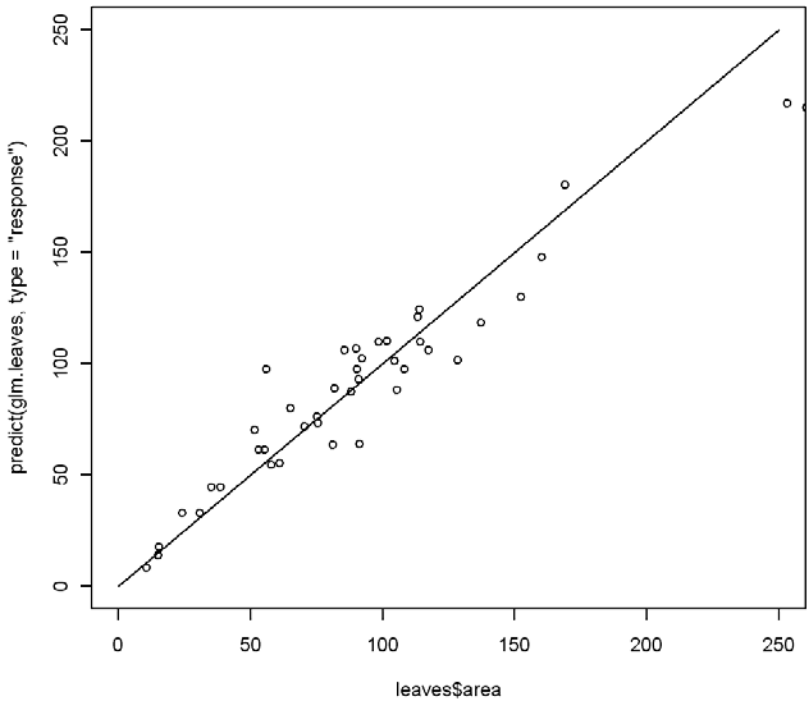
přičemž hodnotu *1.14122* jsem získal jako *exp(0.13210)*.

Kvalitu výsledného modelu můžeme číselně vyjádřit na základě deviance nulového modelu a residuální deviance, obdobně jako u koeficientu determinace:

```
> (16.0754-1.2544)/16.0754
[1] 0.9219677
```

a můžeme tedy říct, že náš model užívající šířku a délku vysvětlí asi 92% variability v hodnotách listové plochy. Graficky můžeme kvalitu modelu prozkoumat například vynesením předpovídaných hodnot plochy proti skutečným, spolu s referenční přímkou (odpovídající shodě mezi pozorovanými a předpovídanými hodnotami):

```
> plot(predict(glm.leaves,type="response")~leaves$area,
+ xlim=c(0,250),ylim=c(0,250))
> lines(c(0,250),c(0,250))
```



Obr. 12

Z obrázku je nejen vidět, že se variabilita skutečných hodnot zvětšuje s rostoucí hodnotou předpovídané plochy (to je ovšem u proměnné s gamma distribucí docela vpořádku), ale že náš model nepracuje příliš dobře pro největší listy (s plochou nad 250 mm^2), jejichž plocha je modelem výrazně podceněna (alespoň pokud lze soudit ze dvou existujících pozorování).

Zobrazení modelu se dvěma prediktory

Právě nafitovaný model popisující závislost velikosti listové plochy na délce a šířce listu nám umožňuje ozkoušet si způsob grafického znázornění hodnot, které pro vysvětlovanou proměnnou regresní model předpovídá pro různé kombinace hodnot dvou vysvětlujících proměnných. Se dvěma vysvětlujícími proměnnými (prediktory) můžeme buď vynést v trojrozměrné podobě nafitovaný odezvový povrch (response surface) nebo

stejnou informaci převést do vrstevnicového diagramu (contour plot). Postup je pro obě varianty shodný, až na poslední krok.

Nejprve si ujasníme rozsah hodnot prediktorů, pro které chceme hodnoty vysvětlované proměnné zobrazit:

```
> summary(leaves)
      area            length          width            distMW
 Min.   : 10.70   Min.   : 5.00   Min.   : 2.000   Min.   : 2.00
 1st Qu.: 56.50   1st Qu.:17.00   1st Qu.: 6.000   1st Qu.: 8.50
 Median : 89.10   Median :22.00   Median : 7.000   Median :11.50
 Mean   : 91.03   Mean   :21.17   Mean   : 7.167   Mean   :11.95
 3rd Qu.:112.15   3rd Qu.:25.00   3rd Qu.: 8.750   3rd Qu.:15.00
 Max.   :260.40   Max.   :40.00   Max.   :12.000   Max.   :24.00
```

Rozsah délky listů, na kterých byl model fitován, je od 5 to 40 mm, rozsah šířek od 2 do 12 mm. Potřebujeme nyní definovat nejen limity tohoto rozsahu, ale i počet kroků, na které budou tyto rozsahy rozděleny – to ovlivňuje především přesnost grafického zobrazení výsledného modelu. V případě délky vystačíme s krokem 1 mm (budeme mít tedy 36 hodnot pro *length*: 5, 6, ..., 40), v případě šířky listu krok raději zmenšíme na 0.5 mm, abychom dosáhli rozumnou přesnost (21 kroků).

Takto definované hodnoty si uložíme do seznamu, s názvy odpovídajícími proměnným v původním datovém rámci *leaves*:

```
> leaves.pred<-list(length=5:40,width=seq(2,12,by=0.5))
> leaves.pred
$length
 [1]  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27
[24] 28 29 30 31 32 33 34 35 36 37 38 39 40
$width
 [1]  2.0  2.5  3.0  3.5  4.0  4.5  5.0  5.5  6.0  6.5  7.0  7.5  8.0  8.5
[15]  9.0  9.5 10.0 10.5 11.0 11.5 12.0
```

Nyní vytvoříme nový datový rámec, ve kterém budou jednotlivá "pozorování" (případy) představovat všechny možné kombinace hodnot dvou proměnných ze seznamu *leaves.pred*. Pro takovéto hodnoty pak budeme předpovídat hodnotu listové plochy pomocí našeho modelu. Bude to tedy celkem 36*21 = 756 hodnot.

```
> leaves.grid<-expand.grid(leaves.pred)
> dim(leaves.grid)
[1] 756   2
```

Teď tedy máme hodnoty prediktorů ve správném formátu, abychom mohli použít funkci *predict* pro výpočet hodnot vysvětlované proměnné nikoliv z dat použitých při fitování (jako jsme to dělali dříve), ale z těchto nových "dat". Současně si necháme vypočítat i standardní chyby těchto odhadů:

```
> leaves.area<-predict(glm.leaves,leaves.grid,type="response",se.fit=T)
> names(leaves.area)
[1] "fit"            "se.fit"         "residual.scale"
> summary(leaves.area)
               Length Class  Mode
fit            756    -none- numeric
se.fit         756    -none- numeric
residual.scale   1    -none- numeric
```

Nás budou zajímat dvě první části výsledku, který funkce *predict* vrátila. Pokud bychom nepožadovali výpočet standardních chyb, funkce *predict* by vrátila jen vektor

s fitovanými hodnotami, o délce 756. Začněme nejprve s vektorem *fit*, který přidáme do datového rámce *leaves.grid*:

```
> leaves.grid$area<-leaves.area$fit
```

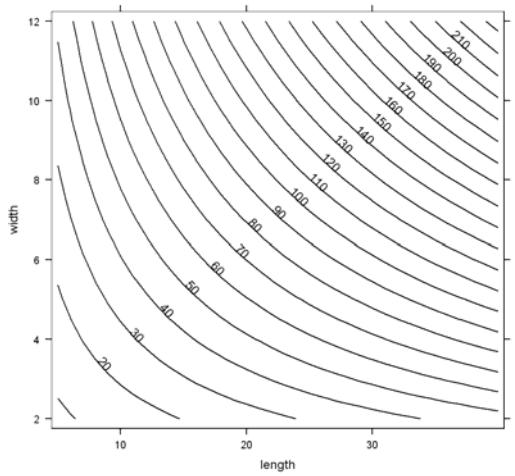
Obdobně můžeme přidat i standardní chybu pro jednotlivé předpovědi:

```
> leaves.grid$area.se<-leaves.area$se.fit
```

Obrázek s fitovaným modelem si vyneseme pomocí funkce *contourplot*, která je součástí balíčku *lattice* (ten se jmenuje v programu S-Plus *trellis*):

```
> library(lattice)
> contourplot(area~length*width,data=leaves.grid,cuts=30)
```

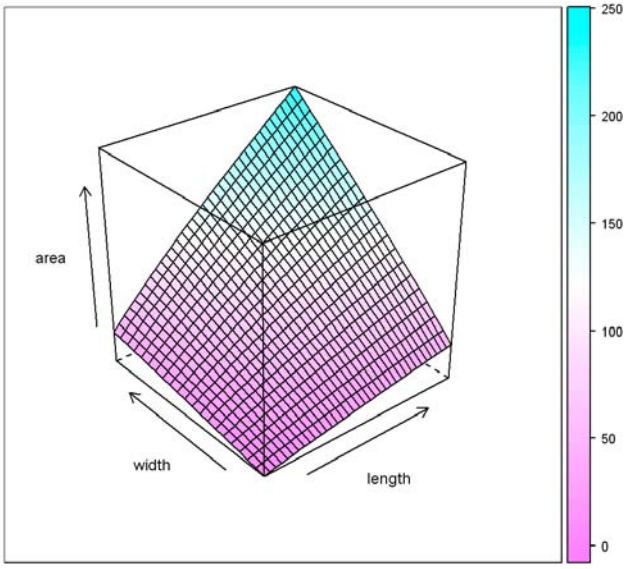
Výsledek je tady:



Obr. 13

Trojrozměrný diagram s regresním povrchem lze vytvořit podobně snadno, jen záměnou názvu funkce:

```
> wireframe(area~length*width,data=leaves.grid,drape=T)
```



Obr. 14

Na závěr, již bez obrázku, ukážu, jak vynést standardní chyby (nespolehlivosti) odhadů:

```
> contourplot(area.se~length*width,data=leaves.grid,cuts=12)
```

Pro výpočet konfidenční oblasti (např. 95%) bychom ale museli tyto chyby zkombinovat ve vzorečku, tj. například dolní hranici 95%-ního intervalu bychom vypočetli takto:

```
leaves.grid$area-qt(0.975,39)*leaves.grid$area.se
```

kde *39* je počet residuálních stupňů volnosti nafitovaného GLM.

3 Zobecněné lineární modely (GLM) – podíly

Motivační příklad

Tento příklad je převzatý z Venables & Ripley (2002). Byla zkoumána toxicita pyrethroidního insekticidu pro můry druhu *Heliothis virescens*. Vždy dvacet samců a dvacet samic bylo vystaveno jednotlivým koncentracím insekticidu (1 μg, 2 μg, 4 μg, 8 μg, 16 μg, 32 μg) a po třech dnech expozice byli spočítáni uhynulí jedinci. Výsledkem jsou tyto počty mrtvých jedinců:

	Dávka [μg]					
Pohlaví	**1**	**2**	**4**	**8**	**16**	**32**
Samci	1	4	9	13	18	20
Samice	0	2	6	10	12	16

Budeme modelovat závislost mortality (pravděpodobnost úmrtí jedince) na koncentraci insekticidu a na pohlaví, a také zkoumat, zda je mezi těmito dvěma proměnnými interakce (tj. zda reakce na odlišné koncentrace závisí na pohlaví). Také bychom rádi určili dávku, při které zemře polovina jedinců (tzv. LD50 statistika).

```
> budworm<-data.frame(ldose=rep(0:5,2),sex=factor(rep(c("M","F"),c(6,6))),
+  numdead=c(1,4,9,13,18,20,0,2,6,10,12,16))
> budworm
   ldose sex numdead
1      0   M       1
2      1   M       4
3      2   M       9
4      3   M      13
5      4   M      18
6      5   M      20
7      0   F       0
8      1   F       2
9      2   F       6
10     3   F      10
11     4   F      12
12     5   F      16
>
```

Dávky insekticidu byly zvoleny jako mocniny dvou (tj. každá je dvojnásobkem dávky předchozí) a protože také předpokládáme násobný (multiplikativní) efekt dávky, je výhodné používat jako vysvětlující proměnnou logaritmus se základem dvě. Tomu odpovídá proměnná *ldose*. Každá dvacítka jedinců, kteří sdíleli stejné experimentální podmínky (byli uzavřeni ve společném prostoru), představuje jedno nezávislé opakování a proto jí odpovídá jedna řádka v datech. Hodnoty v podobě "počet jevů" (v tomto případě smrt) z "celkového počtu možných" (v těchto datech vždy 20) mohou být dobře

popsány tzv. **binomickou distribucí**, alespoň pokud je pravděpodobnost úmrtí pro každého jedince nezávislá (tj. není ovlivněna tím, zda jiní jedinci zemřou nebo přežijí)[15]. V případě závislosti mezi jednotlivými případy dochází k odchylkám od binomické distribuce (které se mohou projevovat například tak, že jedinci umírají, přežívají nebo jsou infikováni v prostorově shloučených skupinách), které se nejčastěji projevují větší variabilitou pozorování, než bychom očekávali u binomické distribuce (takzvaná *overdispersion*, viz dále v této kapitole). Pro binomickou distribuci by mělo platit, že variabilita v počtu jedinců, u kterých daný jev nastal (např. zemřeli) je rovna $n_i*p*(1-p)$, kde p je pravděpodobnost jevu a n_i je počet pokusných případů pro i-tou skupinu.

Jiným příkladem pro data s binomickou distribucí mohou být semena umístěná v Petriho miskách, u kterých sledujeme za různých podmínek, kolik z nich vyklíčí (tj. jakou mají klíčivost). Ani v tomto případě nemusí být počty ve jmenovateli (tj. celkový počet jedinců pro danou řádku) shodné, ale je výhodné, pokud jsou.

Vysvětlovaná proměnná s binomickou distribucí

Vzhledem k tomu, že každé pozorování (skupina jedinců) je v případě dat s binomickou distribucí charakterizováno dvěma hodnotami, musíme vysvětlovanou proměnnou zadávat pomocí dvou vektorů. Funkce *glm* nám nabízí dva způsoby:

```
> glm.bw.1<-glm(cbind(numdead,20-numdead)~sex*ldose,
+ data=budworm,family=binomial)
> glm.bw.1x<-glm(numdead/20~sex*ldose,data=budworm,
+ family=binomial,weights=rep(20,12))
> coef(glm.bw.1)
(Intercept)        sexM       ldose  sexM:ldose
 -2.9935418   0.1749868   0.9060364   0.3529130
> coef(glm.bw.1x)
(Intercept)        sexM       ldose  sexM:ldose
 -2.9935418   0.1749868   0.9060364   0.3529130
```

V první variantě (model *glm.bw.1*) zadáváme vysvětlovanou proměnnou jako matici se dvěma sloupci (vytvoříme pomocí funkce *cbind*). První sloupec udává počet případů, kdy jev nastal, zatímco druhý udává počet případů, u kterých jev nenastal. Tedy pozor, druhý sloupec **neudává** celkový počet jedinců!

V druhé variantě (model *glm.bw.1x*) zadáváme jako vysvětlovanou proměnnou podíl případů z celkového počtu pokusných objektů (tj. číslo mezi 0 a 1), ale pomocí parametru *weights* (který je jinak u regresních modelů používán pro určení *a priori* vah pro jednotlivá pozorování) zadáme celkový počet případů, ze kterých byly podíly spočítány. V našem případě je to vždy 20 jedinců, ale nemusí tomu tak být vždy.

Pomocí funkce *coef* jsme si ověřili shodu výsledků pro oba postupy, ale měli bychom si prostudovat smysluplnost vlastního modelu (budeme pracovat např. s první variantou jeho fitování):

```
> summary(glm.bw.1)
...
Coefficients:
```

[15] Když už jsme u této kategorie "krutých příkladů", tak pokud by např. společně uzavřená pokusná zvířata soutěžila o potravu nebo vzduch, nebylo by jejich přežití resp. nepřežití vzájemně nezávislé.

```
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -2.9935     0.5527  -5.416 6.09e-08 ***
sexM          0.1750     0.7783   0.225    0.822
ldose         0.9060     0.1671   5.422 5.89e-08 ***
sexM:ldose    0.3529     0.2700   1.307    0.191

(Dispersion parameter for binomial family taken to be 1)
    Null deviance: 124.8756  on 11  degrees of freedom
Residual deviance:   4.9937  on  8  degrees of freedom
AIC: 43.104
Number of Fisher Scoring iterations: 4
```

Nejvýraznější je závislost mortality na dávce insekticidu, což nepřekvapí. Nicméně, další dva členy – interakce mezi pohlavím a dávkou a také hlavní efekt pohlaví – se nezdají být průkazné. Začněme od interakce[16]. Její význam bychom měli zhodnotit v případě binomické distribuce pomocí testu založeného na χ^2 statistice. Za platnosti nulové hypotézy (v tomto případě se týká interakčního efektu) by z χ^2 distribuce měla pocházet hodnota představující zvýšení deviance při vynechání interakčního členu (tedy také snížení deviance při jeho přidání).

```
> anova(glm.bw.1,test="Chisq")
Analysis of Deviance Table
Model: binomial, link: logit
Response: cbind(numdead, 20 - numdead)

Terms added sequentially (first to last)

          Df Deviance Resid. Df Resid. Dev P(>|Chi|)
NULL                         11    124.876
sex        1    6.077        10    118.799     0.014
ldose      1  112.042         9      6.757 3.499e-26
sex:ldose  1    1.763         8      4.994     0.184
```

Rychlost zvyšování mortality s rostoucí dávkou se tedy průkazně neliší mezi samci a samicemi (p=0.184). Hlavní efekt pohlaví ale je, z pohledu testu založeného na χ^2 statistice, průkazný. Pro správné posouzení vlivu pohlaví je ale třeba nejprve odstranit interakci a výsledný model porovnat s modelem, ve kterém chybí i hlavní efekt pohlaví:

```
> glm.bw.2<-update(glm.bw.1,.~.-sex:ldose)     # remove interaction term
> glm.bw.3<-update(glm.bw.2,.~.-sex)           # and remove main effect of sex
> anova(glm.bw.3,glm.bw.2,test="Chisq")
Analysis of Deviance Table

Model 1: cbind(numdead, 20 - numdead) ~ ldose
Model 2: cbind(numdead, 20 - numdead) ~ sex + ldose
  Resid. Df Resid. Dev Df Deviance P(>|Chi|)
1        10    16.9840
2         9     6.7571  1  10.2270    0.0014
```

Je vidět, že rozdíl mezi samci a samicemi je docela výrazný.

[16] To je obvyklý přístup: interakce jsou jakousi nadstavbou nad hlavními efekty, které spolu interagují, proto při zjednodušování modelu sáhneme nejprve k možnosti odstranění interakce (interakcí).

Zobrazení modelu

Zůstaneme proto u modelu *glm.bw.2* a vyneseme si do grafu jeho průběh:

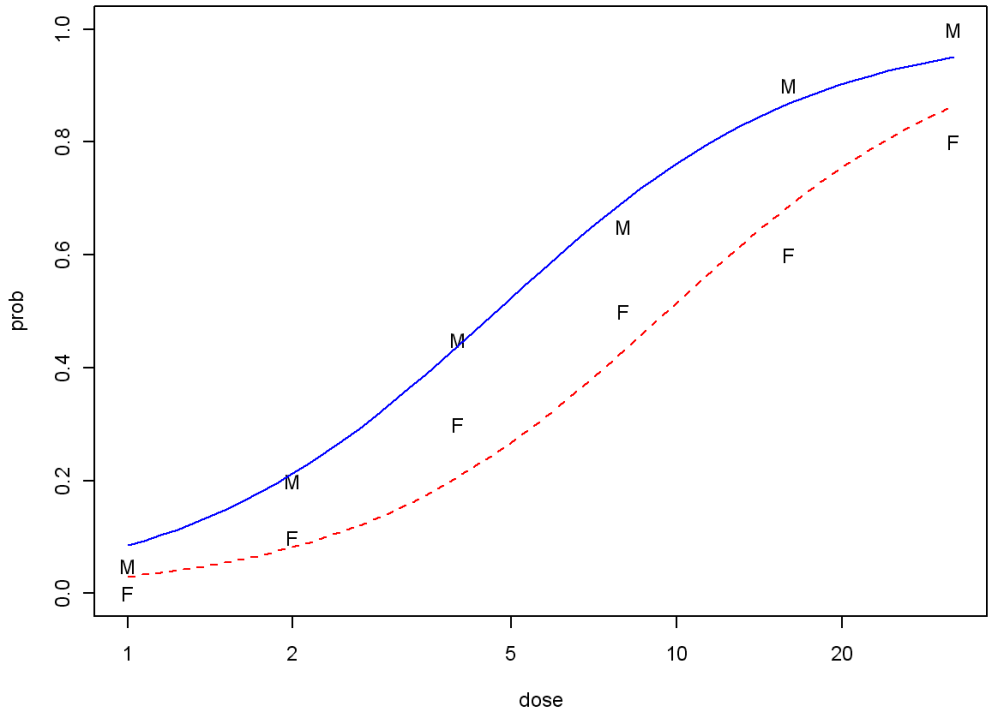
```
> plot(c(1,32),c(0,1),type="n",xlab="dose",ylab="prob",log="x")
> text(2^budworm$ldose,budworm$numdead/20,labels=as.character(budworm$sex))
```

První příkaz (volající funkci *plot* s typem obsahu *"n"*) vytvoří prázdný diagram, ale s definovaným rozsahem a škálováním os (horizontální osa má logaritmické škálování) a také s definovanými popiskami pro obě osy. Druhý příkaz vynáší podíly mrtvých jedinců, a to pomocí písmen odpovídajících jejich pohlaví. Nyní vyneseme vlastní fitovaný model, který je vzhledem k přítomnosti členu, který rozlišuje mezi samci a samicemi, představován dvěma paralelními[17] křivkami.

```
> ld<-seq(0,5,0.1)                 # binary log of dose
> lines(2^ld, predict( glm.bw.2,   # predict values for males and plot as line
+                      data.frame( ldose=ld,sex=factor(rep("M",length(ld)),
+                                                      levels=levels(budworm$sex))),
+                      type="response"), col="blue")
> lines(2^ld,predict(glm.bw.2,     # predict values for females and plot them
+                      data.frame(ldose=ld,sex=factor(rep("F",length(ld)),
+                                                      levels=levels(budworm$sex))),
+                      type="response"), col="red",lty=2)
```

Odsazení jednotlivých pokračovacích řádek není povinné, má za cíl pouze větší přehlednost příkazů. Funkce *lines* vynáší předpovídané hodnoty, přičemže funkci *predict* předáváme několik parametrů: fitovaný model, hodnoty prediktorů, pro které chceme předpovídané hodnoty získat, a dále parametr *type*, který udává, že vrácené hodnoty mají být na škále vysvětlované proměnné (*response*), tj. v našem příkladu pravděpodobnosti mezi 0 a 1. Výsledný diagram je v Obr. 15.

[17] Tyto křivky jsou "paralelní" na škále lineárního prediktoru, na škále pravděpodobnosti se absolutní vzdálenosti mezi křivkami mění (jsou největší pro pravděpodobnost 0.5).



Obr. 15

Měli bychom se ještě ujistit, že je zvyšování mortality s rostoucí logaritmovanou dávkou insekticidu opravdu lineární na škále prediktoru. Nejbližším složitějším způsobem popisu této závislosti je polynom druhého stupně:

```
> glm.bw.4<-update(glm.bw.2,.~sex+poly(ldose,2))
> anova(glm.bw.2,glm.bw.4,test="Chisq")
Analysis of Deviance Table

Model 1: cbind(numdead, 20 - numdead) ~ sex + ldose
Model 2: cbind(numdead, 20 - numdead) ~ sex + poly(ldose, 2)
  Resid. Df Resid. Dev Df Deviance P(>|Chi|)
1         9     6.7571
2         8     5.8505  1   0.9065    0.3410
```

Model *glm.bw.4* tedy není lepší než náš původní model, pokles deviance o *0.9065* neodpovídá průkaznému efektu.

Výpočet LD50

Zbývá nám určit LD50, tj. odhad dávky, která je letální pro 50% jedinců[18]. Základní výpočet není složitý. Pokud je vztah lineární, tj. platí, že

$$L(p) = b_0 + b_1 * x$$

[18] Nebo, řečeno více "statisticky", dávku, pro kterou je předpovídaná pravděpodobnost úmrtí rovna 0.5.

kde *L(p)* je link funkce, v našem případě logit, pak je hodnota prediktoru *x* definována pro pravděpodobnost *p* jako

$$\frac{L(p)-b_0}{b_1}$$

přičemž pro námi požadovanou hodnotu p=0.5 je logit funkce rovna 0 (a menší než 0 pro p<0.5 a pozitivní pro p>0.5). Hodnotu logaritmované dávky pro samice tedy spočteme snadno:

```
> coef(glm.bw.2)
(Intercept)        sexM       ldose
  -3.473155    1.100743    1.064214
> -(-3.473155)/1.064214
[1] 3.263587
```

a dávka v μg je pak rovna (umocnění se základem 2 je inverzní k binárnímu logaritmu)

```
> 2^3.263587
[1] 9.603678
```

Pro samce musíme hodnotu absolutního členu b_0 zvětšit o hodnotu koeficientu pro *sexM*:

```
> -(-3.473155+1.100743)/1.064214
[1] 2.229262
> 2^2.229262
[1] 4.688941
```

Vidíme, že samci jsou na insekticid výrazně citlivější, hodnota LD50 je pro ně poloviční. Pokud chceme spočítat LD50 s konfidenčním intervalem, resp. chceme spočítat například LD25 nebo LD75, výpočty již nejsou tak jednoduché. Package *MASS* nabízí funkci *dose.p*, která provádí výpočet nejen LD*nn*, ale i standardní chyby příslušného odhadu, a pokud předpokládáme, že tento odhad má normální distribuci, můžeme se ze standartní chyby dobrat i konfidenčního intervalu. Nicméně, funkce *dose.p* předpokládá, že pro naši lineární závislost máme vždy dvojici koeficientů, takže místo sčítání dvou koeficientů pro samce musíme náš model změnit, abychom měli jeden absolutní člen pro samice a jeden pro samce:

```
> glm.bw.2x<-update(glm.bw.2,.~sex+ldose-1)
> coef(glm.bw.2x)
     sexF      sexM     ldose
-3.473155 -2.372412  1.064214
```

Čtenáře upozorňuji, že modely *glm.bw.2* a *glm.bw.2x* jsou vpodstatě shodné (objasní stejné množství variability, vedou ke stejným křivkám závislosti), ale jejich parametry jsou odlišně definovány (modely jsou odlišně "parametrizované"). Pro výpočet pravděpodobností úmrtí používáme pro samice první a třetí koeficient, pro samce druhý a třetí. Tomu pak odpovídá volání funkce *dose.p*. Pro samice tedy určíme LD25, LD50 a LD75, spolu se standardními chybami odhadů, takto (parametr *cf* udává, které koeficienty v zadaném modelu použít):

```
> library(MASS)
> dose.p(glm.bw.2x,cf=c(1,3),p=c(0.25,0.5,0.75))
              Dose        SE
p = 0.25: 2.231265 0.2499089
p = 0.50: 3.263587 0.2297539
p = 0.75: 4.295910 0.2746874
```

a pro samce takto:

```
> dose.p(glm.bw.2x,cf=c(2,3),p=c(0.25,0.5,0.75))
              Dose        SE
p = 0.25: 1.196939 0.2635100
p = 0.50: 2.229262 0.2259649
p = 0.75: 3.261585 0.2549838
```

Funkce *dose.p* může být, mimo jiné, použita i v případě, že bychom zvolili jiný typ link funkce[19]. Je to proto, že funkce *dose.p* extrahuje typ distribuce z objektu, který nám vrátila funkce *glm*, a to pomocí funkce *family*. Ta opět vrací objekt, tentokrát představující kombinaci distribuce nevysvětlené variability (*binomial* v našem případě) se zvolenou (v našem případě implicitně) link funkcí. Link funkci lze z tohoto objektu extrahovat odkázáním na jeho složku, nazvanou *linkfun*. Postup si můžeme ozkoušet i sami, bude se nám jistě někdy hodit:

```
> family(glm.bw.2x)
Family: binomial
Link function: logit
```

Objekt vrácený funkcí *family* má předdefinovaný způsob, jakým se zobrazuje, takže nám utají většinou svých detailů. Více informací získáme pomocí funkce *names*, která ze seznamu (kterým objekt je) extrahuje jména všech obsažených složek (proměnných):

```
> names(family(glm.bw.2x))
 [1] "family"     "link"       "linkfun"    "linkinv"    "variance"
 [6] "dev.resids" "aic"        "mu.eta"     "initialize" "validmu"
[11] "valideta"
```

Mnohé z těchto položek jsou funkcemi a slouží při vlastním fitování GLM nebo při následné práci s modelem. Definice link funkce je v položce *linkfun*, zatímco *link* obsahuje jen název této funkce. Definice inverzní link funkce (převádí hodnoty lineárního prediktoru na škálu vysvětlovanné proměnné, v našem případě tedy na škálu pravděpodobností) je v položce *linkinv*.

```
> family(glm.bw.2x)$linkfun(0.5)
[1] 0
> family(glm.bw.2x)$linkfun(0.25)
[1] -1.098612
> family(glm.bw.2x)$linkinv(-2)
[1] 0.1192029
```

Nadměrná variabilita

U modelů s předpokládanou binomickou distribucí nevysvětlené variability se často objevuje tzv. nadměrná variabilita (*overdispersion*). Jde o situaci, ve které je variabilita pozorování kolem předpovídané hodnoty větší, než by odpovídalo binomické distribuci. Jak rozpoznáme případ nadměrné variability a jakým způsobem jí zohledníme, si ukážeme na příkladu výsledků experimentu, ve kterém byli jedinci dvou druhů vířníků odstřeďováni v médiích s různou hustotou. Počet jedinců, kteří přešli do supernatantu, a celkový počet jedinců v daném případě společně představují vysvětlovanou proměnnou,

[19] v případě binomické distribuce lze zvolit např. také *probit*, *cloglog* i další, všechny tyto funkce mapují rozsah hodnot pravděpodobnosti (0-1) na rozsah reálných čísel.

hustota média a druhová identita jsou dva prediktory. Data importujeme z excelovského souboru příkladových dat přes schránku:

```
> rotif<-read.delim("clipboard")
> summary(rotif)
     species      density            y                n
 KeraCoch:20   Min.   :1.019   Min.   :  7.00   Min.   : 14.00
 PolyMajo:20   1st Qu.:1.030   1st Qu.: 12.50   1st Qu.: 47.25
               Median :1.045   Median : 22.00   Median : 75.00
               Mean   :1.045   Mean   : 47.02   Mean   :109.13
               3rd Qu.:1.060   3rd Qu.: 40.50   3rd Qu.:160.25
               Max.   :1.070   Max.   :488.00   Max.   :492.00
```

Základní model s aditivními efekty druhu a hustoty média nafitujeme takto:

```
> glm.rot.1<-glm(cbind(y,n-y)~species+density,family=binomial,data=rotif)
```

První, orientační představu o nadměrné variabilitě můžeme získat z porovnání residuální deviance s počtem residuálních stupňů volnosti. K jejich zobrazení můžeme použít funkci *summary* (ale i *anova*):

```
> summary(glm.rot.1)
...
Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)     -113.17779    3.20304  -35.34   <2e-16 ***
speciesPolyMajo    1.42168    0.09816   14.48   <2e-16 ***
density          107.62275    3.06254   35.14   <2e-16 ***
...
(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 3180.99  on 39  degrees of freedom
Residual deviance:  434.25  on 37  degrees of freedom
AIC: 594.8
Number of Fisher Scoring iterations: 5
```

V případě binomické distribuce (a podobně to platí i pro Poissonovu distribuci) by měla být hodnota residuální deviance zhruba rovna residuálnímu počtu stupňů volnosti. Pokud model předpovídá hodnoty vysvětlované proměnné špatně (např. nulový model nebo model s prediktory, jejichž vliv je neprůkazný) a nebo pokud z nějakých vnějších příčin nejsou nezávislé jednotky, které se v rámci binomické distribuce sčítají (např. jedinci vířníků v našem případě), bývá hodnota residuální deviance výrazně větší než residuální DF. Poměr residuální deviance a residuálních DF (např. *434.25/37 = 11.74* v našem příkladě) pak odpovídá odhadu tzv. **dispersního parametru ϕ**.

Pokud je tento dispersní parametr výrazně odlišný od jedničky (viz též varovná poznámka ve výstupu z funkce *summary*), testy založené na předpokladu χ^2 distribuce hodnot vysvětlené deviance (tak jak jsme je výše prováděli pomocí funkce *anova* s parametrem *test="Chisq"*) ani testy založené na Waldově statistice (*z* hodnoty a následující signifikance ve výše uvedené tabulce koeficientů) nejsou správné. Jsou totiž příliš "optimistické", podceňují pravděpodobnost chyby I. druhu. Například pro Waldovu statistiku, která se počítá vydělením odhadu regresního koeficientu standardní chybou (standard error) tohoto odhadu, platí, že v případě nadměrné variability je její hodnota zhruba $\sqrt{\phi}$-krát větší, než by měla být, protože odmocnina z dispersního parametru je koeficient, kterým se musí násobit uvedené standardní chyby odhadů. Mimochodem, také

AIC statistika je zkreslená, protože člen zohledňující složitost modelu by měl být správně roven nikoliv *2*p*, jak jsme si dříve uváděli, ale *2*ϕ*p*.

Ačkoliv nám poměr residuální variance a residuálních DF může posloužit jako první indikátor nadměrné variability (overdispersion) ve většině případů (snad s výjimkou dat s Bernoulliho distribucí – viz další sekce), v případě její detekce musíme použít odlišný způsob práce se zobecněnými lineárními modely. Použijeme typ GLM, jehož odhad je založený na tzv. přibližné (zdánlivé) věrohodnosti[20] (*quasi-likelihood*). Program R podporuje tento ty p modelů, které lze fitovat buď velmi obecným způsobem (ve kterém nezadáváme konkrétní typ distribuce, jenom link funkci a funkci popisující vztah mezi variancí a očekávanou hodnotu vysvětlované proměnné) nebo za pomocí dvou specializovaných "distribucí", *quasipoisson* a *quasibinomial*. V obou případech jde o obdoby "klasických" distribučních předpokladů (tedy Poissonovy resp. binomické distribuce), u kterých je ale také odhadována hodnota dispersního parametru (a to jinou, přesnější metodou než jako poměr residuální deviance a počtu stupňů volnosti).

```
> glm.rot.2<-update(glm.rot.1,family=quasibinomial)
> summary(glm.rot.2)
...
Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)     -113.1778    11.7583  -9.625 1.29e-11 ***
speciesPolyMajo    1.4217     0.3603   3.945 0.000342 ***
density          107.6227    11.2425   9.573 1.49e-11 ***
...
(Dispersion parameter for quasibinomial family taken to be 13.47605)

    Null deviance: 3180.99  on 39  degrees of freedom
Residual deviance:  434.25  on 37  degrees of freedom
AIC: NA
Number of Fisher Scoring iterations: 5
```

Všimněme si ve výstupu, že jednak poklesly hodnoty signifikancí u dílčích testů regresních koeficientů, dále že odhad dispersního parametru se dosti liší od jednoduchého odhadu, který jsme vypočetli výše, a také toho, že funkce *summary* nepočítá hodnotu AIC. Je to proto, že autoři tohoto software nepovažují (na rozdíl od Chambers & Hastie, 1992) za správné počítat AIC (které vychází z modelu maximální věrohodnosti) v případě fitování pomocí zdánlivé věrohodnosti. Pokud toto omezení chceme obejít, je asi nejlepší použít funkci *anova*, s parametrem *type="Cp*", protože *Cp* (Mallows-ova) statistika je obdobou AIC pro lineární model a v případě GLM je její hodnota velmi blízká AIC[21].

Pokud ovšem při výběru modelu chceme zůstat u klasického přístupu statistické inference (tj. porovnávání modelů na základě testování hypotéz, nikoliv podle jejich *parsimony*), je třeba v případě nadměrné variability přejít k testům založeným na F statistice, nikoliv na χ^2 distribuci:

```
> anova(glm.rot.2,test="F")
Analysis of Deviance Table
Model: quasibinomial, link: logit
```

[20] Od klasické metody maximální věrohodnosti se liší v tom, že předpokládaná distribuce nevysvětlené variability není zcela a priori daná, jeden nebo více jejích parametrů odhadujeme při fitování modelu.
[21] Chambers & Hastie 1992 dokonce implikují (p. 235), že pořadí modelů založené na hodnotě Cp statistiky bude shodné s pořadím založeným na AIC.

```
Response: cbind(y, n - y)

Terms added sequentially (first to last)

        Df Deviance Resid. Df Resid. Dev       F    Pr(>F)
NULL                       39     3181.0
species  1    558.4        38     2622.6  41.439 1.610e-07 ***
density  1   2188.3        37      434.3 162.385 4.211e-15 ***
```

Bernoulliho distribuce

Bernoulliho distribuce je extrémním typem binomické distribuce, pro který je "počet zkoumaných případů" (pokusů) pro každé nezávislé pozorování roven jedné, takže vysvětlovaná proměnná má pro každé pozorování hodnotu buď 0 nebo 1: jedinec nepřežil nebo přežil, v této sezóně se rozmnožoval nebo ne, druh se na dané lokalitě vyskytoval nebo chyběl, atd. Zobecněný lineární model, ve kterém předpokládáme, že má vysvětlovaná proměnná Bernoulliho distribuci, zadáváme obdobně jako jsme viděli u proměnných s obecnou binomickou distribucí, nicméně ve vzorci modelu je pohodlnější zadat vysvětlovanou proměnnou buď jako numerický vektor s hodnotami 0 a 1 nebo jako vektor logických hodnot TRUE a FALSE. I u tohoto typu dat se často vyskytuje nadměrná variabilita (overdispersion), ale na rozdíl od proměnných s binomickou distribucí, které mají dostatečně velký počet pozorování ve jmenovateli, zde nemůžeme spoléhat na rozpoznání nadměrné variability z poměru residuální deviance k residuálnímu počtu DF (bývá často větší než jedna i pro data, která jsou "v pořádku"). Je proto asi vhodné *a priori* fitovat model s parametrem *family* zvoleným jako *quasibinomial* a v případě, že odhadnutá hodnota dispersního parametru není příliš odlišná od jedné, model přepočítat s parametrem *family=binomial*.

Příkladová data importujeme opět z excelovského souboru přes schránku:

```
> ReprEff<-read.delim("clipboard")
> summary(ReprEff)
    Survived         Flowers          Rhizome
 Min.   :0.0000   Min.   :  7.00   Min.   :0.0100
 1st Qu.:0.0000   1st Qu.: 21.00   1st Qu.:0.2000
 Median :0.0000   Median : 38.00   Median :0.2500
 Mean   :0.3898   Mean   : 48.76   Mean   :0.4986
 3rd Qu.:1.0000   3rd Qu.: 59.50   3rd Qu.:0.6600
 Max.   :1.0000   Max.   :165.00   Max.   :2.3600
```

Pomocí těchto dat budeme pro studovaný druh vytrvalé rostliny modelovat pravděpodobnost přežití jedince přes zimu, v závislosti na množství květů, které vytvořil v předcházející sezóně, a velikosti oddenku na konci sezóny.

Začneme nejprve s nulovým modelem a budeme jej postupně rozšiřovat:

```
> glm.re.0<-glm(Survived~+1,family=binomial,data=ReprEff)
> add1(glm.re.0,.~Flowers*Rhizome,test="Chisq")
Single term additions

Model:
Survived ~ +1
        Df Deviance    AIC    LRT Pr(Chi)
<none>       78.903 80.903
Flowers  1   75.008 79.008  3.895 0.04843 *
Rhizome  1   78.136 82.136  0.767 0.38114
```

Množství květů má - zdá se - vliv, zatímco velikost rhizomu pravděpodobnost přežití příliš nevysvětluje. Přidáme tedy nejprve proměnnou *Flowers* do modelu a zhodnotíme opět vliv proměnné *Rhizome*:

```
> glm.re.1<-update(glm.re.0,.~Flowers)
> add1(glm.re.1,.~Flowers*Rhizome,test="Chisq")
Single term additions

Model:
Survived ~ Flowers
        Df Deviance    AIC    LRT   Pr(Chi)
<none>       75.008 79.008
Rhizome  1   54.068 60.068 20.940 4.738e-06 ***
```

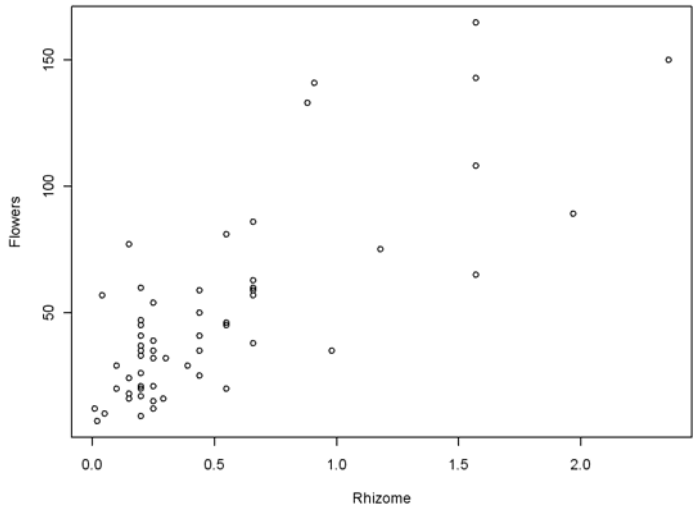
Výsledek je docela překvapivý: proměnná *Rhizome*, která samostatně nebyla příliš dobrou vysvětlující proměnnou, představuje kvalitní prediktor, pokud je v modelu již zohledněn vliv kvetení. Do modelu tedy rozhodně patří, ale musíme ještě vyjasnit příčinu toho neobvyklého jevu.

```
> glm.re.2<-update(glm.re.1,.~.+Rhizome)
> summary(glm.re.2)
...

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.96152    0.61676   1.559  0.11900
Flowers     -0.10642    0.03343  -3.183  0.00146 **
Rhizome      6.60042    2.10051   3.142  0.00168 **
...
(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 78.903  on 58  degrees of freedom
Residual deviance: 54.068  on 56  degrees of freedom
AIC: 60.068

...
```

Vidíme, že vlivy počtu květů a velikosti rhizomu jsou podobně silné[22], ale mají opačný směr (znaménko): čím rostlina více kvetla, tím nižší je pravděpodobnost přežití v následující zimě, a čím větší má oddenek, tím je tato pravděpodobnost větší. Jaký je ale vztah mezi těmito dvěma prediktory?

```
> plot(Flowers~Rhizome,data=ReprEff)
```

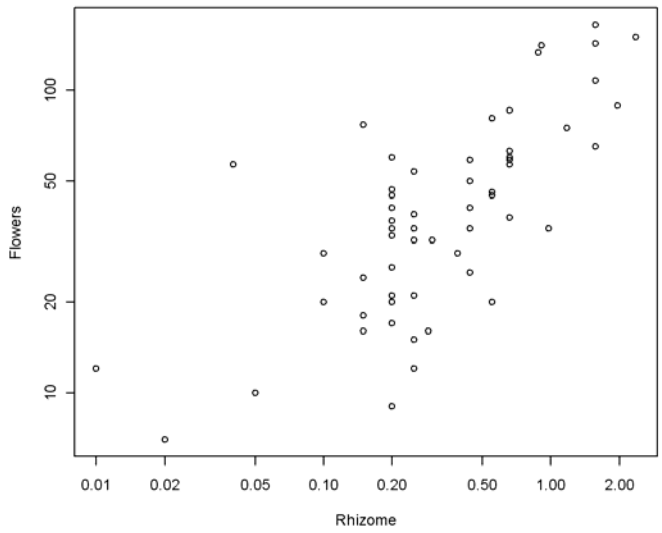
[22] Na velikost efektu nemůžeme usuzovat z velikosti regresního koeficientu – obě proměnné jsou měřeny v různých jednotkách, rozsah hodnot velikosti rhizomu je řádově nižší než počty květů, proto je také regresní koeficient u proměnné *Rhizome* řádově větší. Určitou indikaci velikosti efektu nám poskytuje *z* statistika.



Obr. 16

V diagramu (Obr. 16) vidíme především výraznou pozitivní korelaci mezi oběma vysvětlujícími proměnnými, ale zvětšující se rozptyl hodnot a také zakřivenost v jejich vztahu nám naznačují, že obě tyto proměnné bychom měli logaritmicky transformovat. Čtenáře, který o tom není zcela přesvědčen, snad více přesvědčí diagram (Obr. 17), ve kterém tuto transformaci provedeme, i když jen logaritmickým škálováním os, nikoliv transformací vlastních proměnných:

```
> plot(Flowers~Rhizome,data=ReprEff,log="xy")
```



Obr. 17

Ke stejnému závěru (logaritmická transformace obou proměnných) nás ale může dovést i představa (zahrnující i vysvětlovanou proměnnou, pravděpodobnost přežití), že efekt určité, konstantní velikost, bude na pravděpodobnost přežití mít spíše násobná změna

v počtu květů nebo velikosti rhizomu[23]. Podívejme se, jaké důsledky má logaritmická transformace na kvalitu modelu[24]:

```
> glm.re.3<-update(glm.re.2,.~log(Flowers)+log(Rhizome))
> anova(glm.re.2,glm.re.3,test="Cp")
Analysis of Deviance Table

Model 1: Survived ~ Flowers + Rhizome
Model 2: Survived ~ log(Flowers) + log(Rhizome)
  Resid. Df Resid. Dev Df Deviance     Cp
1        56     54.068             60.068
2        56     46.964  0    7.104 52.964
```

Nový model, který je stejně složitý (či spíše jednoduchý) jako *glm.re.2*, vysvětlil podstatně větší množství variability (deviance), a je proto kvalitnější, jak naznačuje nižší hodnota *Cp* (viz diskuse vztahu Cp a AIC v předchozí sekci). V tomto případě nemůžeme oba modely porovnat parametrickým způsobem (F test nebo χ^2 test), protože nedošlo ke změně počtu stupňů volnosti (dělili bychom nulou resp. Porovnávali s χ^2 distribucí s počtem stupňů .

Publikace Crawley (2002), ze které byla data převzata, jejich analýzu uzavírá modelem, který obsahuje také interakci mezi proměnnými *Flowers* a *Rhizome*. Ačkoliv by nám test signifikance ukázal, že obdobná interakce (mezi logaritmovanými prediktory) je průkazná i v našem přístupu, čtenáři její použití nedoporučuji. Interakce mezi dvěma kvantitativními vysvětlujícími proměnnými je definována jako součin jejich hodnot a je velmi obtížné (až nemožné) její smysl vysvětlit (součin počtu květů a velikosti rhizomu). Domnívám se, že interakci mezi kvantitativními prediktory je lepší popisovat pomocí neparametrických modelů (např. pomocí loess modelu, kterým se budeme zabývat v příští kapitole) a interpretovat pak výslednou grafickou podobu.

Poměry biomas a rozměrů

V případě, kdy nás zajímá poměr (podíl) dvou kvantitativních, kladných veličin, je vhodné použít zobecněný lineární model s předpokládanou gamma distribucí a logaritmickou, případně inverzní link funkcí. Logaritmická link funkce je vhodnou volbou zejména tehdy, kdy studujeme poměr dvou veličin (řekněme *A*/*B*), u kterých neplatí, že jedna je součástí druhé (tj. neporovnáváme třeba nadzemní biomasu trav s celkovou nadzemní biomasou ale např. biomasu trav s biomasou dvouděložných bylin). Pak totiž platí, že při shodě hodnot obou proměnných je poměr roven jedné a pokud má *A* hodnotu větší než *B*, roste koeficient nad hodnotou 1 neomezeně. Pokud je ale hodnota *A*

[23] Na příkladu proměnné *Flowers*: při změně počtu květů z 10 na 20 se energetická a/nebo materiálová investice rostliny do reprodukce zdvojnásobí, zatímco při změně např. ze 100 na 110 vzroste jen málo. Nicméně v regresním modelu, ve kterém hodnoty proměnné *Flowers* nejsou transformovány, mají obě změny (čistě z definici tohoto modelu) stejně velký vliv na změnu hodnoty lineárního prediktoru.

[24] Rozhodnutí o transformaci vysvětlujících (nebo i vysvětlovaných) proměnných by se ale nemělo řídit změnou ve vysvětlené variabilitě, protože ta není jedinou součástí kvality modelu. Často se stává, že modely, ve kterých by např. vysvětlující proměnná měla být transformována, vysvětlí více variability bez takové transformace (resp. ukáží průkazný vliv prediktoru jen bez jeho transformace), ale tato "kvalita" je pak obvykle dána jen přítomností několika extrémních ("odlehlých") hodnot, které mají na výsledný model nadměrný vliv. Rozhodnutí o transformaci by proto mělo být apriorní, před zkoumáním odhadnutého modelu.

menší než *B*, veškeré možné hodnoty jejich poměru jsou "stísněny" do rozsahu mezi 0 a 1. Logaritmování poměru *A/B* zesymetričtí vztah proměnných – výsledek je záporný pro *A*<*B*, roven nule pro *A* rovné *B*, a kladný pro *A*>*B*. Pro ilustraci takového typu modelů použijeme příkladová data, která importujeme z listu *Barley* v excelovském souboru s příklady:

```
> barley<-read.delim("clipboard")
> summary(barley)
     s.Hor            s.Sin         n.Hor           n.Sin
 Min.   :  0.00   Min.   :  0   Min.   : 0.00   Min.   :  0.00
 1st Qu.:  3.00   1st Qu.:  5   1st Qu.: 3.00   1st Qu.:  5.00
 Median :  7.00   Median : 10   Median : 7.00   Median : 10.00
 Mean   : 22.67   Mean   : 34   Mean   :18.07   Mean   : 32.91
 3rd Qu.: 34.00   3rd Qu.: 51   3rd Qu.:28.00   3rd Qu.: 51.00
 Max.   :115.00   Max.   :173   Max.   :70.00   Max.   :158.00
     w.Hor            w.Sin
 Min.   :  0.00   Min.   :  0.0
 1st Qu.:  3.30   1st Qu.: 81.5
 Median : 33.70   Median :209.8
 Mean   : 72.65   Mean   :171.0
 3rd Qu.:120.50   3rd Qu.:244.6
 Max.   :263.60   Max.   :332.0
```

Jde o výsledky pokusu, ve kterém byla studována kompetice mezi ječmenem (tomu odpovídají proměnné s názvy končícími na *.Hor*) a hořčicí (*.Sin*), nicméně různě vyvážené směsi obou druhů byly doplněny i pozorováními, ve kterých byly pěstovány jednodruhové kultury o různé hustotě. Ve všech případech je počáteční hustota (počet vysetých semen) dána proměnnými *s.Hor* a *s.Sin*. Stav na konci pokusu je pro oba druhy charakterizován počtem rostlin (*n.*Xxx) a celkovou nadzemní biomasou rostlin daného druhu (*w.*Xxx).

Protože se v našem příkladě zaměříme na kompetici mezi oběma druhy, vyloučíme pozorování, ve kterých jeden z druhů nebyl vyset:

```
> bar<-barley[(barley$s.Hor>0)&(barley$s.Sin>0),]
> dim(barley)
[1] 45  6
> dim(bar)
[1] 25  6
```

Základní model kompetice vysvětluje logaritmicky transformovaný poměr biomasy jednoho a druhého druhu počátečním poměrem vysetých semen (opět logaritmován) a také celkovou hustotou porostu (součet *s.Hor* a *s.Sin*):

$$\log(\frac{w_{Hor}}{w_{Sin}}) = \alpha + \beta * \log(\frac{s_{Hor}}{s_{Sin}}) + \gamma * \log(s_{Hor} + s_{Sin})$$

V případě, že by oba druhy byly kompetičně stejně silné, měl by koeficient u poměru počátečního počtu semen mít hodnotu 1 (koeficient větší než jedna odpovídá kompetitivní výhodě druhu v čitateli, tj. ječmene) a u logaritmu celkové density koeficient rovný nule (tj. celková densita ovlivňuje oba druhy stejně, nemění jejich očekávaný poměr). Nyní tento model nafitujeme pro naše data:

```
> glm.bar<-glm(w.Hor/w.Sin~log(s.Hor/s.Sin)+log(s.Hor+s.Sin),
+ data=bar, family=Gamma(link=log))

> anova(glm.bar,test="F")
```

```
Analysis of Deviance Table
Model: Gamma, link: log
Response: w.Hor/w.Sin

Terms added sequentially (first to last)
                   Df Deviance Resid. Df Resid. Dev        F    Pr(>F)
NULL                                  24     53.751
log(s.Hor/s.Sin)    1   47.960        23      5.791 205.2197 1.235e-12 ***
log(s.Hor + s.Sin)  1    1.072        22      4.718   4.5887   0.04351 *
```

Je důležité, abychom porozuměli významu poměru počátečních počtů semen coby vysvětlující proměnné. Že je poměr výsledných biomas závislý na počátečním počtu vyklíčených rostlinek se rozumí samo sebou a tak je jasné, že velké rozpětí počátečního poměru způsobí, že tento člen vysvětlí velkou část variability. Podobně i to, že je příslušný regresní koeficient průkazně odlišný od nuly, tak jak ukazuje následující část výstupu z funkce *summary*, není překvapivé.

```
> summary(glm.bar)
...
Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)        -2.20458    0.45291  -4.868 7.26e-05 ***
log(s.Hor/s.Sin)    0.96130    0.06098  15.764 1.80e-13 ***
log(s.Hor + s.Sin)  0.24830    0.11386   2.181   0.0402 *
...
(Dispersion parameter for Gamma family taken to be 0.2337026)
    Null deviance: 53.7512  on 24  degrees of freedom
Residual deviance:  4.7185  on 22  degrees of freedom
```

Biologicky zajímavější je ale otázka, zda je tento koeficient průkazně odlišný od jedné, což zjistíme upraveným *t*-testem (násobení 2 odpovídá testu oboustranné hypotézy):

```
> (1-pt((1-0.9613)/0.06098,22))*2
[1] 0.5322119
```

Koeficient β tedy není průkazně odlišný od jedné a ani jeden z druhů není jednoznačně kompetičně zdatnější. Nicméně koeficient γ je průkazně (i když na hranici, *p=0.04*) odlišný od nuly a jeho kladná hodnota naznačuje, že ječmen se vypořádává s rostoucí celkovou densitou lépe než hořčice.

Ještě se zastavme u otázky, jak zobrazit vztahy mezi vysvětlovanou proměnnou a těmi vysvětlujícími proměnnými, které jsme v modelu použili. Pokud chci zobrazit vztah poměru hmotností k poměru vysetých semen, prvním nápadem asi bude použít přímo hodnoty těchto vztahů (výsledný diagram nezobrazuji):

```
> plot(log(bar$w.Hor/bar$w.Sin)~log(bar$s.Hor/bar$s.Sin))
```

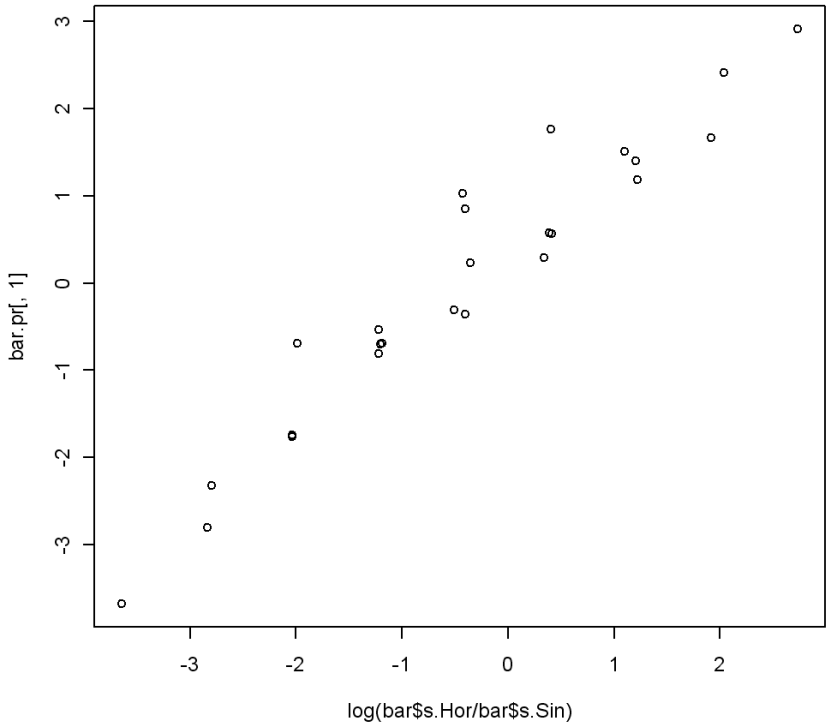
Tento graf ale odpovídá modelu, ve kterém není obsažen vliv celkové hustoty výsevu – ta totiž vysvětluje část variability v poměru biomas, zčásti se také vysvětlující schopnosti obou členů překrývají. Je proto třeba použít místo původní vysvětlované proměnné tzv. parciální residuály, představující variabilitu nevysvětlenou regresním modelem, ve kterém byl „vynechán“ (přesněji řečeno odečten) jeden z jeho členů. Pro náš model se dvěma prediktory (poměr počtů a součet počtů semen) dostáváme tedy dvě sady (dva vektory) parciálních residuálů – první z nich odpovídá vynechání vlivu poměru počtů semen z modelu a v těchto residuálech je tedy obsažena variabilita daná poměru biomas,

kterou je poměr počtů semen schopen vysvětlit navíc k tomu, co již vysvětlil vliv celkové hustoty.

```
> bar.pr<-residuals(glm.bar,type="partial")
> dim(bar.pr)
[1] 25  2
```

Správný typ diagramu dostaneme tedy takto (viz Obr. 18):

```
> plot(log(bar$s.Hor/bar$s.Sin), bar.pr[,1])
```



Obr. 18

Podívejme se ještě na regresní přímku, která takto definovanému vztahu odpovídá:

```
> summary(lm(bar.pr[,1]~log(s.Hor/s.Sin),data=bar))
...
Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)      0.40404    0.09778   4.132 0.000405 ***
log(s.Hor/s.Sin) 0.96130    0.05919  16.241 4.28e-14 ***
...
```

Není náhodou, že regresní koeficient je shodný s hodnotou, kterou jsme u stejného členu viděli výše u složitějšího GLM, ale není také náhodou, že odhad standardní chyby, a tím i hodnota *t* statistiky a odhad signifikance nejsou zcela shodné: zde jsme totiž použili metodu nejmenších čtverců v rámci klasického lineárního modelu, nikoliv metodu maximální věrohodnosti, kterou používá funkce *glm*. Vysvětlovanou proměnnou (tj. parciální residuály s vyloučeným vlivem poměru počtů semen) jsme nemuseli logaritmicky transformovat, protože tyto residuály jsou vyjádřeny na škále lineárního prediktoru.

4 Vyhlazování – loess smoother

Motivační příklad

Náš příklad, na kterém si základní práci s metodou loess ukážeme, nemá sice biologickou povahu, nicméně předpokládám, že tato data zaujmou čtenáře přinejmenším svým názvem:

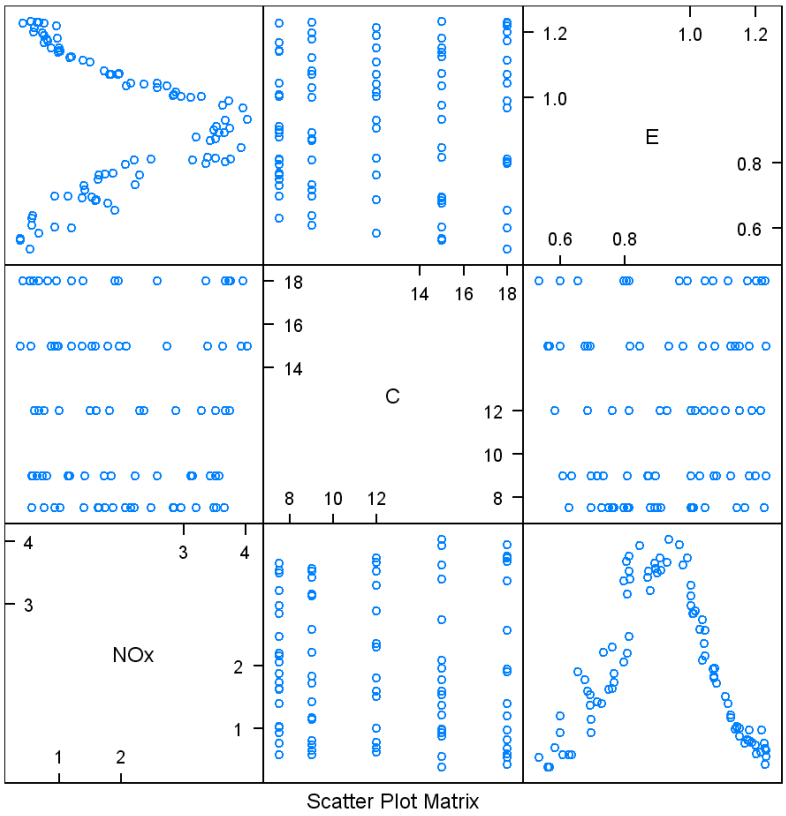
```
> data(ethanol)
> summary(ethanol)
      NOx              C                E
 Min.   :0.370   Min.   : 7.500   Min.   :0.5350
 1st Qu.:0.953   1st Qu.: 8.625   1st Qu.:0.7618
 Median :1.754   Median :12.000   Median :0.9320
 Mean   :1.957   Mean   :12.034   Mean   :0.9265
 3rd Qu.:3.003   3rd Qu.:15.000   3rd Qu.:1.1098
 Max.   :4.028   Max.   :18.000   Max.   :1.2320
```

Jde o údaje z technologického experimentu s pokusným, jednoválcovým motorem, ve kterém se spaloval ethanol za různých parametrů práce motoru – kompresní poměr[25] (proměnná compression ratio, *C*) a podíl vzduchu a paliva ve válci (proměnná equivalence ratio, *E*). Cílem studie bylo zjistit vliv těchto dvou parametrů na produkci zplodin – oxidů dusíku (proměnná *NOx*). Vztah mezi *NOx* na jedné straně a dvěma vysvětlujícími proměnnými na straně druhé si můžeme zobrazit pomocí dvou XY diagramů, můžeme ale také použít tzv. matici XY diagramů (scatter-plot matrix), ve které jsou vynášeny vždy dvě proměnné proti sobě (přičemž každá proměnná může být na vodorovné i na svislé ose):

```
> library(lattice)
> splom(~ethanol)
```

Výsledný diagram je v následujícím Obr. 19. Vidíme, že kombinace hodnot *C* a *E* byly zvoleny tak, aby byly tyto dvě vysvětlující proměnné co nejméně korelovány, a že proměnná *C* měla jen pět různých hodnot, zatímco proměnná *E* se měnila plynuleji. Nejvýraznější se zdá být závislost proměnné *NOx* na proměnné *E*, v podobě jednovrcholové křivky (panel v pravém dolním rohu matice). Naproti tomu se zdá, že na proměnné *C* proměnná *NOx* nezávisí vůbec.

[25] Jde o poměr objemů prostoru ve válci při minimálním a maximálním stlačení pístu.



Obr. 19

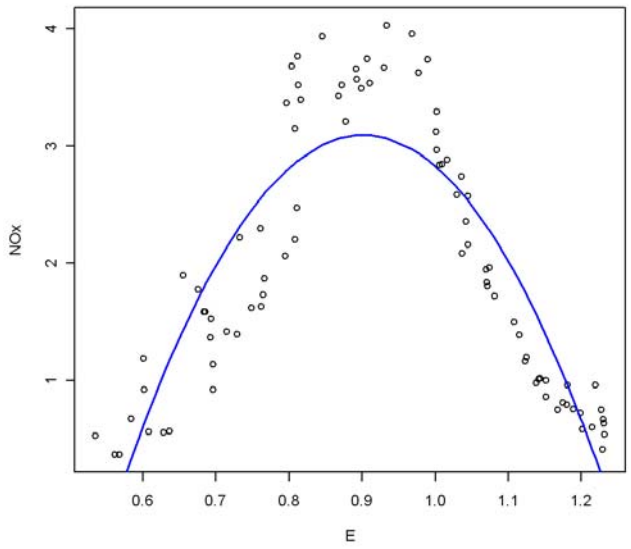
Pokud bychom chtěli zůstat u statistických modelů, které již známe, můžeme uvažovat o popsání závislosti pomocí lineárního modelu. Vysvětlující proměnná *E* v něm ovšem nemůže vystupovat jen jako lineární člen, měli bychom uvažovat o závislosti polynomické:

```
> lm.eth.1<-lm(NOx~poly(E,2),data=ethanol)
> anova(lm.eth.1)
Analysis of Variance Table

Response: NOx
           Df Sum Sq Mean Sq F value    Pr(>F)
poly(E, 2)  2 86.914  43.457  149.49 < 2.2e-16 ***
Residuals  85 24.710   0.291
---
```

Polynom druhého stupně vysvětluje velkou část variability *NOx* (kolem 77%), přesto ale neodpovídá skutečnému tvaru závislosti, jak uvidíme z grafu:

```
> plot(NOx~E,data=ethanol)
> ord.E<-order(ethanol$E)
> lines(ethanol$E[ord.E],fitted(lm.eth.1)[ord.E],col="blue",lwd=2)
```

Proměnná *ord.E* obsahuje indexy jednotlivých pozorování uspořádané tak, aby hodnoty proměnné *E* (v datovém rámci *ethanol*) byly setříděny vzestupně. To je pro vynášení modelem fitovaných hodnot proti *E* nutné, abychom dostali spojitou křivku. Výsledný diagram je v následujícím Obr. 20.



Obr. 20

Polynom druhého stupně[26] nepopisuje tvar závislosti věrně, tvar křivky je příliš "oblý" pro malé i velké hodnoty *E*, ve střední části jsou hodnoty *NOx* výrazně podceněny.

Loess model s jednou vysvětlující proměnnou

Přejdeme proto k metodě loess, kterou si na nafitujeme pomocí funkce *locfit* v rámci balíčku stejného jména[27]. Jednoduchý loess model získáme takto:

```
> lf.eth.1<-locfit(NOx~lp(E,nn=2/3),data=ethanol)
```

Na pravé straně vzorce modelu je proměnná *E* uzavřena ve funkci *lp*, označující tzv. lokální polynom. Co to znamená a jaký význam má parametr *nn* se dozvíme vzápětí, nejprve si ale model vyneseme. Pro porovnání s naším původním přístupem můžeme křivku vynést do výše uvedeného obrázku (doplněný graf není zobrazen), příkazem:

```
> lines(ethanol$E[ord.E],fitted(lf.eth.1)[ord.E],col="red",lwd=2)
```

Samostatný graf ale lehce vytvoříme pomocí funkce *plot*. Základní příkaz (výsledný obrázek opět neukazuji)

```
> plot(lf.eth.1)
```

zobrazí jen křivku nafitovaného modelu, bez původních hodnot pozorování. Následující volání funkce původní hodnoty zobrazuje (parametr *get.data*) a zobrazuje také konfidenční interval pro předpovídané střední hodnoty (parametr *band*):

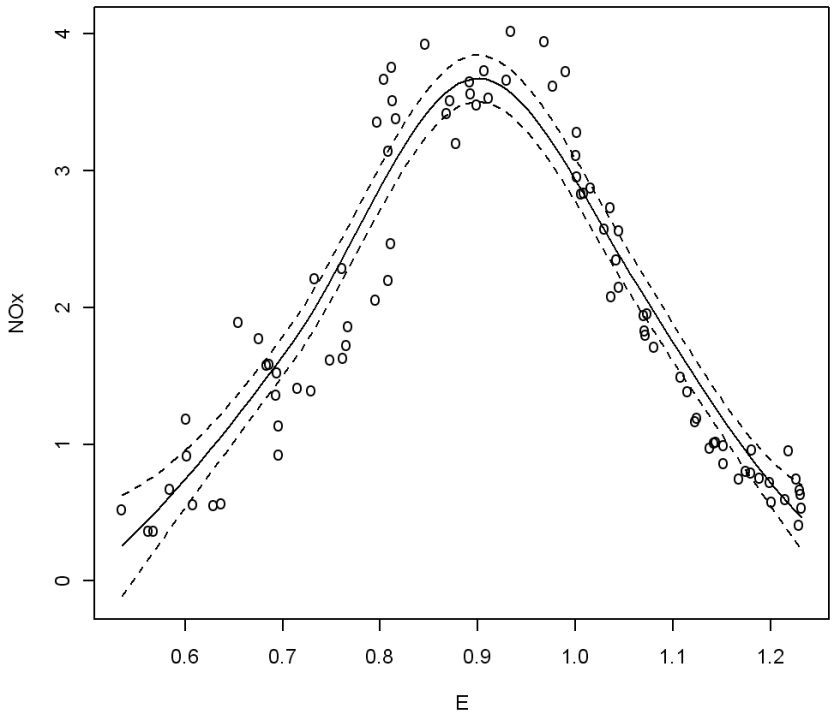
```
> plot(lf.eth.1,get.data=T,band="global")
```

Výsledný graf je v Obr. 21.

---

[26] Ale ani polynom třetího stupně, jak si může čtenář sám ozkoušet.

[27] Program R (podobně i S-Plus) nám již v základní implementaci nabízí funkci *loess*, nicméně její možnosti jsou značně omezené, pokud jde o vynášení grafické podoby modelu (jen v R, v S-Plus funguje obstojně), ale také pokud jde o pokročilejší postupy, například určování hodnoty parametru šířky pásma. Funkci *locfit* lze naopak použít i k fitování jiných typů regresních modelů.

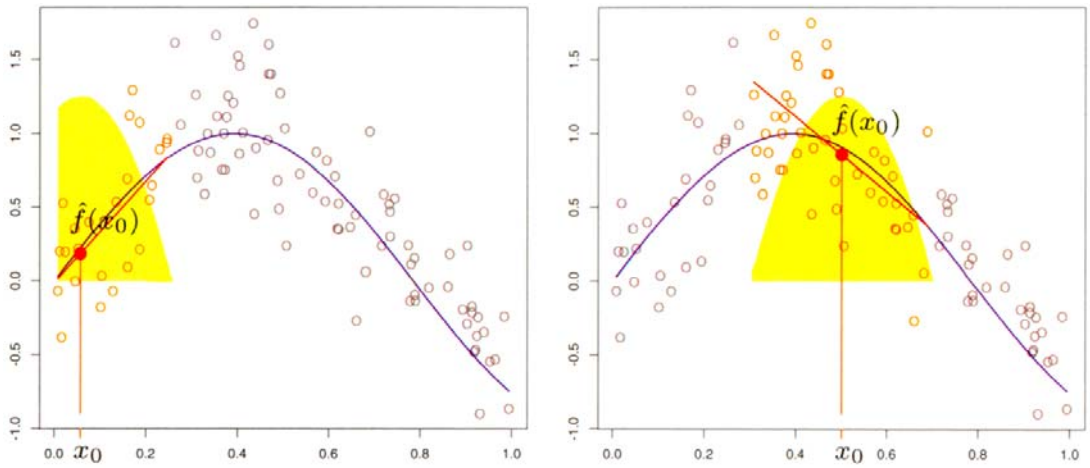


Obr. 21

Na základě zkušeností s lineárními a zobecněnými lineárními modely se čtenář nyní bude ptát, jaká rovnice odpovídá zobrazené křivce, jaké množství variability v hodnotách vysvětlované proměnné jsme vysvětlili, a za jakou cenu (počet stupňů volnosti). Začněme tedy od posledního bodu:

```
> summary(lf.eth.1)
Estimation type: Local Regression

Call:
locfit(formula = NOx ~ lp(E, nn = 2/3), data = ethanol)

Number of data points:  88
Independent variables:  E
Evaluation structure: Rectangular Tree
Number of evaluation points:  5
Degree of fit:  2
Fitted Degrees of Freedom:  4.958
```

Ve výstupu z funkce *summary* je zdůrazněna poslední řádka, ve které je uveden počet stupňů volnosti, které z našich 88 pozorování "spotřeboval" nafitovaný model. To, že složitost takovéhoto typu modelu nelze obecně vyjádřit celým číslem, je sice nezvyklé, ale není to špatně, musíme si na to jen zvyknout. Složitost výsledného loess modelu je určena dvěma parametry – tzv. šířkou pásma (*bandwidth*) a stupněm lokálního polynomu (*degree*). Co tyto dva parametry znamenají nejlépe pochopíme z obrázku, který ukazuje postup, jakým je loess model fitován (Obr. 22, převzato z Hastie et al. 2001, zjednodušeno).



Obr. 22

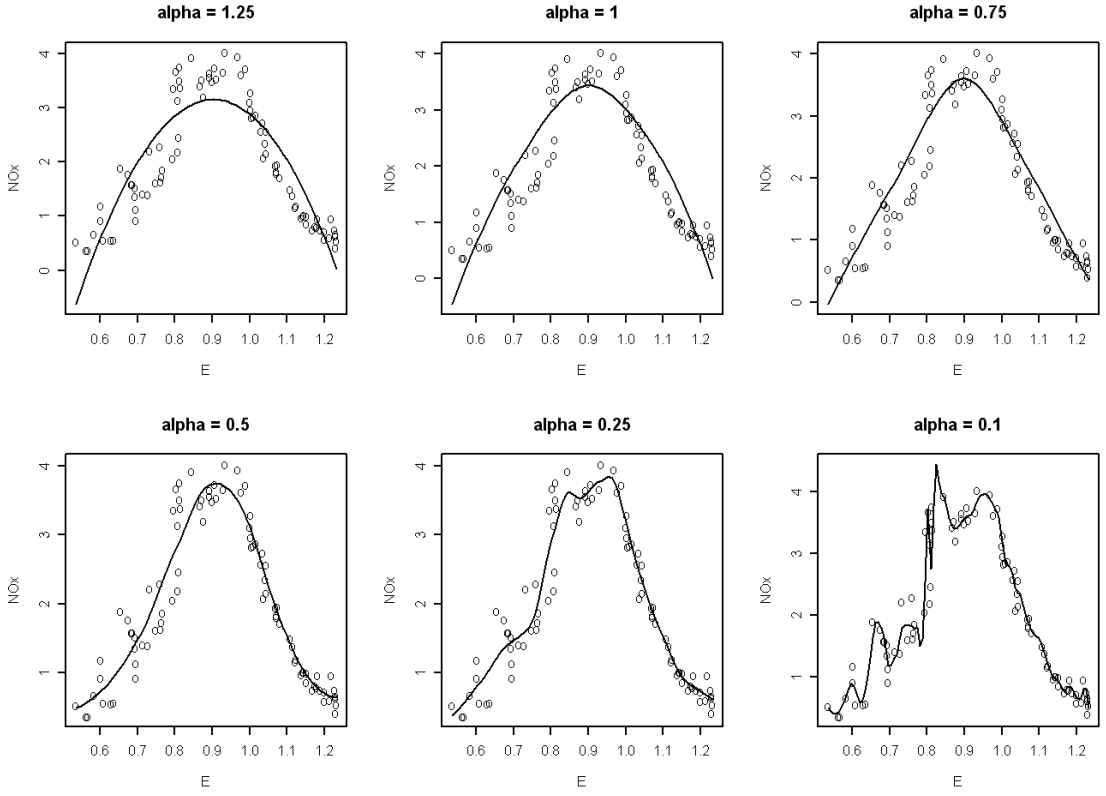
Fitování loess modelu si můžeme představit tak, že je očekávaná hodnota vysvětlované proměnné *y* vypočtena pro dostatečně velký počet hodnot vysvětlující proměnné *x*. V obrázku je znázorněn výpočet předpovídané hodnoty *y* jen pro dvě možné hodnoty *x* (označeny jako x_0) - jednu ležící na dolní hranici rozsahu *x* a druhou zhruba ve středu tohoto rozsahu. Loess model je fitován pomocí lokálního modelu, tj. v každém z bodů nejsou užívána všechna pozorování, ale jen jejich určitá část, ležící nejblíže uvažovanému bodu x_0.

Rozsah uvažovaného okolí se obvykle označuje jako šířka pásma (*bandwidth*) α a nejčastěji je jeho velikost určena jako podíl z celkového počtu pozorování. Například v našem výše uvedeném modelu *lf.eth.1* jsme šířku pásma určili hodnotou *nn=2/3*, tj. vždy se používají dvě třetiny ze všech 88 pozorování (tj. 59 bodů), ležící nejblíže uvažovanému bodu x_0. V příkladu na Obr. 22 je šířka pásma zjevně menší. Takto vybrané okolní body jsou v Obr. 22 zvýrazněny oranžovou barvou koleček. Následně je pro tyto body nafitován lineární model – v Obr. 22 je to regresní přímka, ale může to být například i polynom druhého stupně. Tato volba se nazývá stupeň (degree) lokálního polynomu λ (pro lokální přímku je hodnota 1, pro polynom druhého stupně pak 2).

Všechny vybrané body ale nemají v lokálním regresním modelu stejnou váhu. Největší váhu mají body, které mají hodnotu *x* shodnou s x_0, váha bodů se postupně snižuje k nule pro bod ležící na hranici uvažovaného okolí. Relativní váha je v Obr. 22 naznačena žlutou oblastí (křivkou). Hodnotu váhy udává obvykle tzv. trikubická funkce (i když lze zvolit i jiný typ symetrické jednovrcholové křivky).

Výsledný lokální regresní model je použit k určení očekávané hodnoty loess modelu pouze pro hodnotu prediktoru rovnou x_0 (červené vyplněné kolečko v obou případech).

Při volbě loess modelu musíme tedy zvolit hodnotu pro dva parametry: šířku pásma (α) a stupeň lokálního váženého modelu (λ, s hodnotami 1 nebo 2). Je jasné, že hodnota α ovlivňuje složitost modelu, a tím i složitost tvaru fitované křivky. Ukážeme si to na našich příkladových datech, která byla fitována loess modelem s λ=2 a měnící se hodnotou α (Obr. 23).



Obr. 23

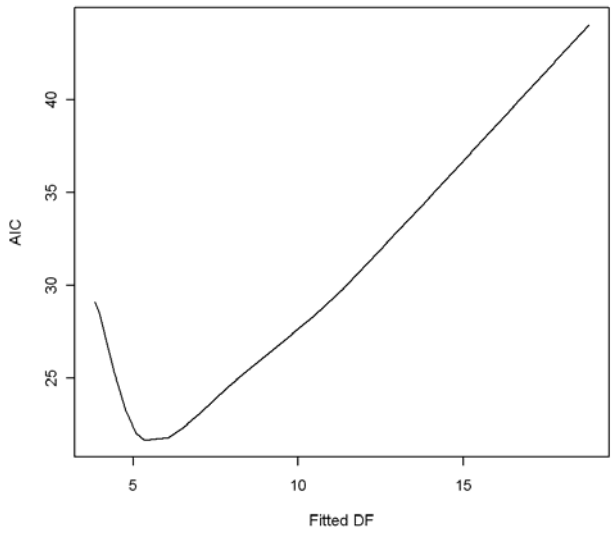
Vidíme, že modely s α rovným 0.25 a 0.1 jsou málo obecné - příliš ovlivněné konkrétními pozorováními v našem výběru. Alternativně je lze označit jako příliš složité, vzhledem k jejich vypovídací schopnosti. Naproti tomu modely s $\alpha > 0.75$ jsou příliš velkým zjednodušením vztahu mezi NOx a E.

Výběr parametru α na základě AIC

Čtenáře možná již napadlo, že bychom mohli vhodnou hodnotu α určit na základě úspornosti (parsimony) jednotlivých modelů. Hodnotu úspornosti (např. AIC statistiku) jsme schopni pro tyto modely vypočíst, protože známe množství neobjasněné variability (lze vyjádřit například pomocí variability residuálů) a známe složitost modelu (tu lze vyjádřit jako počet stupňů volnosti, ne nutně celé číslo, jak jsme viděli výše). Knihovna *locfit* nám takovýto výběr usnadňuje svojí funkcí *aicplot*:

```
> x<-aicplot(NOx~E,data=ethanol,alpha=seq(0.2,1.0,by=0.05))
> plot(x,type="l")
```

Výsledný graf vidíme v Obr. 24 a ukazuje závislost AIC hodnoty modelu na jeho složitosti (vyjádřené počtem stupňů volnosti).



Obr. 24

Při zadávání modelu ovšem nepoužíváme stupně volnosti, nýbrž hodnotu α (zadávanou parametrem *nn*), která je v proměnné *x* (vrácené funkcí *aicplot*) také uložena:

```
> names(x)
[1] "alpha"  "cri"    "df"     "values"
> cbind(x$alpha,x$values)
      [,1]     [,2]
 [1,] 0.20 44.06548
 [2,] 0.25 34.75360
 [3,] 0.30 30.03265
 [4,] 0.35 28.32876
 [5,] 0.40 25.36156
 [6,] 0.45 24.53621
 [7,] 0.50 23.08970
 [8,] 0.55 22.36692
 [9,] 0.60 21.78790
[10,] 0.65 21.65410
[11,] 0.70 22.03374
[12,] 0.75 23.27263
[13,] 0.80 24.45471
[14,] 0.85 25.37932
[15,] 0.90 26.94261
[16,] 0.95 28.47666
[17,] 1.00 29.12767
```

Vidíme, že hodnoty *nn* (α) blízké 0.65 vedou k modelu s nejvyšší úsporností.

Volba stupně lokálního modelu

Zatím jsme se vyhýbali otázce volby stupně modelu, tj. zda používat lokální lineární model nebo lokální polynom druhého či vyššího stupně. Pro jednodušší závislosti (rostoucí či klesající křivka nebo jednoduchá "oblá" křivka s maximem či minimem) se doporučuje hodnota λ rovná 1. Naopak pro složitější závislosti, ve kterých se směr křivky poměrně rychle mění, je nutné použít lokální polynom druhého stupně. Pokud bychom zvolili jen lineární lokální model, museli bychom příliš snížit hodnotu α, a nejspíše

bychom nezabránili "rozkmitání" křivky způsobem, který je vidět v Obr. 23 pro α=0.1. Použití polynomů vyššího stupně než druhého se nedoporučuje.

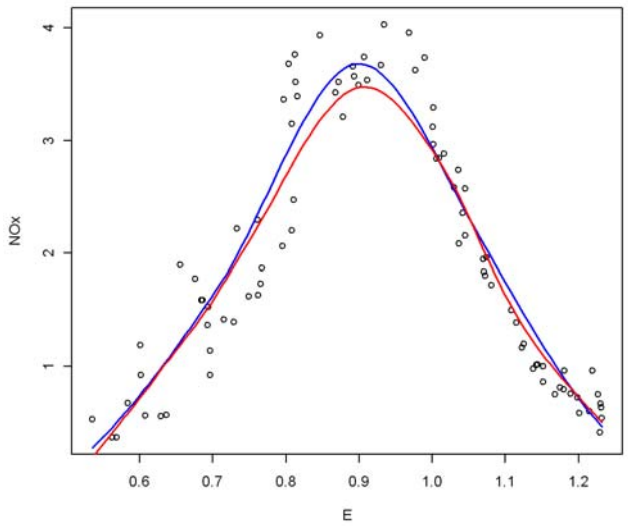
V mnoha konkrétních případech je ovšem obtížné se rozhodnout, zda námi studovaná závislost je "složitější" nebo "jednodušší". Pak si můžeme zvolit současně λ i α pomocí AIC statistiky:

```
> x2<-aicplot(NOx~E,data=ethanol,deg=1,alpha=seq(0.2,1.0,by=0.05))
> cbind(x$alpha,x2$values,x$values)
      [,1]     [,2]     [,3]
 [1,] 0.20 27.68957 44.06548
 [2,] 0.25 24.21036 34.75360
 [3,] 0.30 22.27071 30.03265
 [4,] 0.35 21.51887 28.32876
 [5,] 0.40 21.04721 25.36156
 [6,] 0.45 20.83453 24.53621
 [7,] 0.50 21.45839 23.08970
 [8,] 0.55 21.58329 22.36692
 [9,] 0.60 22.38306 21.78790
[10,] 0.65 24.31099 21.65410
[11,] 0.70 25.59105 22.03374
[12,] 0.75 27.59543 23.27263
[13,] 0.80 29.40467 24.45471
[14,] 0.85 31.72821 25.37932
[15,] 0.90 35.49152 26.94261
[16,] 0.95 41.31842 28.47666
[17,] 1.00 45.63887 29.12767
```

Vidíme, že z hlediska úspornosti je preferován model s lokálním lineárním modelem a menší šířkou pásma (α=0.45). Oba modely si můžeme porovnat, přičemž výsledný model vyneseme červenou barvou:

```
> lf.eth.2<-update(lf.eth.1,.~lp(E,nn=0.65))
> lf.eth.3<-update(lf.eth.1,.~lp(E,nn=0.45,deg=1))
> plot(NOx~E,data=ethanol)
> lines(lf.eth.2,col="blue",lwd=2)
> lines(lf.eth.3,col="red",lwd=2)
```

Výsledek by měl vypadat takto:



Obr. 25

Mnoho rozdílů mezi oběma křivkami nevidíme, snad jen to, že lokální přímkový model lépe popisuje vztah při okrajích rozsahu hodnot prediktoru *E*, zatímco lokální polynomiální model věrněji vystihuje závislost ve středu tohoto rozsahu, což je obecný závěr, který najdeme i v Hastie et al. (2001, s. 172). Lokálně-lineární model je celkově jednodušší, s odhadnutými DF modelu zhruba 4.1.

Závislost na dvou nebo více proměnných

Loess model je spíše než pro modely s jednou vysvětlující proměnnou[28] významný v situacích, kdy studujeme závislost vysvětlované proměnné na dvou nebo více kvantitativních prediktorech, u nichž předpokládáme interakci jejich vlivů. Ta může vypadat až tak, že se vztah k prediktoru *x1* kvalitativně (svým tvarem) mění s hodnotou prediktoru *x2* (nebo naopak). Jindy jde jen o dílčí proměny či posuny závislosti, loess model ale může být užitečný naopak i k tomu, abychom obavy z takovéto složité interakce vyvrátili (například srovnáním se zobecněným aditivním modelem, který uvedeme v příští kapitole).

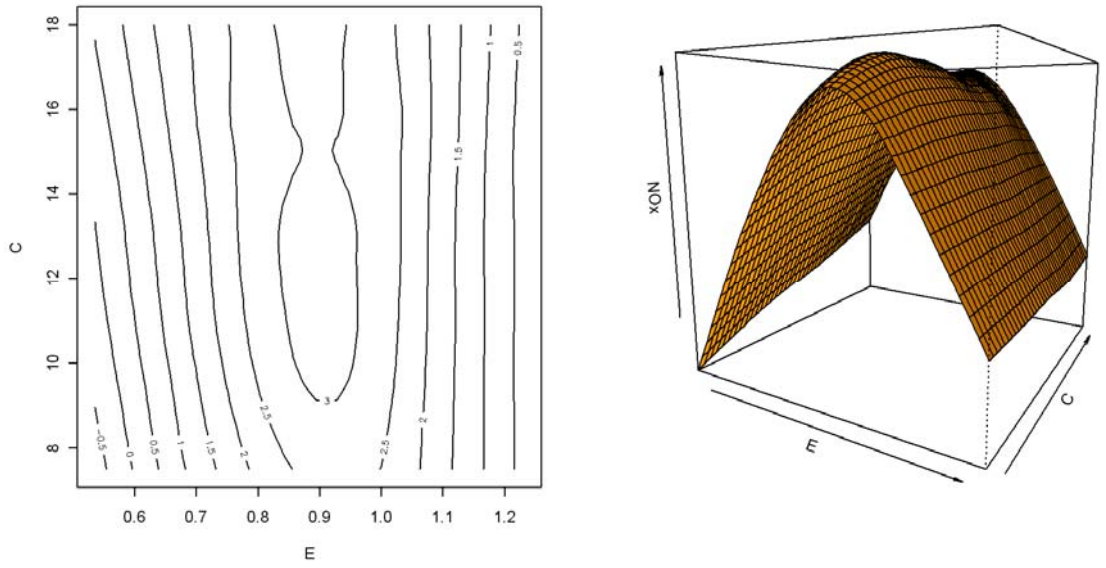
Studium interakcí si ukážeme na stejném příkladě, jaký jsme používali dosud. Podíváme se, jak závisí *NOx* na *E* a *C*, a jak se tyto závislosti mění podle hodnot druhého z prediktorů:

```
> lf.eth.4<-update(lf.eth.3,.~lp(E,C,nn=0.45,deg=1,scale=T))
```

V modelu jsme ponechali hodnotu α vybranou v situaci, kdy *E* bylo jediným prediktorem, což nemusí být správná volba v případě dvou prediktorů, ale pro počáteční seznámení tento model postačí. Důležitý je přidaný parametr *scale*, který zajistí, že pro hledání "nejbližších" bodů je brána v úvahu odlišná škála měření proměnných *E* a *C*. Výsledný model je trojrozměrný povrch (dva rozměry představují prediktory *E* a *C*, třetí pak vysvětlovaná proměnná *NOx*). Můžeme jej proto zobrazit buď jako konturový (isočárový) diagram nebo jako perspektivní náhled na tento povrch (výsledky obou přístupů jsou v Obr. 26).

```
> plot(lf.eth.4)
> plot(lf.eth.4,type="persp",theta=30,phi=10,shade=0.25,col="orange")
```

[28] Kde se může hodit coby "stavební kámen" např. při tvorbě zobecněných aditivních modelů (GAM)

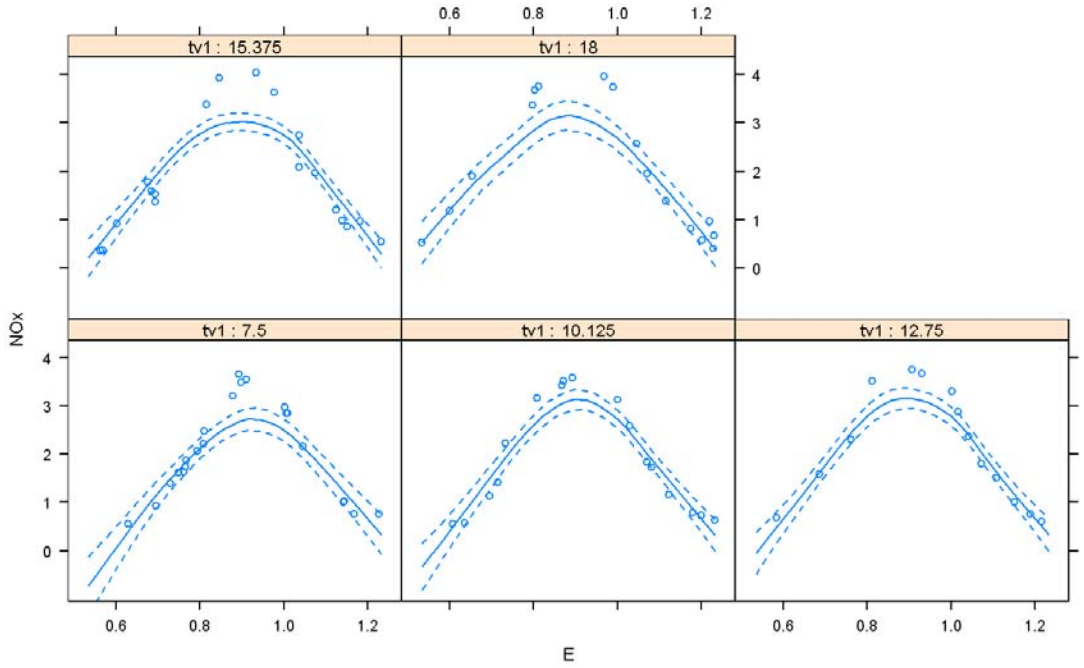


Obr. 26

V obou případech je ale obtížné oddělit základní tvar závislosti *NOx* na každém z obou prediktorů, a také interakci mezi těmito prediktory. V tom nám lépe pomůže tzv. podmíněný diagram (conditioning plot, často také nazývaný coplot). Jde o jeden z typů tzv. mřížových diagramů (trellis plot nebo také lattice plot), opět jej vytvoříme stejnou funkcí *plot*, pokud přidáme parametr udávající, která proměnná bude sloužit k podmiňování (k odlišení panelů; parametr *tv*) a která bude tvořit horizontální osu v každém z panelů (parametr *pv*):

```
> plot(lf.eth.4,pv="E",tv="C",mt=5,get.data=T,band="global")
```

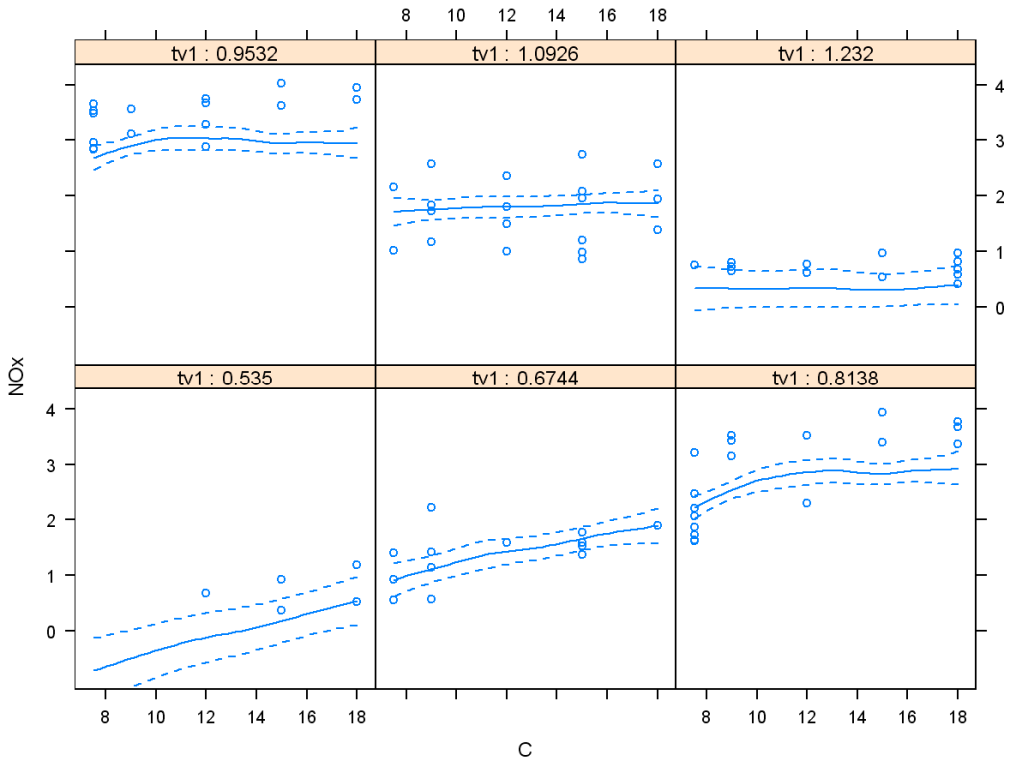
Výsledný diagram je vidět v následujícím obrázku:



Obr. 27

Každý z pěti (*mt=5*) zobrazených panelů ukazuje fitovanou závislost vysvětlované proměnné na zvolené vysvětlující proměnné (*pv="E"*), a to pro různé, na sebe navazující rozsahy hodnot proměnné podmiňující (trellis variable, *tv="C"*). Do diagramu byla přidána i původní data (*get.data=T*) a také 95% konfidenční oblast (*band="global"*). Tento diagram se zaměřuje na vliv *E*, pro různé hodnoty *C*, ale můžeme vytvořit i doplňující coplot s vyměněnou rolí proměnných *E* a *C* (Obr. 28).

```
> plot(lf.eth.4,pv="C",tv="E",mt=6,get.data=T,band="global")
```



Obr. 28

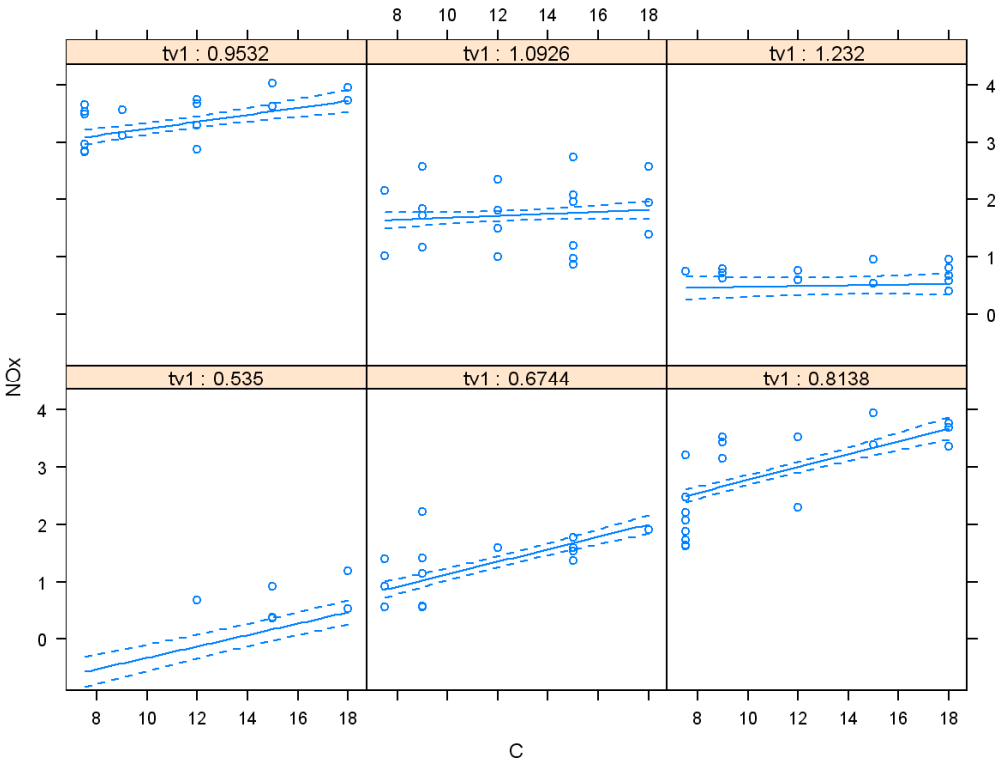
Z Obr. 28 je vidět nejen to, že *NOx* na hodnotách *C* závisí (což není vidět při pohledu na diagram, ve kterém je *NOx* vynášeno proti *C* bez ohledu na hodnotu *E*), ale že tato závislost mizí s rostoucí hodnotou *E*. Tvar této závislosti (pro nízké hodnoty *E*) je také velmi jednoduchý, proto bychom v tomto případě mohli použít jednoduchý přímkový model (ale s regresním koeficienty měnícími se s hodnotou *E*). Jak takový "hybridní" model nafitovat?

Závislost na proměnné *C* můžeme označit jako tzv. podmíněně parametrickou (conditionally parametric) závislost, a to v okamžiku fitování loess modelu:

```
> lf.eth.5<-update(lf.eth.3,
+ .~lp( E, C, nn=0.45, deg=1, scale=T, style=c("n","cpar"))
```

Podívejme se, jak to změnilo podobu závislosti *NOx* na *C*, pro různé hodnoty *E*:

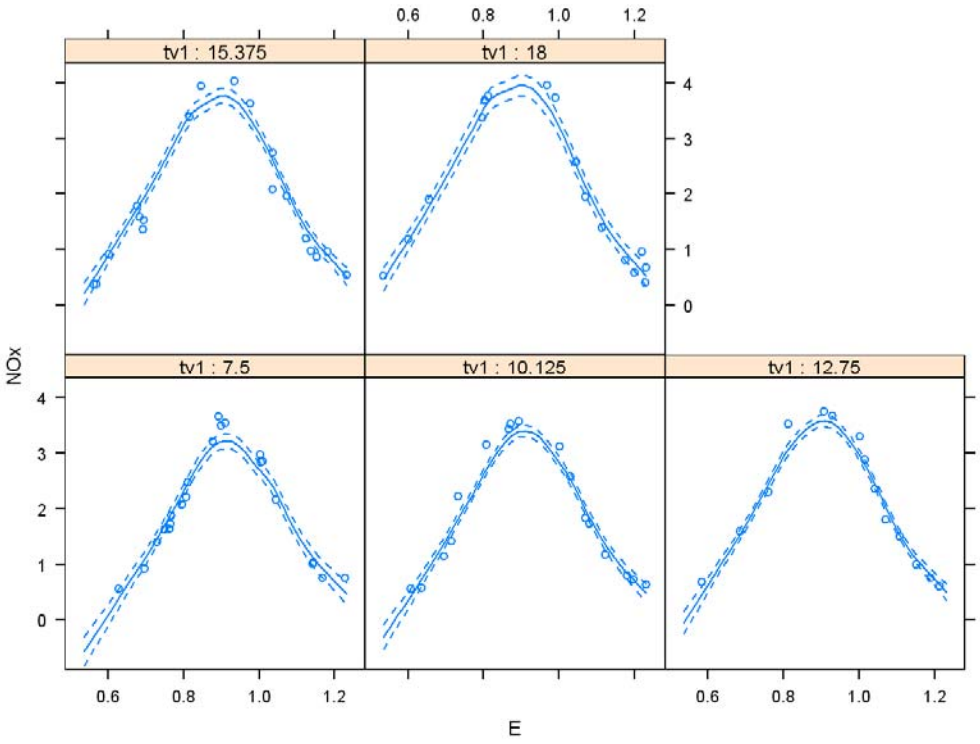
```
> plot(lf.eth.5,pv="C",tv="E",mt=6,get.data=T,band="global",aspect=1)
```



Obr. 29

Podmíněně lineární model fituje hodnoty poměrně dobře, snad s výjimkou pozorování s nejnižší hodnotou *E* (panel vlevo dole). Ještě se podíváme, co změna způsobu, jakým je vyjádřen v modelu vliv *C*, způsobila v popisu vlivu proměnné *E*:

```
> plot(lf.eth.5,pv="E",tv="C",mt=5,get.data=T,band="global")
```



Obr. 30

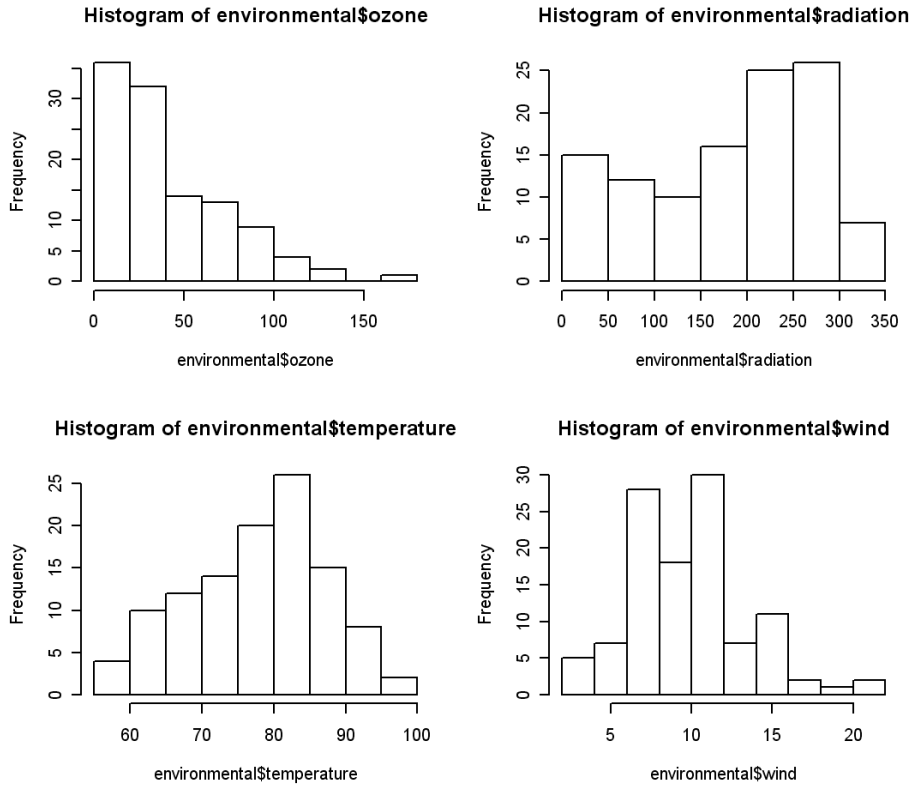
Výsledek je poměrně poučný (Obr. 30): lepší popis vlivu *C* způsobil i lepší fit ve vztahu k proměnné *E*: křivka dobře sedí i na body ve vrcholu závislosti (v porovnání s Obr. 27), aniž bychom museli zvyšovat hodnotu parametru λ z 1 na 2!

5 Zobecněné aditivní modely (GAM)

Motivační příklad

Budeme používat denní hodnoty koncentrace ozónu, rychlosti větru, teploty vzduchu a intenzity slunečního záření v New Yorku, měřené od května do září roku 1973.

```
> data(environmental,package="lattice")
> summary(environmental)
     ozone          radiation      temperature        wind
 Min.   :  1.0   Min.   :  7.0   Min.   :57.0   Min.   : 2.300
 1st Qu.: 18.0   1st Qu.:113.5   1st Qu.:71.0   1st Qu.: 7.400
 Median : 31.0   Median :207.0   Median :79.0   Median : 9.700
 Mean   : 42.1   Mean   :184.8   Mean   :77.8   Mean   : 9.939
 3rd Qu.: 62.0   3rd Qu.:255.5   3rd Qu.:84.5   3rd Qu.:11.500
 Max.   :168.0   Max.   :334.0   Max.   :97.0   Max.   :20.700
> par(mfrow=c(2,2))
> hist(environmental$ozone)
> hist(environmental$radiation)
> hist(environmental$temperature)
> hist(environmental$wind)
```

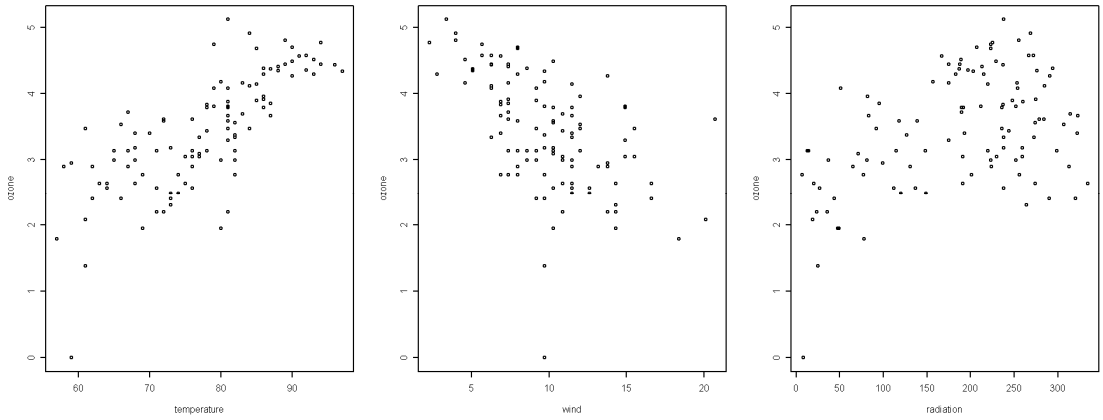


Obr. 31

Distribuce hodnot (Obr. 31) naznačuje, že přinejmenším koncentraci ozónu bychom měli transformovat, například logaritmicky.

```
> environmental$ozone<-log(environmental$ozone)
> par(mfrow=c(1,3))
```

```
> plot(ozone~temperature,data=environmental)
> plot(ozone~wind,data=environmental)
> plot(ozone~radiation,data=environmental)
> par(mfrow=c(1,1))
```



Obr. 32

Nejvýraznější vztah má logaritmovaná koncentrace ozónu (Obr. 32) k teplotě a k rychlosti větru, proto se zaměříme na tyto dva prediktory.

Parciální residuály v lineární regresi

Začneme ale nejprve lineárním modelem:

```
> lm.1<-lm(ozone~temperature+wind,data=environmental)
> anova(lm.1)
Analysis of Variance Table

Response: ozone
             Df Sum Sq Mean Sq F value    Pr(>F)
temperature   1 45.751  45.751 149.983 < 2.2e-16 ***
wind          1  3.774   3.774  12.373 0.0006384 ***
Residuals   108 32.945   0.305
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> coef(lm.1)
(Intercept) temperature        wind
-0.38602085  0.05653587 -0.05998093
```

Je vidět, že model popisující lineární vztah ozónu k teplotě a rychlosti větru má oba členy průkazně odlišné od nuly a že koncentrace ozónu roste s teplotou a klesá s rostoucí rychlostí větru. Již z Obr. 32 je ale vidět, že lineární vztah nemusí být nutně nejlepším způsob, jak závislost popsat. Nemá ale cenu měnit podobu členů například na polynomy druhého stupně, protože takovýmto způsobem půjde těžko popsat závislost, ve které například s teplotou roste koncentrace do určité limity, a pak se již výrazně nemění. Rádi bychom také zachovali podobu modelu, ve kterém jsou vlivy obou prediktorů popsány samostatnými členy, které se spolu sčítají, ale tvar závislosti ve vztahu k jednomu z prediktorů se nemění s hodnotou prediktoru druhého (tzv. aditivita vlivů).

Při hledání takového modelu můžeme začít od **parciálních residuálů**. Máme-li regresní model se dvěma prediktory (x_1 a x_2) a fitovaný model vypadá takto:

$y = b_0 + b_1*x_1 + b_2*x_2+e$

pak jsou parciální residuály ve vztahu k x_1 definovány jako:

$y – b_0 – b_2*x_2$

a parciální residuály ve vztahu k x_2 jako:

$y – b_0 – b_1*x_1$

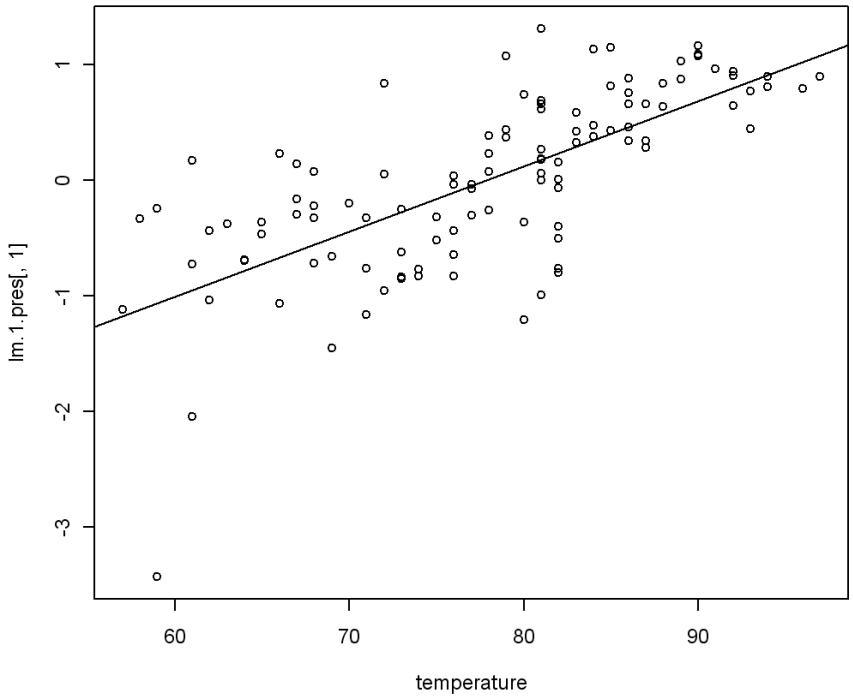
Jde tedy opět o rozdíl mezi pozorovanými a fitovanými hodnotami vysvětlované proměnné, fitované hodnoty jsou ale vypočítány s vyloučením vlivu jedné z vysvětlujících proměnných. Pokud tedy parciální residuály ve vztahu k x_1 resp. k x_2 vyneseme proti vlastním hodnotám x_1 resp. x_2 a proložíme takovým vztahem jednoduchou regresní přímku, sklon přímky bude odpovídat příslušnému koeficientu (b_1 nebo b_2) z původního modelu a zobrazený vztah nám tedy představuje roli, kterou prediktor x_1 nebo x_2 hraje v modelu sdíleném s druhým z obou prediktorů. Podívejme se, jak to vypadá například s rolí proměnné *temperature* ve výše uvedeném modelu *glm.1*:

```
> lm.1.pres<-resid(lm.1,type="partial")
> summary(lm.1.pres)
  temperature             wind
 Min.   :-3.430e+00   Min.   :-2.353e+00
 1st Qu.:-4.825e-01   1st Qu.:-3.115e-01
 Median : 5.130e-02   Median : 5.945e-02
 Mean   :-1.288e-16   Mean   :-1.574e-17
 3rd Qu.: 6.441e-01   3rd Qu.: 3.971e-01
 Max.   : 1.316e+00   Max.   : 1.527e+00
> plot(lm.1.pres[,1]~temperature,data=environmental)
> abline(lm(lm.1.pres[,1]~temperature,data=environmental))
> coef(lm(lm.1.pres[,1]~temperature,data=environmental))
(Intercept) temperature
-4.39808292  0.05653587
```

Koeficient závislosti *ozone* na *temperature* je 0.0563587, stejně jako v modelu mnohonásobné regrese *lm.1* výše. Současně také platí, že závislost na *temperature* vypadá jinak, pokud ignorujeme existenci druhého z prediktorů:

```
> coef(lm(ozone~temperature,data=environmental))
(Intercept) temperature
-1.84851790  0.06767266
```

Výše vytvořený diagram vypadá takto:

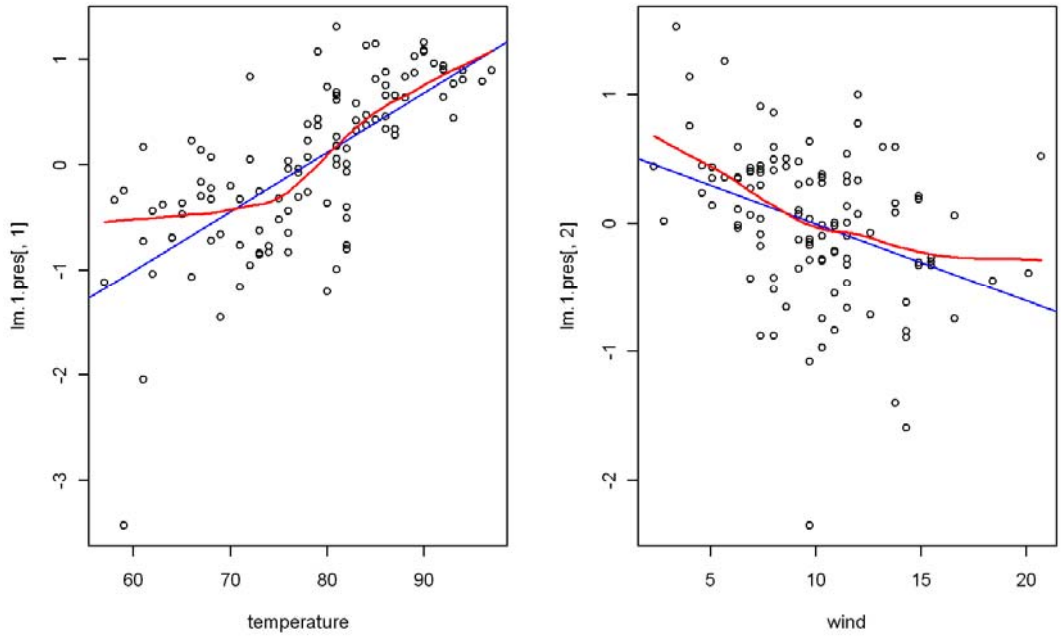


Obr. 33

Opět vidíme, že lineární vztah není - zejména na pravé straně diagramu - příliš dobrým popisem, a možná nás napadne nahradit jej obecnějším modelem, jakým je například loess model, se kterým jsme se seznámili v předchozí kapitole.

Hladké neparametrické členy

Můžeme jej do vztahu mezi parciálními residuály pro *temperature* a proměnnou *temperature* nafitovat a obdobně můžeme postupovat i v případě druhé vysvětlující proměnné – *wind*. Výsledek pak vypadá takto (loess modely jsou znázorněny červenými křivkami, původní lineární závislosti modrými přímkami):



Obr. 34

Příkazy, které tento obrázek vytvořily, vypadají takto:

```
> par(mfrow=c(1,2))
> plot(lm.1.pres[,1]~temperature,data=environmental)
> abline(lm(lm.1.pres[,1]~temperature,data=environmental),col="blue")
> lines(loess.smooth(environmental$temperature,lm.1.pres[,1]),col="red",lwd=2)

> plot(lm.1.pres[,2]~wind,data=environmental)
> abline(lm(lm.1.pres[,2]~wind,data=environmental),col="blue")
> lines(loess.smooth(environmental$wind,lm.1.pres[,2]),col="red",lwd=2)
> par(mfrow=c(1,1))
```

Dílčí vztah mezi ozónem a teplotou je charakterizován červenou křivkou ne jako lineární (tj. b_1**temperature*), ale jako hladká závislost f_1*(temperature)*. Podobně je i vztah ozónu k rychlosti větru popsán hladkou křivkou (smooth curve) f_2*(wind*). Náš postup má ale jeden problém: parciální residuály byly definovány odečtením lineárního vlivu druhého z prediktorů, ale my je teď používáme, abychom popsali nelineární vliv obou prediktorů. Měli bychom tedy výpočet parciálních residuálů zopakovat, přičemž bychom je tentokrát vypočetli s použitím předběžně nafitovaných křivek f_1 a f_2. Tím by se odhady těchto křivek trochu změnily, nicméně po několika cyklech zpřesňování odhadů parciálních residuálů a následně i hladkých křivek bychom dospěli ke konečnému řešení.

Tento postup se doopravdy používá při odhadu tzv. aditivních modelů (additive models), a to pod označením *backfitting*. Výsledkem fitování aditivního modelu je pak hladká křivka pro každý (kvantitativní) prediktor, a tato křivka graficky popisuje, jakým způsobem daný prediktor ovlivňuje hodnoty vysvětlované proměnné v modelu, ve kterém je kombinován vliv několika prediktorů (vysvětlujících proměnných).

Fitování GAM

Aditivní modely můžeme zobecnit podobně jako klasické lineární modely, pokud si definujeme link funkce převádějící škálu prediktorů (kombinovaných pomocí hladkých funkcí f_i do tzv. **aditivního prediktoru**) a také různé distribuce, ze kterých může pocházet nevysvětlená variabilita v hodnotách vysvětlované proměnné. Výsledek se nazývá zobecněné aditivní modely (*generalized additive models*, GAM). Protože se pro konkrétní prediktory (některé, všechny nebo žádné) můžeme rozhodnout, že nejlepším způsobem popisu hladké funkce $f_i(x_i)$ je lineární vztah (tj. $f_i(x_i)=\beta_i*x_i$), můžeme zobecněné aditivní modely považovat i za rozšíření zobecněných lineárních modelů, které jsou jejich "speciálním případem" (tj. když mají všechny hladké členy lineární podobu).

Podívejme se nyní na to, jakým způsobem nafitujeme nelineární vztah mezi koncentrací ozónu a proměnnými *temperature* a *wind* pomocí zobecněného aditivního modelu:

```
> library(gam)
Loading required package: splines
> gam.1<-gam(ozone~lo(temperature,span=1)+lo(wind,span=1),data=environmental)
> summary(gam.1)
Call: gam(formula = ozone ~ lo(temperature, span = 1) + lo(wind, span = 1),
    data = environmental)
Deviance Residuals:
     Min       1Q   Median       3Q      Max
-2.40388 -0.24438  0.05827  0.34454  1.12196

(Dispersion Parameter for gaussian family taken to be 0.2964)
    Null Deviance: 82.47 on 110 degrees of freedom
Residual Deviance: 31.725 on 107.0211 degrees of freedom
AIC: 185.943

Number of Local Scoring Iterations: 2

DF for Terms and F-values for Nonparametric Effects

                           Df Npar Df Npar F   Pr(F)
(Intercept)               1.0
lo(temperature, span = 1) 1.0     0.5 1.3596 0.21031
lo(wind, span = 1)        1.0     0.5 6.6287 0.02902 *
---
```

Hlavní odlišností proti funkci *glm* je na pravé straně vzorce modelu, kde u prediktorů, u kterých chceme vliv na vysvětlovanou proměnnou popsat hladkou funkcí, použijeme buď funkci *lo* (pro vyhlazování pomocí metody loess) nebo funkci *s* (pro vyhlazování metodou kubického spline-u). Při volání funkce *gam* zadáváme obvykle také parametr *family* se stejnými možnostmi jako u funkce *glm*, protože jsme ale v našem příkladě vycházeli z klasického lineárního modelu pro logaritmovanou koncentraci ozónu, nemuseli jsme parametr *family* pro předpokládanou Gaussovu distribuci zadávat. Přesto by ale bylo lepším (i když ne nutně výrazně odlišným) řešením hodnoty koncentrace nelogaritmovat a při použití funkce *gam* zadat *family=Gamma(link=log)*. Volba parametru *span* (který odpovídá hodnotě α pro loess smoother užívaný během backfitting procedury k definici hladkých členů *lo*) zajistila velmi hladké, jednoduché křivky, z nichž každá má složitost odpovídající zhruba 1.5 stupně volnosti (viz pokles residuálního počtu DF ze 110 na (zaokrouhleně) 107 DF, který zobrazuje funkce *summary*).

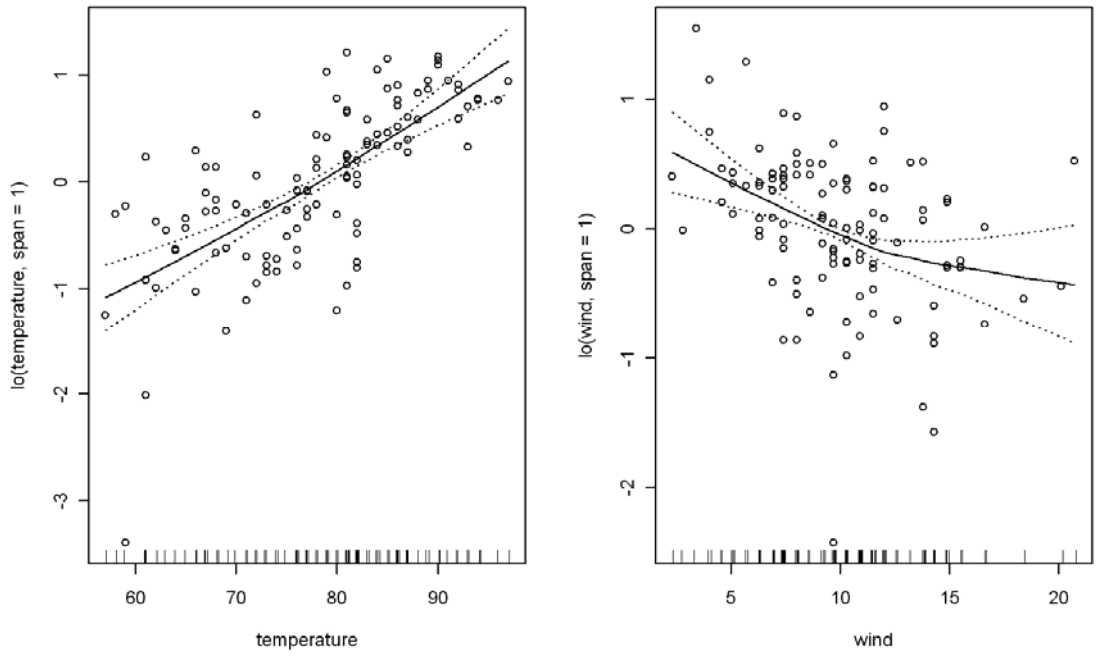
Ve výstupu funkce *summary* je zajímavá i závěrečná část, ve které je tabulka, která na prvý pohled vypadá jako test signifikance jednotlivých členů modelu. Nicméně testy v této tabulce se vztahují k jiné, konkrétnější hypotéze: složitost každého ze členů (dvou v našem příkladě) je rozdělena na lineární část (které vždy odpovídá jeden stupeň volnosti) a nelineární zbytek (zbylá část "složitosti" členu, vyjádřená pomocí stupňů volnosti). Předmětem testu je právě nelineární část a můžeme tedy (s určitou nepřesností) říci, že testujeme signifikanci nelineárnosti daného členu. Závěr v našem konkrétním případě by tedy byl, že zatímco popsaný vztah k proměnné *temperature* není průkazně odlišný od parametrického lineárního vztahu, závislost na rychlosti větru (*wind*) je složitější než lineární vztah. Je ale třeba dodat, že test je vždy prováděn proti lineární parametrické závislosti, například závislost v podobě parametrického polynomu druhého stupně není uvažována.

Grafické znázornění GAM

Než se rozhodneme o případném zjednodušení modelu *gam.1* (s ohledem na výsledek testu pro proměnnou *temperature*), podívejme se na jeho grafickou podobu – to je jediný uspokojivý nástroj pro popis zobecněných aditivních modelů:

```
> par(mfrow=c(1,2))
> plot(gam.1,resid=T,rug=T, se=T)
> par(mfrow=c(1,1))
```

Parametr *resid=T* ve volání funkce *plot* způsobil vynesení parciálních residuálů do obou diagramů, díky parametru *se=T* jsou vyneseny konfidenční intervaly křivek a parametr *rug=T* zase poskytuje, v podobě "strniště" na vnitřní části x-ové osy, informaci o hodnotách, které pro obě vysvětlující proměnné máme k dispozici. Výsledné dva diagramy jsou v Obr. 35.



Obr. 35

Obsah diagramů (obvykle volby, které jinak předáváme funkci *plot* jako parametry) můžeme definovat také interaktivně, pomocí jednoduchého menu, které funkce *plot* zobrazuje, pokud zvolíme parametr *ask=T*):

```
> plot(gam.1,ask=T)
Make a plot selection (or 0 to exit):

1: plot: lo(temperature, span = 1)
2: plot: lo(wind, span = 1)
3: plot all terms
4: residuals on
5: rug off
6: se on
7: scale (0)

Selection: 4

Make a plot selection (or 0 to exit):
1: plot: lo(temperature, span = 1)
2: plot: lo(wind, span = 1)
3: plot all terms
4: residuals off
5: rug off
6: se on
7: scale (0)

Selection: 6

Make a plot selection (or 0 to exit):
1: plot: lo(temperature, span = 1)
2: plot: lo(wind, span = 1)
3: plot all terms
4: residuals off
5: rug off
6: se off
7: scale (0)

Selection: 2
Plots performed:
 lo(wind, span = 1)
           3.97399

Make a plot selection (or 0 to exit):
1: plot: lo(temperature, span = 1)
2: plot: lo(wind, span = 1)
3: plot all terms
4: residuals off
5: rug off
6: se off
7: scale (0)

Selection: 0
```

Abychom pochopili, jak toto menu správně užívat, musíme si uvědomit, že text příkazů nám ukazuje, čeho příslušnou volbou dosáhneme. Nejprve jsme příkazem **4** zvolili, že se budou residuály vynášet (v dalším zobrazení menu se proto změnil text příkazu na "residuals off" – jeho opětovným zvolením bychom tedy dosáhli "vypnutí" residuálů). Pak jsme příkazem **6** zvolili zobrazení konfidenčních intervalů a nakonec jsme volbou příkazu **2** dosáhli vynesení diagramu pro člen, popisující efekt rychlosti větru. V tomto případě nám funkce *plot* také zobrazila číselnou hodnotu, odpovídající škále svislé osy,

která byla ve vytvořeném diagramu použita. Pokud bychom chtěli i pro další diagramy použít stejnou škálu, mohli bychom tuto hodnotu předvolit pro další diagramy příkazem **7**. Místo toho jsme ale práci s menu ukončili příkazem **0**.

Složitější diagramy můžeme vytvářet za použití funkcí *residuals*, *fitted*, *predict*, podobně jako u jiných typů regresních modelů.

Složitost hladkého členu v GAM

Při fitování modelu *gam.1* jsme použili loess smoother, u kterého se složitost členu zadává kombinací parametrů *span* a *degree*. Druhý z nich odpovídá parametru λ loess modelu a svoji hodnotou – 1 nebo 2 – tedy určuje, zda se při fitování používá lokální vážená přímka nebo lokální vážený polynom druhého stupně. Obvyklejší je ale u zobecněných aditivních modelů použití jiné hladké funkce - kubického splinu ["splajnu"]. Jeho užití volíme nahrazením *lo* písmenem *s* ve vzorci modelu a složitost se zadává přímo stupni volnosti – parametrem *df*. Jeho hodnotou nemusí být celé číslo. Následující příkaz nafituje pro naše data GAM se (skoro) stejnou složitostí obou členů modelu, ale s užitím kubických splinů:

```
> gam.2<-gam(ozone~s(temperature,df=1.5)+s(wind,df=1.5),data=environmental)
```

I pro tento model dává test neparametrické (nelineární) části členů, který zobrazuje funkce *summary*, podobný závěr (výstup vynechán) jako při užití loess křivek, tedy že vliv teploty může být popsán lineárně. Model proto přefitujeme takto:

```
> gam.3<-update(gam.2,.~temperature+s(wind,df=1.5))
> anova(gam.2,gam.3,test="F")
Analysis of Deviance Table

Model 1: ozone ~ s(temperature, df = 1.5) + s(wind, df = 1.5)
Model 2: ozone ~ temperature + s(wind, df = 1.5)
  Resid. Df Resid. Dev        Df Deviance      F Pr(>F)
1 106.99999     31.658
2 107.50000     31.905  -0.50001   -0.247 1.6673 0.1812
```

Funkce *anova* nám umožnila, pomocí parciálního F-testu hodnotícího změnu deviance, testovat rozdíl mezi původním modelem *gam.2* a modelem *gam.3*, ve kterém vystupuje teplota v lineární podobě. Mezi oběma modely není průkazný rozdíl, proto dáme přednost jednoduššímu *gam.3*. Může nás napadnout zkusit pokračovat v dalším zjednodušování modelu, tedy:

```
> gam.4<-update(gam.2,.~temperature+wind)
```

nicméně následný test ukazuje, že odstranění nelineární složky v efektu rychlosti větru kvalitu modelu zhorší:

```
> anova(gam.3,gam.4,test="F")
Analysis of Deviance Table

Model 1: ozone ~ temperature + s(wind, df = 1.5)
Model 2: ozone ~ temperature + wind
  Resid. Df Resid. Dev    Df Deviance      F  Pr(>F)
1     107.5     31.905
2     108.0     32.945  -0.5   -1.040 7.0086 0.02515 *
```

Naopak může být zajímavé se podívat, jak si vede model, ve kterém umožníme pružnější (složitější) vliv rychlosti větru na koncentraci ozónu, například takto:

```
> gam.5<-update(gam.3, .~temperature+s(wind,2))
> anova(gam.3,gam.5,test="F")
Analysis of Deviance Table

Model 1: ozone ~ temperature + s(wind, df = 1.5)
Model 2: ozone ~ temperature + s(wind, 2)
  Resid. Df Resid. Dev        Df Deviance      F  Pr(>F)
1 107.50000     31.905
2 106.99998     31.369   0.50002    0.535 3.6517 0.07916 .
```

Změna v residuální devianci sice není obrovská, ale vzhledem ke zvýšení složitosti jen o půl stupně volnosti je téměř průkazná.

To nám ukazuje obecnou slabinu zobecněných aditivních modelů. K již velké pružnosti (zobecněných) lineárních modelů přibývá nutnost další volby, vyplývající z plynulé změny složitosti (a tím i kvality) hladkých členů pro každou z vysvětlujících proměnných. Tato pružnost na jedné straně umožňuje vybrat co nejlepší popis vztahů mezi proměnnými, na druhé straně je dosti obtížné, v případě většího počtu prediktorů skoro nemožné, najít optimální kombinaci prediktorů – které použít a s jakou složitostí popisu jejich vlivu na vysvětlovanou proměnnou. Pomoc s tímto problémem nabízí automatizovaný nástroj pro výběr modelu – funkce *step* modifikována pro zobecněné aditivní modely.

Funkce *step.gam*

Tato funkce představuje vítané rozšíření funkce *step*, kterou jsme poznali u klasických lineárních modelů a u GLM. Snaží se změnit zadaný výchozí model, a pro každý z potenciálních prediktorů se rozhoduje – v uživatelem zadaném sledu - o složitosti jeho vlivu, přičemž úplná absence prediktoru je obvykle užívanou krajní možností v každém takovém sledu. Tento abstraktní popis snad čtenář pochopí lépe z praktického příkladu:

```
> scope.1<-list(temperature=~1+temperature+s(temperature,1.5)+s(temperature,2)+
+ s(temperature,4), wind=~1+wind+s(wind,1.5)+s(wind,2)+s(wind,4))
> gam.0<-gam(ozone~+1,data=environmental)
> gam.final<-step.gam(gam.0,scope=scope.1)
Start:  ozone ~ +1; AIC= 286.0267
Trial:  ozone ~  temperature + 1; AIC= 198.2116
Trial:  ozone ~  1 + wind; AIC= 246.9945
Step :  ozone ~ temperature ; AIC= 198.2116

Trial:  ozone ~  s(temperature, 1.5) + 1; AIC= 197.5533
Trial:  ozone ~  temperature + wind; AIC= 188.1721
Step :  ozone ~ temperature + wind ; AIC= 188.1721

Trial:  ozone ~  s(temperature, 1.5) + wind; AIC= 188.0268
Trial:  ozone ~  temperature + s(wind, 1.5); AIC= 185.6114
Step :  ozone ~ temperature + s(wind, 1.5) ; AIC= 185.6114

Trial:  ozone ~  1 + s(wind, 1.5); AIC= 241.9764
Trial:  ozone ~  s(temperature, 1.5) + s(wind, 1.5); AIC= 185.7499
Trial:  ozone ~  temperature + s(wind, 2); AIC= 184.7333
Step :  ozone ~ temperature + s(wind, 2) ; AIC= 184.7333

Trial:  ozone ~  1 + s(wind, 2); AIC= 239.8881
Trial:  ozone ~  s(temperature, 1.5) + s(wind, 2); AIC= 184.995
Trial:  ozone ~  temperature + s(wind, 4); AIC= 185.4202
```

V prvém příkazu jsme vytvořili proměnnou, která nám pro každý potenciálně použitelný prediktor definuje rozsah možností (regimen), v jaké podobě se může prediktor v modelu použít. Tuto proměnnou (seznam, který má tolik položek, kolik máme potenciálních prediktorů) užíváme ve funkci *step.gam* jako hodnotu pro parametr *scope*.

Výběr modelu začíná od nulového (*gam.0*), ve kterém není žádný prediktor použit. Funkce *step.gam* se snaží pro každý prediktor změnit podobu jeho zastoupení v modelu, a to výběrem jednoho z členů jemu odpovídajícího vzorce definovaného v parametru *scope*. Pravidlem je, že se funkce může v rámci jednoho kroku pokoušet změnit složitost každého z prediktorů pouze posunem k následujícímu nebo předchozímu výrazu vzorce. To například pro první krok znamená, že funkce *step.gam* zkouší přidat lineární podobu pro *temperature* nebo *wind*. Vybrán byl lineární člen pro *temperature*, protože ten vedl k největšímu poklesu hodnoty AIC. V dalším kroku zkouší *step.gam* buď "zesložitit" závislost na *temperature* nebo přidat lineární závislost na proměnné *wind*, a tato druhá změna je nakonec vybrána. V následujícím kroku zkouší *step.gam* opět složitější (DF 1.5) výraz pro *temperature* nebo složitější výraz (s DF 1.5) pro *wind*, a vybrána je druhá možnost. V tomto okamžiku tedy model vypadá takto: *ozone~temperature+s(wind,1.5)*. V dalším kroku pak zkouší *step.gam* odebrat úplně proměnnou *temperature* (posun v jejím vzorci doleva), zvýšit složitost efektu této proměnné (posun v jejím vzorci doprava), resp. zvýšit složitost efektu proměnné *wind* (posun v jejím vzorci doprava – posun doleva nemá smysl, protože tím by se funkce jenom vrátila o krok zpátky). Nejlepší variantou je – podle hodnoty AIC - zvýšení složitosti efektu *wind*. Funkce *step.gam* zkouší ještě další krok, nicméně všechny varianty mají AIC vyšší než výsledek předchozího kroku a funkce se proto zastaví.

Výsledný model je uložen do objektu *gam.final*, se kterým můžeme pracovat jako se standardním objektem vraceným funkcí *gam*. Nicméně, obsahuje také další informace – především položku *anova*, která stručně popisuje sekvenci výběru:

```
> gam.final$anova
Stepwise Model Path
Analysis of Deviance Table

Initial Model:
ozone ~ +1

Final Model:
ozone ~ temperature + s(wind, 2)

Scale: 0.749727

     From      To        Df Deviance Resid. Df Resid. Dev     AIC
1   1.000   1.000                    110.00000     82.470 286.027
2   1.000   4.000 -1.00000  -45.751 109.00000     36.719 198.212
3   1.000   5.000 -1.00000   -3.774 108.00000     32.945 188.172
4   3.000   2.000 -0.50000   -1.040 107.50000     31.905 185.611
5   2.000   3.000 -0.50002   -0.535 106.99998     31.369 184.733
```

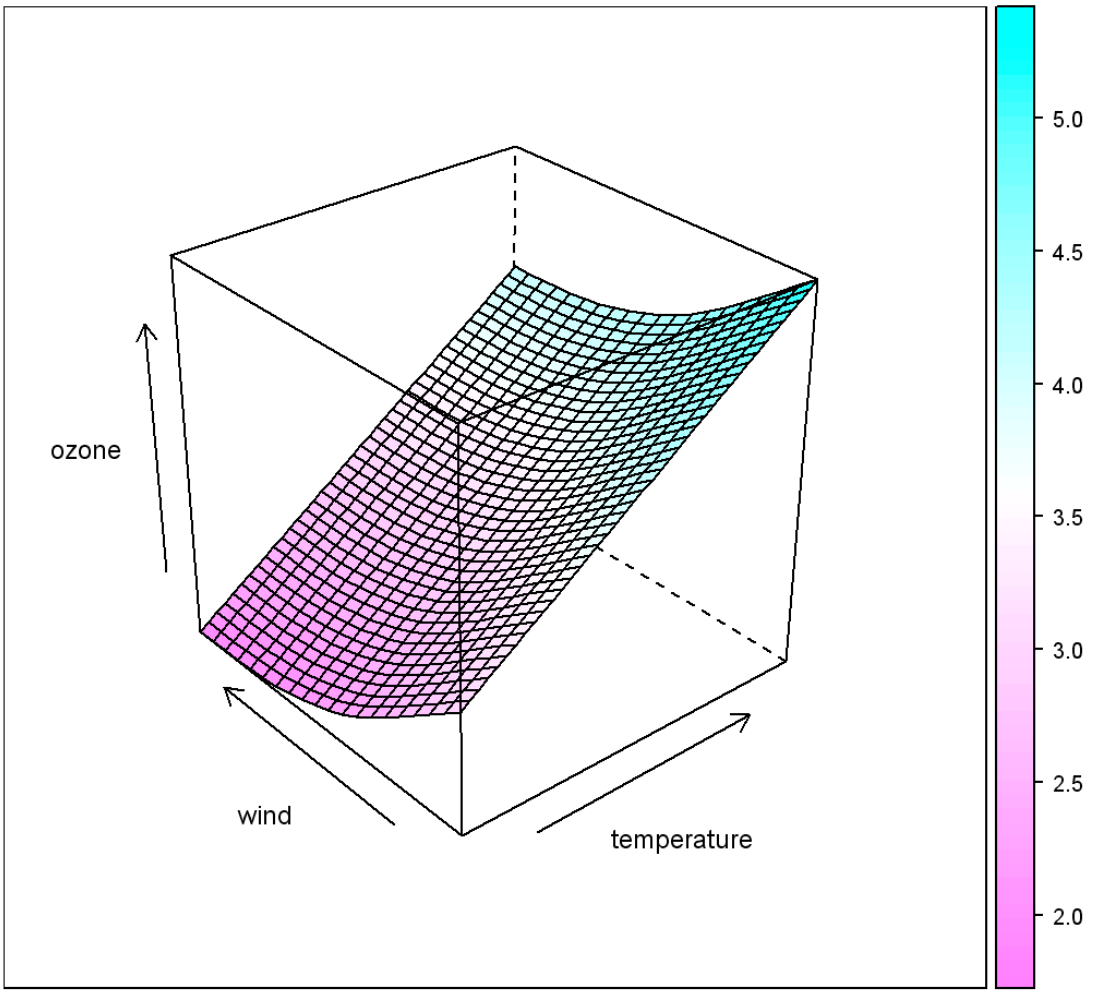
Zobrazení odezvového povrchu GAM

Náš výsledný model můžeme vynést také jako odezvový povrch, postupem obdobným tomu, který jsme si ukázali již na konci druhé kapitoly.

```
> lapply(environmental[,3:4], range)
```

```
$temperature
[1] 57 97

$wind
[1]  2.3 20.7
> env.pred<-list(temperature=seq(57,97,by=1),wind=seq(2,21,by=1))
> fit.1<-predict(gam.final,expand.grid(env.pred),type="response")
> wireframe(fit.1~temperature*wind,data=expand.grid(env.pred),drape=T,
+   zlab="ozone")
```

Výsledný perspektivní pohled na odezvových povrch je v následujícím Obr. 36.



Obr. 36

6 Analýza přežívání

Regresními modely analýzy přežívání (survival analysis) se snažíme popsat délku života jedinců nebo dobu trvání určité události. Tyto délky nebo doby se obvykle snažíme vysvětlit buď příslušností jedince či události k určité skupině (tj. hodnotou nějakého faktoru) nebo i kvantitativními vysvětlujícími proměnnými.

Motivační příklad

```
> seedlings<-read.delim("clipboard")
> summary(seedlings)
 species  place         day             died
 AM:10   les :15   Min.   : 7.00   Min.   :0.0000
 HL:10   pole:15   1st Qu.:37.75   1st Qu.:0.0000
 PL:10             Median :63.00   Median :1.0000
                   Mean   :52.30   Mean   :0.5667
                   3rd Qu.:70.00   3rd Qu.:1.0000
                   Max.   :70.00   Max.   :1.0000
```

Data pocházejí z pilotní studie, ve které bylo zjišťováno, jak dlouho přežívají semenáčky lučních rostlin, vyklíčené ve skleníku a vysazené do lučního porostu, a také jak se toto přežívání liší v různých místech louky (místo označené *"les"* bylo výrazně sušší, s nižší biomasou - a tím i zástinem - okolní vegetace, místo *"pole"* bylo ve vyšší luční vegetaci, s větší půdní vlhkostí i obsahem živin). Semenáčky tří druhů rostlin (řebříček *AM*, medyněk *HL* a jitrocel *PL*) byly vysazeny na konci března na každé z obou míst vždy v počtu pěti jedinců a jejich osud byl sledován při pravidelných týdenních návštěvách lokality. Pokud semenáček nebyl při kontrole nalezen nebo byl mrtvý, zaznamenal se čas (počet dní od začátku pokusu), ve kterém ke smrti či zmizení došlo (ten mohl být ve skutečnosti kratší – přesnost je určena četností návštěv). Tyto semenáčky mají v proměnné *died* hodnotu 1. Pokus byl ukončen po deseti týdnech a všechny semenáčky, které byly k tomuto datu živé, mají v proměnné *day* hodnotu 70 a v proměnné *died* hodnotu 0. V tomto druhém případě jde o tzv. **cenzorovaná** (censored) **pozorování** a takové případy se vyskytují v analýze přežívání velmi často. V našem případě šlo o nejtypičtější (a z praktického hlediska analýzy nejpřijatelnější) druh cenzorování, tzv. **cenzorování zprava** (right censoring). Pokud bychom semenáčky nevysazovali ve známý den, ale sledovali bychom v přírodě vyklíčené jedince, u kterých přesné stáří ve dnech neznáme, měli bychom v případě, že zaznamenáme alespoň datum jejich smrti, tzv. **cenzorování zleva** (left censoring). Třetí možnost - intervalové cenzorování - většina známých metod analýzy přežívání nepodporuje.

Funkce přežívání

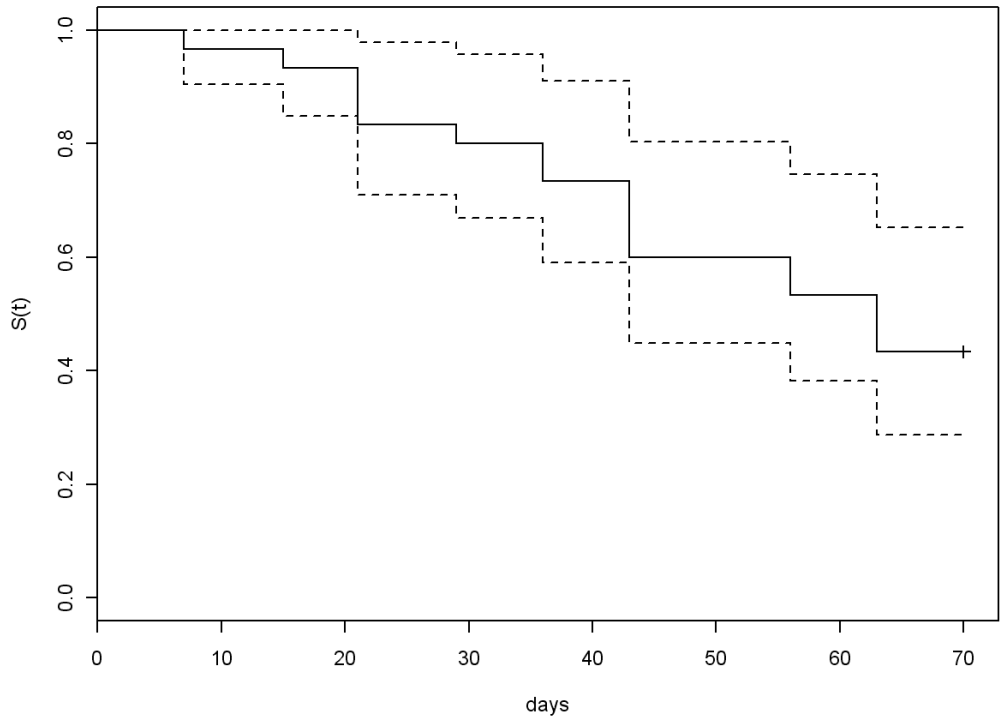
Začněme nejprve grafickým shrnutím našich dat. Vzhledem k tomu, že nás zajímá délka života jedinců, měli bychom se nejspíše soustředit na distribuci hodnot těchto délek, nicméně místo této distribuční funkce se obvykle používá jiný typ znázornění, jak brzo uvidíme. Chceme-li vlastnosti dat shrnout, musíme nejprve nafitovat jednoduchý model

přežívání. Funkce *survfit* je, podobně jako ostatní metody analýzy přežívání, dostupná v knihovně *survival*:

```
> library(survival)
> sf.1<-survfit(Surv(day,died)~+1,data=seedlings)
```

Pomocí následujícího kódu si vytvoříme nejčastěji používaný typ grafu, výsledek je v Obr. 37.

```
> plot(sf.1,xlab="days",ylab="S(t)")
```



Obr. 37

Schodovitě sestupující křivka znázorněná plnou čárou představuje odhad tzv. **funkce přežívání** (survival function, někdy též survivor function). Tato funkce nám pro danou dobu života semenáčku v porostu (kterou lze odečíst na horizontální ose) ukazuje (na svislé ose) pravděpodobnost, že se semenáček daného věku dožije. Odhady hodnoty funkce pro různé časy, ve kterých byla pozorována změna v počtu žijících semenáčků (tj. pro časy kontrol), jsou spočteny tzv. Kaplan-Meierovou metodou a lze k nim také vypočítat jejich standardní chybu. Jakým způsobem jsou hodnoty fukce přežívání z našich dat spočteny? Odpověď nám dá funkce *summary*, aplikovaná na objekt *sf.1*:

```
> summary(sf.1)
```

```
 time n.risk n.event survival std.err lower 95% CI upper 95% CI
    7     30       1    0.967  0.0328        0.905        1.000
   15     29       1    0.933  0.0455        0.848        1.000
   21     28       3    0.833  0.0680        0.710        0.978
   29     25       1    0.800  0.0730        0.669        0.957
   36     24       2    0.733  0.0807        0.591        0.910
   43     22       4    0.600  0.0894        0.448        0.804
   56     18       2    0.533  0.0911        0.382        0.745
   63     16       3    0.433  0.0905        0.288        0.652
```

Obecný vzoreček pro výpočet *S(t)* v čase *t* je tento:

$$\hat{S}_{KM}(t) = \prod_{t_i<t} \frac{r(t_i) - d(t_i)}{r(t_i)}$$

kde $r(t_i)$ je počet jedinců, kteří stále přežívají v čase t_i (tj. jsou vystaveni **r**iziku), $d(t_i)$ je počet úmrtí (**d**eaths), která byla pozorována od předchozího uvažovaného času do času t_i. Velké písmeno Π označuje součin. První úmrtí – popsané v prvém řádku tabulky funkce *summary* (jde o úmrtí jednoho semenáčku, viz hodnota ve sloupci *n.event*) nastalo mezi dnem 0 a 7, předchozí časy nejsou, takže odhad *S(t=7)* je *(30-1)/30=0.967*. V čase 15 dní bylo zaznamenáno další úmrtí, proto dosavadní hodnotu funkce *S* (0.967) násobíme poměrem *(29-1)/29=0.9655*, výsledek *S(t=15)* je tedy *0.967*0.9655=0.933*, atd. Standardní chyba těchto odhadů je funkcí *survfit* obvykle[29] počítána tzv. Greenwood-ovou metodou.

Pokud bychom popsali distribuci hodnot délek života *t* pomocí kumulativní distribuční funkce (kterou můžeme označit *F(t)*), vyjadřovala by tato funkce, pro danou hodnotu délky života *t*, pravděpodobnost, že se náhodně vybraný jedinec dožije takovéhoto nebo nižšího věku (tj. že zemře nejpozději v čase *t*). Funkce *F(t)* tedy – na rozdíl od funkce přežívání – **roste** od nuly do jedničky zleva doprava a mezi těmito dvěma funkcemi je jednoduchý vztah:

$$S(t) = 1 - F(t)$$

Hodnotu kumulativní distribuční funkce vypočteme (obecně, ne jen v případě časových délek) integrací tzv. densitní distribuční funkce, kterou označíme jako *f(t)* a která vyjadřuje pravděpodobnost, že se náhodně zvolený semenáček dožije právě daného stáří. Kombinací *f(t)* a *S(t)* lze definovat jinou veličinu, která se v analýze přežívání často používá, a to míru rizika:

$$\lambda(t) = \frac{f(t)}{S(t)}$$

Míra rizika (hazard rate), $\lambda(t)$, je definována jako pravděpodobnost, že jedinec zemře v čase *t*, pokud se mu již podařilo času *t* dožít. Je-li tato definice pro čtenáře příliš složitá, doporučuji $\lambda(t)$ interpretovat jako okamžité riziko úmrtí v daném věku. Pokud bychom sledovali rostliny (či živočichy) během jejich celého životního cyklu, můžeme předpokládat buď růst tohoto rizika u starších jedinců nebo nižší hodnoty ve středním věku a s vyšším rizikem v mládí a/nebo ve stáří). V případě našich semenáčků můžeme očekávat buď konstantní míru rizika nebo její zvýšenou hodnotu $\lambda(t)$ v počátečních dnech či týdnech, než se semenáček po přesazení "uchytí". I pro míru rizika lze definovat kumulativní funkci (cumulative hazard rate), která se označuje velkým písmenem lambda a je definována takto:

$$\Lambda(t) = \int_0^t \lambda(t)dt = -\log S(t)$$

[29] Pokud funkci *survfit* použijeme pro fitovaný Coxův model (viz dále v této kapitole), je k odhadu chyb používána Tsiatis-ova metoda.

Druhý způsob definice Λ, tj. jako záporný logaritmus funkce přežívání, je významný ze dvou důvodů. Prvním je, že vede k myšlence vynášení hodnot funkce *S(t)* na logaritmické škále. Takto škálované funkce přežívání pak lze porovnávat se třemi teoretickými typy, známými z učebnic populační ekologie: Typ I se zvýšenou mortalitou v pozdních fázích života, Typ III se zvýšenou mortalitou v juvenilních fázích, a Typ II s konstatní mortalitou, který je v grafu s logaritmovanou funkcí přežívání znázorněn klesající přímkou. Pokud chceme vynést graf s logaritmovanou funkcí přežívání, provedeme to takto (výsledný diagram nezobrazen):

```
> plot(sf.1,xlab="days",ylab="log Survival curve\nlog(S(t))", log=T)
```

Druhým důvodem významnosti kumulativní míry rizika je to, že přímý odhad Λ umožňuje alternativní způsob odhadu funkce přežívání, a to tzv. Fleming-Harringtonovou metodou. V ní se nejprve nezávisle odhadne kumulativní míra rizika (tzv. Nelsonův odhad):

$$\hat{\Lambda}_N(t) = \sum_{t_i<t} \frac{d(t_i)}{r(t_i)}$$

kde smysl $d(t_i)$ a $r(t_i)$ je stejný jako u Kaplan-Meierova odhadu *S(t)*. Funkce přežívání je pak vypočtena takto:

$$\hat{S}_{FH}(t) = e^{-\Lambda_N(t)}$$

Odhady funkce *S(t)* založené na Kaplan-Meierovu resp. na Fleming-Harringtonovu postupu jsou obecně shodné, liší se ale v případě výskytu shodných délek života (dob trvání jevů) u více případů (což nastává i u našich semenáčků, kde například v čase 21 dní odumřeli tři jedinci). Odlišnosti odhadů můžeme zjistit porovnáním výše uvedeného výstupu z funkce *survfit* s tímto:

```
> sf.2<-survfit(Surv(day,died)~+1,type="flem",data=seedlings)
> summary(sf.2)
Call: survfit(formula = Surv(day, died) ~ +1, data = seedlings, type = "flem")

 time n.risk n.event survival std.err lower 95% CI upper 95% CI
    7     30       1    0.967  0.0328        0.905        1.000
   15     29       1    0.934  0.0456        0.849        1.000
   21     28       3    0.839  0.0685        0.715        0.985
   29     25       1    0.807  0.0736        0.674        0.965
   36     24       2    0.742  0.0817        0.598        0.921
   43     22       4    0.619  0.0922        0.462        0.829
   56     18       2    0.554  0.0946        0.396        0.774
   63     16       3    0.459  0.0958        0.305        0.691
```

Rozdíly v přežívání mezi skupinami

V dosavadní diskusi jsme ignorovali skutečnost, že naše semenáčky patří k různým druhům a také byly vysazeny do různě riskantních podmínek. Z celkové křivky přežívání můžeme tedy usoudit, že pokud poběží náš pokus např. 5 týdnů (a experimentální manipulace nezmění míru rizika), bude na konci pokusu zhruba polovina vysazených semenáčků živá (konfidenční interval pro pravděpodobnost přežití do 36. dne je 0.591-0.910 podle Kaplan-Meirova/Greenwoodova odhadu). Nicméně, pokud by se křivka přežívání lišila mezi třemi srovnávanými druhy, měli bychom pro více citlivý druh

vysadit více než jen dvojnásobek žádaného počtu opakování, která nám mají přežít do konce pokusu. Jak rozdíly v přežívání mezi druhy (nebo mezi dvěma stanovišti pokusné louky) otestovat? Pro porovnání křivek přežívání mezi dvěma nebo více skupinami slouží funkce *survdiff*, kterou lze použít takto:

```
> survdiff(Surv(day,died)~species,data=seedlings)
Call: survdiff(formula = Surv(day, died) ~ species, data = seedlings)

            N Observed Expected (O-E)^2/E (O-E)^2/V
species=AM 10        6     5.83   0.00467   0.00776
species=HL 10        5     6.34   0.28331   0.49334
species=PL 10        6     4.82   0.28626   0.43699

 Chisq= 0.6  on 2 degrees of freedom, p= 0.731
```

Pozorované počty událostí (úmrtí semenáčků) jsou pomocí χ^2 testu porovnávány s očekávaným počtem, který předpovídá funkce přežívání fitovaná podle nulové hypotézy – tj. pro všechny tři skupiny společně. Funkce *survdiff* ale nabízí více testů, konkrétní typ je určen zvolenou hodnotou parametru *rho*. Implicitní hodnota *rho=0*, použitá v našem výše uvedeném příkladu, vede k tzv. Mantel-Haenszelovu (nebo též log-rank) testu. Pokud chceme v testu zdůraznit rozdíly v počátečních fázích křivky přežívání (tj. pro krátké časy života), lze použít tzv. Peto test, a to s použitím hodnoty *rho=1*:

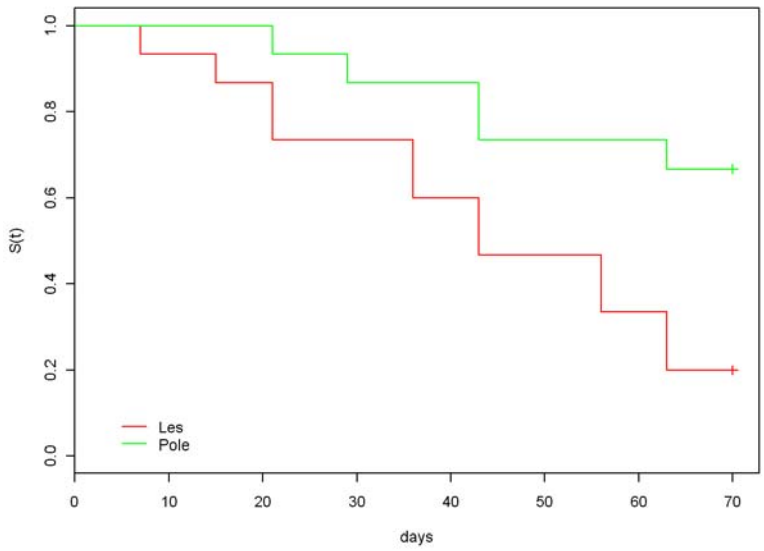
```
> survdiff(Surv(day,died)~species,data=seedlings,rho=1)
Call:
survdiff(formula = Surv(day, died) ~ species, data = seedlings,
    rho = 1)

            N Observed Expected (O-E)^2/E (O-E)^2/V
species=AM 10     4.37     4.47   0.00224   0.00464
species=HL 10     3.47     4.80   0.37037   0.80290
species=PL 10     5.10     3.67   0.56030   1.06292

 Chisq= 1.3  on 2 degrees of freedom, p= 0.53
```

Vidíme, že ani takto není rozdíl mezi druhy průkazný. Pokud se podíváme na rozdíly mezi dvěma stanovišti (efekt faktoru *place*), dostáváme odlišné výsledky:

```
> survdiff(Surv(day,died)~place,data=seedlings)
Call:
survdiff(formula = Surv(day, died) ~ place, data = seedlings)

             N Observed Expected (O-E)^2/E (O-E)^2/V
place=les   15       12     7.08      3.41      6.45
place=pole 15        5     9.92      2.44      6.45

 Chisq= 6.5  on 1 degrees of freedom, p= 0.0111
```

Semenáčky tedy přežívají na sušším stanovišti (*les*) výrazně méně než na stanovišti druhém (*pole*). Naše výsledky by proto bylo lepší popsat dvěma křivkami přežívání, odděleně fitovanými pro tyto dvě skupiny:

```
> sf.2<-survfit(Surv(day,died)~place,data=seedlings)
> plot(sf.2,xlab="days",ylab="S(t)",
+ col=c("red","green"),legend.text=c("Les","Pole"))
```



Obr. 38

Z Obr. 38 vidíme, že pokud vynášíme do diagramu více než jednu křivku přežívání, funkce *plot* je nedoplňuje konfidenčními regiony (v případě dvou křivek by to znamenalo celkem šest čar v obrázku). S trochu úsilí je ale možné si jejich přítomnost v grafu vynutit.

Ještě bych rád čtenáře upozornil na to, že funkce *survdiff* prezentuje výsledky oboustraného testu nulové hypotézy. Pokud ale porovnáváme například kontrolní skupinu se skupinou, u kterého by měl experimentální zásah např. snižovat míru rizika, měli bychom použít jednostraný test. Toho dosáhneme vydělením zobrazené signifikance dvěma, ovšem jen za situace, kdy je pozorovaný počet událostí (úmrtí) nižší (pro manipulovanou skupinu) než počet očekávaný.

Coxův model relativního rizika

Pokud vysvětlujeme přežívání jedinců pomocí faktorů, můžeme jejich vliv testovat porovnáním křivek přežívání, fitovaných pro jednotlivé skupiny, jak jsme si právě ukázali. Pokud ale chceme studovat vliv více faktorů najednou nebo pokud máme jednu nebo více kvantitativních vysvětlujících proměnných, s tímto přístupem již nevystačíme. Můžeme ale často použít tzv. **Coxův model relativního rizika** (Cox proportional hazard model, dále jen Coxův model). Nulový Coxův model předpokládá, že všechna pozorování lze společně popsat křivkou vyjadřující míru rizika $\lambda(t)$ pro různé doby trvání života t. Tato křivka nemá předem daný tvar, jde o neparametrický odhad a představuje tzv. referenční křivku relativního rizika (baseline hazard rate curve), λ_0. Pokud mají některé vysvětlující proměnné x_j vliv na přežívání jedinců, lze jej popsat následujícím způsobem:

$$\hat{\lambda}(x_i, t) = \lambda_o(t) e^{\eta_i}$$

kde η_i je – podobně jako u GLM – lineární prediktor, tedy lineární kombinace vysvětlujících proměnných:

$$\eta_i = \sum_j \beta_j x_{ij}$$

Vzhledem k tomu, že prediktory ovlivňují míru rizika násobně přes exponencielu, bývá zvykem vyjadřovat tyto vlivy v podobě *exp(β_j)*, a v této podobě pak koeficient vyjadřuje, **kolikrát** se změní referenční míra rizika[30] s jednotkovou změnou příslušné vysvětlující proměnné.

Takto popsaný model lze změnit přidáním parametru *strata* do vzorce modelu – tento parametr určuje rozdělení pozorování do skupin (na základě jednoho nebo více faktorů) a referenční křivka relativního rizika je pak fitována pro každou skupinu zvlášť.

Jako příkladová data použijeme údaje, které jsou součástí package *survival*, a popisující přežívání pacientů s diagnózou rakoviny plic:

```
> data(cancer)
> summary(cancer)
      inst             time           status           age       
 Min.   : 1.00   Min.   :   5.0   Min.   :1.000   Min.   :39.00  
 1st Qu.: 3.00   1st Qu.: 166.8   1st Qu.:1.000   1st Qu.:56.00  
 Median :11.00   Median : 255.5   Median :2.000   Median :63.00  
 Mean   :11.09   Mean   : 305.2   Mean   :1.724   Mean   :62.45  
 3rd Qu.:16.00   3rd Qu.: 396.5   3rd Qu.:2.000   3rd Qu.:69.00  
 Max.   :33.00   Max.   :1022.0   Max.   :2.000   Max.   :82.00  
 NA's   : 1.00                                                   
      sex           ph.ecog          ph.karno        pat.karno     
 Min.   :1.000   Min.   :0.0000   Min.   : 50.00   Min.   : 30.00  
 1st Qu.:1.000   1st Qu.:0.0000   1st Qu.: 75.00   1st Qu.: 70.00  
 Median :1.000   Median :1.0000   Median : 80.00   Median : 80.00  
 Mean   :1.395   Mean   :0.9515   Mean   : 81.94   Mean   : 79.96  
 3rd Qu.:2.000   3rd Qu.:1.0000   3rd Qu.: 90.00   3rd Qu.: 90.00  
 Max.   :2.000   Max.   :3.0000   Max.   :100.00   Max.   :100.00  
                 NA's   :1.0000   NA's   :  1.00   NA's   :  3.00  
    meal.cal         wt.loss       
 Min.   :  96.0   Min.   :-24.000  
 1st Qu.: 635.0   1st Qu.:  0.000  
 Median : 975.0   Median :  7.000  
 Mean   : 928.8   Mean   :  9.832  
 3rd Qu.:1150.0   3rd Qu.: 15.750  
 Max.   :2600.0   Max.   : 68.000  
 NA's   :  47.0   NA's   : 14.000  
```

V naší analýze se zaměříme jen na některé z možných prediktorů a jejich význam zde stručně popisuji: *time* udává délku života pacienta ve dnech, *status* udává způsob, kterým pro daného pacienta pozorování skončilo (1=cenzorováno zprava, tj. pacient nezemřel, ale nebyl již dále sledován; 2=smrt). Je vidět, že jde o alternativní kódování rozdílu mezi cenzorováním a událostmi (hodnoty 1-2 místo 0-1). Proměnná *age* udává stáří pacienta v letech, *sex* kóduje jeho pohlaví, *ph.ecog* představuje standardizovanou diagnostickou charakteristiku, založenou na různých měřeních na pacientovi. Proměnné *ph.karno* a *pat.karno* vyjadřují relativní schopnost pacienta vykonávat běžné každodenní činnosti, přičemž tato schopnost (na škále 0-100) je posuzována buď lékařem (*ph.karno*) nebo samotným pacientem (*pat.karno*).

[30] Zvětší, pro hodnotu >1, či zmenší, pro hodnotu <1

K výběru vhodného modelu můžeme přistupovat dvojím způsobem: buď můžeme nafitovat plný model a následně odstranit vysvětlující proměnné s neprůkaznými efekty, nebo můžeme začít od nulového modelu a ten eventuelně obohatit metodou postupného výběru, s použitím funkce *stepAIC*[31], která je definována v package *MASS*. Ukážeme si oba postupy. Nejprve ale musíme začít odstraněním chybějících údajů, které v datech máme:

```
> cancer.x<-na.omit(cancer)
```

Coxův model fitujeme pomocí funkce *coxph*. V případě nulového modelu (bez prediktorů) vypadá její použití takto:

```
> ph.0<-coxph(Surv(time,status)~+1,data=cancer.x)
```

Volbu modelu pomocí funkce *stepAIC* provedeme následovně:

```
> step.1<-stepAIC(ph.0,~age+sex+ph.ecog+ph.karno+pat.karno)
Start:  AIC= 1016.23
 Surv(time, status) ~ +1

            Df    AIC
+ ph.ecog    1 1005.8
+ pat.karno  1 1009.4
+ sex        1 1012.0
+ age        1 1014.7
+ ph.karno   1 1015.0
<none>         1016.2

Step:  AIC= 1005.82
 Surv(time, status) ~ ph.ecog

            Df    AIC
+ sex        1 1000.8
+ ph.karno   1 1005.7
<none>         1005.8
+ pat.karno  1 1006.5
+ age        1 1007.0
- ph.ecog    1 1016.2

Step:  AIC= 1000.75
 Surv(time, status) ~ ph.ecog + sex

            Df    AIC
+ ph.karno   1 1000.1
<none>         1000.8
+ pat.karno  1 1001.9
+ age        1 1002.2
- sex        1 1005.8
- ph.ecog    1 1012.0

Step:  AIC= 1000.07
 Surv(time, status) ~ ph.ecog + sex + ph.karno

            Df    AIC
<none>         1000.1
+ pat.karno  1 1000.7
- ph.karno   1 1000.8
+ age        1 1000.8
```

[31] funkce *step*, kterou jsme užívali dříve, zde nebude fungovat

```
- sex        1 1005.7
- ph.ecog    1 1011.0
```

Proběhlý postupný výběru lze shrnout i následovně:

```
> step.1$anova
Stepwise Model Path
Analysis of Deviance Table

Initial Model:
Surv(time, status) ~ +1

Final Model:
Surv(time, status) ~ ph.ecog + sex + ph.karno


        Step Df  Deviance Resid. Df Resid. Dev      AIC
1                              167  1016.2336 1016.234
2  + ph.ecog  1 12.408747       166  1003.8249 1005.825
3      + sex  1  7.073454       165   996.7514 1000.751
4 + ph.karno  1  2.678999       164   994.0724 1000.072
```

Vidíme, že poslední krok (přidání prediktoru *ph.karno*) nesnížil AIC nijak podstatně. Tomu odpovídá i výsledek parametrického testu pro jednotlivé prediktory:

```
> summary(step.1)
Call:
coxph(formula = Surv(time, status) ~ ph.ecog + sex + ph.karno,
    data = cancer.x)

  n= 167
            coef exp(coef) se(coef)      z       p
ph.ecog   0.7492     2.115   0.2118  3.54 0.00041
sex      -0.5314     0.588   0.1973 -2.69 0.00710
ph.karno  0.0178     1.018   0.0111  1.60 0.11000

         exp(coef) exp(-coef) lower .95 upper .95
ph.ecog      2.115      0.473     1.397     3.204
sex          0.588      1.701     0.399     0.865
ph.karno     1.018      0.982     0.996     1.040

Rsquare= 0.124   (max possible= 0.998 )
Likelihood ratio test= 22.2  on 3 df,   p=6.04e-05
Wald test            = 21.9  on 3 df,   p=6.7e-05
Score (logrank) test = 22.3  on 3 df,   p=5.72e-05
```

Nejvýznamnějším prediktorem je diagnostické skóre *ph.ecog*, jeho každé zvýšení o jednotku[32] zvýšuje míru rizika 2.115-krát. Významný je i rozdíl podle pohlaví pacienta.

Alternativní přístup k výběru modelu vypadá takto:

```
> ph.all<-update(ph.0,.~age+sex+ph.ecog+ph.karno+pat.karno)

> summary(ph.all)
Call:
coxph(formula = Surv(time, status) ~ age + sex + ph.ecog + ph.karno +
    pat.karno, data = cancer.x)

  n= 167
```

[32] hodnoty jsou v rozsahu 1-5, ale 5 znamená smrt a pacienty s hodnotou 4 již kliniky nepřijímaly

```
              coef exp(coef) se(coef)     z      p
age        0.01213     1.012  0.01147  1.06 0.2900
sex       -0.51076     0.600  0.19805 -2.58 0.0099
ph.ecog    0.65801     1.931  0.22201  2.96 0.0030
ph.karno   0.02193     1.022  0.01146  1.91 0.0560
pat.karno -0.00856     0.991  0.00788 -1.09 0.2800

          exp(coef) exp(-coef) lower .95 upper .95
age           1.012      0.988     0.990     1.035
sex           0.600      1.667     0.407     0.885
ph.ecog       1.931      0.518     1.250     2.984
ph.karno      1.022      0.978     0.999     1.045
pat.karno     0.991      1.009     0.976     1.007

Rsquare= 0.137   (max possible= 0.998 )
Likelihood ratio test= 24.6  on 5 df,   p=0.000164
Wald test            = 24.6  on 5 df,   p=0.000166
Score (logrank) test = 25.2  on 5 df,   p=0.000125
```

Závěry jsou tedy shodné s těmi, které jsme udělali při použití postupného výběru – podstatnými vysvětlujícími proměnnými jsou *ph.ecog* a *sex*. Konečný model bude tedy vypadat takto:

```
> ph.1<-update(ph.0,.~ph.ecog+sex)
> summary(ph.1)
Call:
coxph(formula = Surv(time, status) ~ ph.ecog + sex, data = cancer.x)

  n= 167
          coef exp(coef) se(coef)     z       p
ph.ecog  0.483      1.62    0.132  3.65 0.00027
sex     -0.510      0.60    0.197 -2.59 0.00960

        exp(coef) exp(-coef) lower .95 upper .95
ph.ecog      1.62      0.617     1.250     2.100
sex          0.60      1.665     0.408     0.883

Rsquare= 0.11   (max possible= 0.998 )
Likelihood ratio test= 19.5  on 2 df,   p=5.88e-05
Wald test            = 19.4  on 2 df,   p=6.3e-05
Score (logrank) test = 19.6  on 2 df,   p=5.49e-05
```

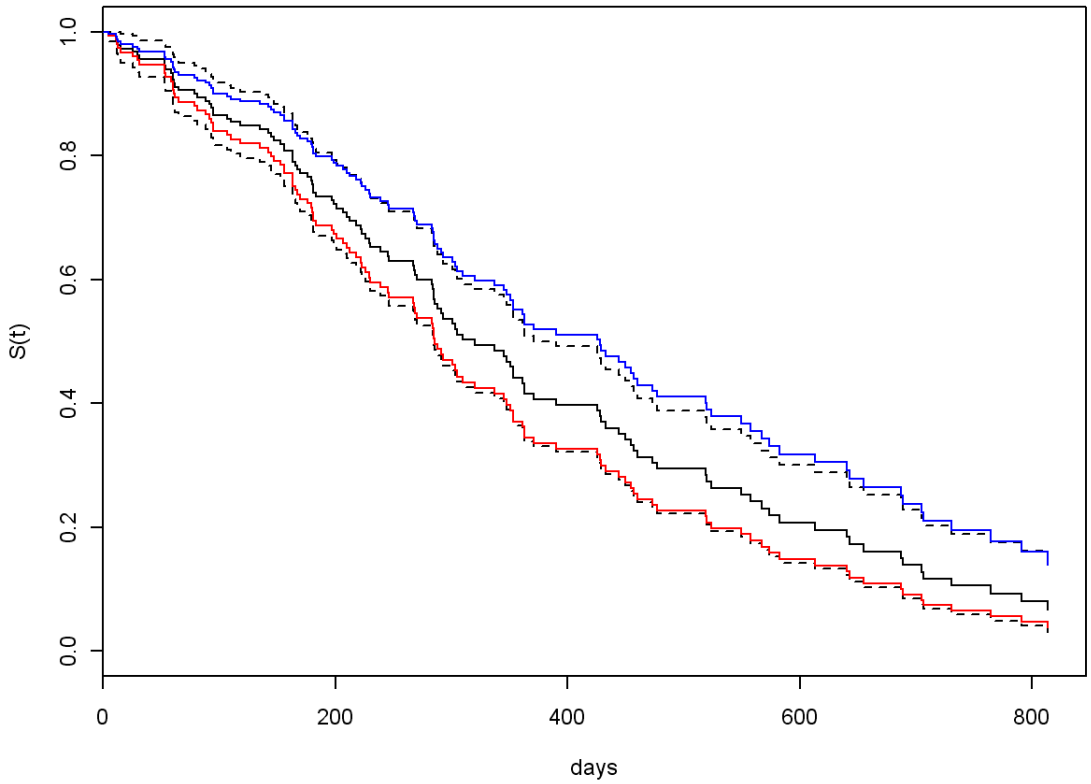
I v případě Coxova modelu je možné zobrazit nafitovanou křivku přežívání. Překvapivě k tomu použijeme opět funkci *survfit*, kterou jsme dosud používali jen k fitování Kaplan-Meierova modelu:

```
> sf.2<-survfit(ph.1)
> plot(sf.2,xlab="days",ylab="S(t)")
```

Výše uvedené použití funkce *plot* zobrazí funkci přežívání předpokládající **průměrné** hodnoty obou prediktorů (tj. *ph.ecog* a *sex*). Pokud bychom ale chtěli například vidět, jak se liší křivky přežívání mezi muži a ženami (tj. pro *sex* rovno 1 či 2, a současně pro průměrné hodnoty *ph.ecog*), museli bychom použít funkci *survfit* s parametrem *newdata*:

```
> sf.2s1<-survfit(ph.1,newdata=list(sex=1,ph.ecog=0.9515))
> sf.2s2<-survfit(ph.1,newdata=list(sex=2,ph.ecog=0.9515))
> lines(sf.2s1,col="red")
> lines(sf.2s2,col="blue")
```

Výsledný graf s doplněnými křivkami je v Obr. 39.

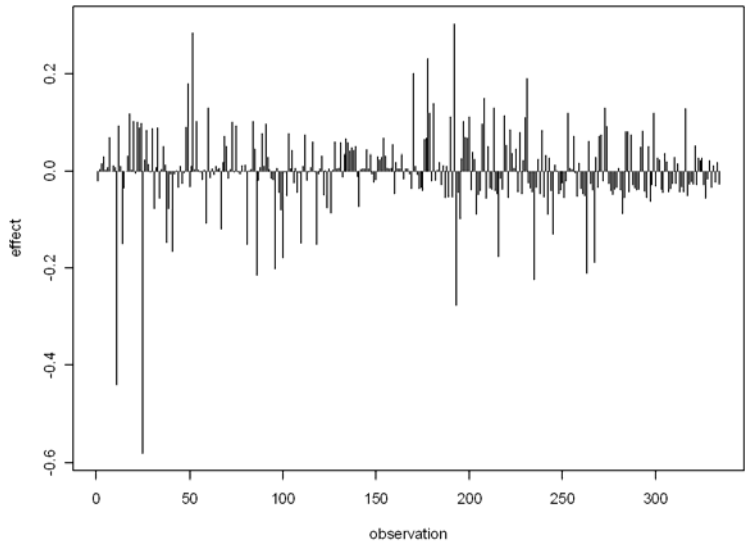


Obr. 39

Vliv jednotlivých pozorování na výsledný fitovaný Coxův model si můžeme graficky znázornit následujícím způsobem:

```
> plot(1:334,resid(ph.1,type="dfbetas"),type="h",xlab="observation",
+ ylab="effect")
```

Ve výsledném grafu (Obr. 40) jsou na horizontální ose jednotlivá pozorování, hodnota na svislé ose vyjadřuje, jak moc se odhady regresních koeficientů změní při vynechání daného pozorování.

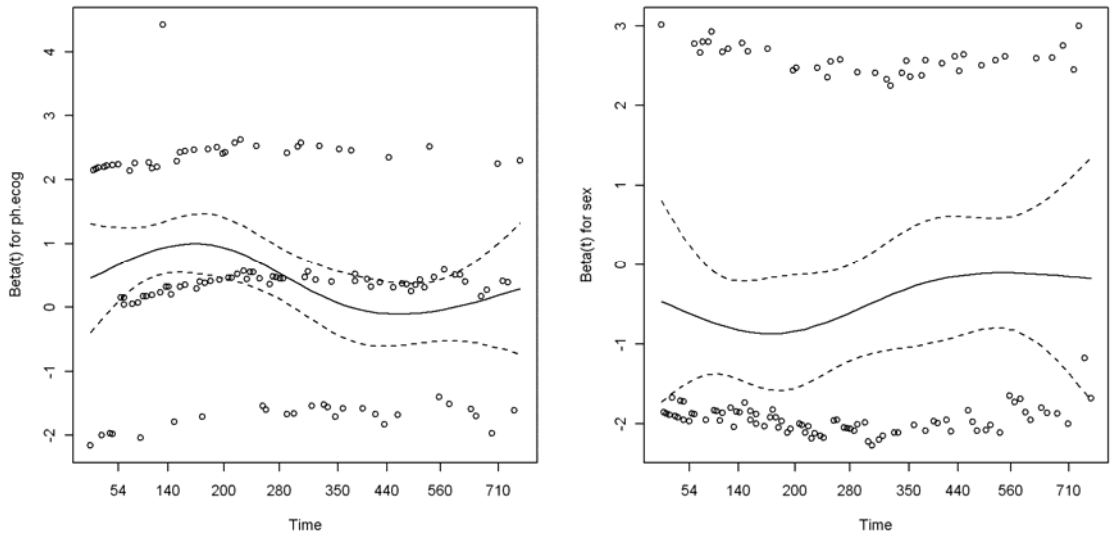


Obr. 40

Jedním z důležitých předpokladů pro použití Coxova modelu je ten, že se vliv prediktorů na míru rizika nemění s dobou trvání rizika. Například v našem příkladě předpokládáme, že vliv hodnoty *ph.ecog* je stejný pro pacienty bezprostředně po diagnóze i po delší době. Tento předpoklad lze ověřit použitím funkce *cox.zph*, například grafickým znázorněním:

```
> par(mfrow=c(1,2))
> plot(cox.zph(ph.1))
> par(mfrow=c(1,1))
```

Výsledný obrázek (Obr. 41) ukazuje, že zatímco pro vliv pohlaví (*sex*) lze předpoklad o konstatním vlivu potvrdit[33], efekt *ph.ecog* se ale s časem uplynulým od diagnózy mění.



Obr. 41

[33] Neměnnost vlivu reprezentuje horizontální čára, která se do konfidenčního regionu pro prediktor *sex* "vejde".

Exaktnější potvrzení těchto závěrů dostaneme ze statistického testu, který stejná funkce také nabízí:

```
> cox.zph(ph.1)
           rho chisq      p
ph.ecog -0.211  5.60 0.0180
sex      0.109  1.38 0.2403
GLOBAL      NA  6.78 0.0337
```

Pokud by šlo o pilotní studii, z tohoto výsledku by vyplývalo doporučení měřit *ph.ecog* pro každého pacienta opakovaně – lze totiž předpokládat, že se toto skóre bude postupně měnit. Pokud bychom tedy měli pro každého pacienta více měření dané vysvětlující proměnné, museli bychom při statistické analýze postupovat trochu odlišným způsobem:

(1) každý pacient by byl charakterizován (podle počtu stanovení *ph.ecog*) jedním nebo více řádky, takže bychom mezi vysvětlující proměnné přidali faktor *pat.id*, který by měl odlišnou hodnotu pro každého z pacientů. Ve vzorci Coxova modelu bychom pak na pravou stranu přidali člen *+cluster(pat.id)*, abychom tak popsali vzájemnou závislost mezi více pozorováními na jednom pacientovi.

(2) levá strana vzorce modelu by také neměla podobu *Surv(time,status)*, nýbrž *Surv(StartTime,StopTime, status*), přičemž pro všechna pozorování na témže pacientovi by byl *status* "censored", s možnou výjimkou posledního (pokud by pacient na konci pozorování zemřel).

7 Regresní a klasifikační stromy

Metody klasifikačních a regresních stromů (classification and regression trees, často označovány zkratkou *CART*) představují asi nejméně parametrický přístup k analýze vlivu vysvětlujících proměnných na kvantitativní (v případě regresního stromu) nebo na kvalitativní (tj. faktor, v případě klasifikačního stromu) vysvětlovanou proměnnou. Přestože nejdůležitějším způsobem zobrazení jejich výsledku je větvený strom s povahou "určovacího klíče", metody CART jsou exaktní, jejich výsledky lze porovnávat (přes množství vysvětlené variability) s jinak sestavenými modely, můžeme testovat hypotézy a vybírat optimální složitost stromu pro predikci neznámých hodnot vysvětlované proměnné.

Motivační příklad

Začneme příkladem s kvantitativní vysvětlovanou proměnnou a (shodou okolností) pouze s kvantitativními prediktory. Taková data bychom tedy mohli analyzovat také pomocí lineárního nebo zobecněného lineárního modelu.

V datech máme údaj, pro jednotlivé správní (a volební) okrsky v rámci státu Texas, kolik procent obyvatel volilo republikánského kandidáta v prezidentských volbách v roce 2000. K vysvětlení budeme používat základní demografické statistiky těchto okrsků (několik okrsků, s méně než dvěma tisíci obyvatel, bylo z dat vyloučeno):

```
> TX<-read.delim("clipboard",row.names=1)
> names(TX)
 [1] "RepPref"     "PopChg"      "Over65"      "White"       "Black"
 [6] "Native"      "Hispanic"    "Foreign"     "UnivDeg"     "BirthRate"
[11] "InfantDeath" "HousSize"    "HomeOwn"     "PovRate"     "HousIncome"
[16] "SocSecRec"   "UnempRate"   "EstablRate"  "IncomePC"    "RetSalesPC"
[21] "BuildPerm"   "FedFunds"    "FarmSize"    "LandValue"   "CrimeRate"
[26] "HiSchl"      "Vehicles"    "FamIncome"   "TransfPC"
> dim(TX)
[1] 236  29
```

Data mají vlastnost typickou pro studie, u kterých je použití CART nezbytné: mají příliš mnoho vysvětlujících proměnných a také značné množství pozorování (často ale bývají data ještě mnohem větší, se stovkami potenciálních vysvětlujících proměnných a desítkami tisíc pozorování). Za takových okolností není vhodné používat v našich modelech všechny prediktory, které máme k dispozici, ale nalezení vhodné podmnožiny se stává obtížným úkolem. CART lze použít k základní orientaci v našich datech, k rozpoznání kvalitních prediktorů a vztahů mezi nimi.

Vzhledem k počtu proměnných není praktické, abychom zde zobrazovali výstup z funkce *summary* aplikované na celý datový rámec, a nakonec ani abychom popisovali význam jednotlivých prediktorů. Podíváme se proto jen na hodnoty vysvětlované proměnné a popíšeme význam až těch prediktorů, které metoda vybere do regresního stromu.

```
> summary(TX$RepPref)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  20.70   60.05   66.85   64.98   73.12   90.70
```

Je vidět, že v Texasu obecně volí radši republikány, nicméně je zde poměrně velká variabilita, kterou by nám mohly demografické údaje vysvětlit.

Regresní stromy

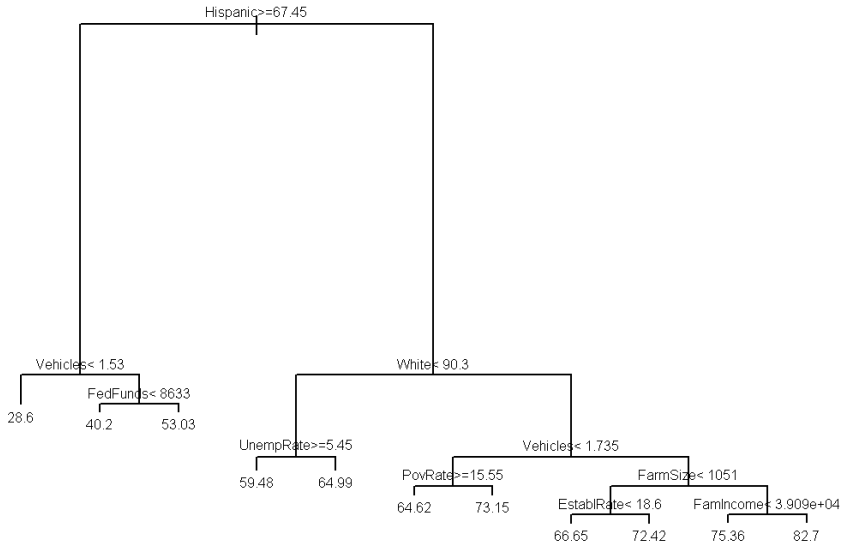
Vlastní regresní strom nafitujeme pomocí funkce *rpart*, která je k dispozici v package stejného jména:

```
> library(rpart)
> rp.1<-rpart(RepPref~.,data=TX,minsplit=7,minbucket=3)
```

Vzorec modelu pojmenovává vysvětlovanou proměnnou a použití tečky na pravé straně značí, že všechny ostatní proměnné v datovém rámci *TX* mohou být použity jako prediktory (pokud je metoda zvolí). Parametr *minsplit* udává, jak velké skupiny pozorování se mohou v rámci stromu ještě dále větvit (viz níže) a parametr *minbucket* zase určuje, jaká může být nejmenší velikost koncové (terminální) skupiny (tj. "listu" našeho stromu). Začneme tedy zobrazením nafitovaného stromu:

```
> plot(rp.1,margin=0.05)
> text(rp.1,cex=0.67)
```

Výsledný graf je v Obr. 42. Funkce *plot* nám nakreslila strukturu stromu, vysvětlující text jsme pak doplnili samostatnou funkcí *text*. Parametr *margin* rezervoval bílé místo kolem zobrazovaného stromečku, tak aby některé delší popisky vytvářené funkcí *text* "nevyběhly" z grafu. Parametr *cex* pro funkci *text* zadává zmenšení znaků na dvě třetiny standardní velikosti.



Obr. 42

Obsah regresního (ale i klasifikačního) stromu je nejsnazší interpretovat stejně jako určovací klíč používaný v biologii. Při určování hodnoty vysvětlované proměnné (průměrné procento republikánských voličů) jsme dotazováni na sérii otázek, jejichž sled závisí na odpovědích na otázky předcházející. Sled začíná u kořene stromu, který se (pro biologa trochu překvapivě) nachází v horní části diagramu. Pokud je odpověď na kteroukoliv z otázek "ano", pokračujeme do levé větve, pokud "ne", směřujeme do pravé větve. Má-li tedy námi uvažovaný volební okrsek například 50% španělsky mluvících (nebo alespoň ke španělskému či latinsko-americkému původu se hlasících) obyvatel,

není podmínka (>=67.45%) splněna a pokračujeme proto doprava. Pokud (a zde si dovolím několik jednotlivých otázek=větvení shrnout) je zde např. 95% bělochů, průměrný počet vozidel na domácnost je např. 1.9, průměrná velikost farem přesahuje 10.5 ha a průměrný příjem rodin je například 40 tisíc dolarů ročně (tj. přesahuje $3.909*10^4$), je předpovídanou preferencí republikánského kandidáta hodnota 82.7%. Pro pochopení ostatních, v předchozím popisu nezmíněných vysvětlujících proměnných doplňuji, že *FedFunds* představuje roční objem federální podpory v USD na hlavu, *UnempRate* je procento nezaměstnanosti, *PovRate* je procento osob s příjmem pod hranicí chudoby a *EstablRate* představuje počet soukromých firem (s výjimkou farem) přepočtený na 1000 obyvatel.

Délka větví (tj. jejich výška v Obr. 42) ukazuje, o kolik se snížila neobjasněná variabilita (měřená residuální sumou čtverců v případě regresního stromu, nebo indexem Gini-ho v případě klasifikačních stromů), tj. jak se zpřesnily naše předpovědi. Otázkám, které zodpovídáme v jednotlivých místech větvení (uzly, nodes), se říká pravidla (rules) a jak je vidět, tatáž proměnná (např. *Vehicles*) může být použita v několika odlišných místech. Chceme-li ale zobrazit strom rozsáhlejší, musíme často upustit od toho, aby délka větve odrážela kvalitu pravidla, a používat uniformní délku větví. Pokud bychom se současně rozhodli přidat do terminálních skupin (listů) informaci, kolik pozorování z našich dat do dané skupiny patří, mohli bychom příkaz zobrazující strom změnit takto (výsledný diagram vynechán):

```
> plot(rp.1,uniform=T,margin=0.05)
> text(rp.1,cex=0.67,use.n=T)
```

Jakým způsobem je strom vytvářen? Jednotlivá větvení stromu představují rozdělení všech (první větvení u kořene) nebo části dat na dvě podskupiny, a to na základě hodnot vždy jedné vysvětlující proměnné. V případě kvantitativní vysvětlující proměnné se snažíme najít takovou hranici v rozsahu jejích hodnot, aby dvě skupiny vzniklé na jejím základě (pozorování s hodnotami menšími resp. většími a rovnými dané hranici) byly navzájem co nejvíce odlišné a vnitřně co nejpodobnější, pokud jde o hodnoty vysvětlované proměnné. Hranice nesmí být příliš blízko maximu nebo minimu (nesmíme "odloupávat" příliš malé skupinky pozorování; dolní hranici velikosti jsme určili parametrem *minbucket* při volání funkce *rpart*). V případě, kdy vysvětlující proměnná má charakter faktoru (s neuspořádanými hladinami), funkce *rpart* se – místo hledání hraniční hodnoty – snaží rozdělit množinu možných hodnot (hladin faktoru) do dvou skupin. Tento postup můžeme zopakovat pro každou z potenciálních vysvětlujících proměnných (tj. pro 28 prediktorů v našem příkladě) a jednotlivé kandidáty můžeme spolu snadno porovnat: jejich srovnávaná kvalita je dána poklesem residuální sumy čtverců pro vysvětlovanou proměnnou. Vítěz porovnání je pak zvolen jako pravidlo. S každou ze dvou vzniklých skupin se celý postup opakuje a v tomto novém dělení mají opět všechny prediktory šanci definovat nové rozdělovací pravidlo. Rozdělování končí v okamžiku, kdy jsou uvažované dvě podskupiny již příliš malé (default hodnotu 20 pozorování jsme snížili pomocí parametru *minsplit*), nebo je skupina již dostatečně homogenní (v hodnotách vysvětlované proměnné).

Klestění stromů

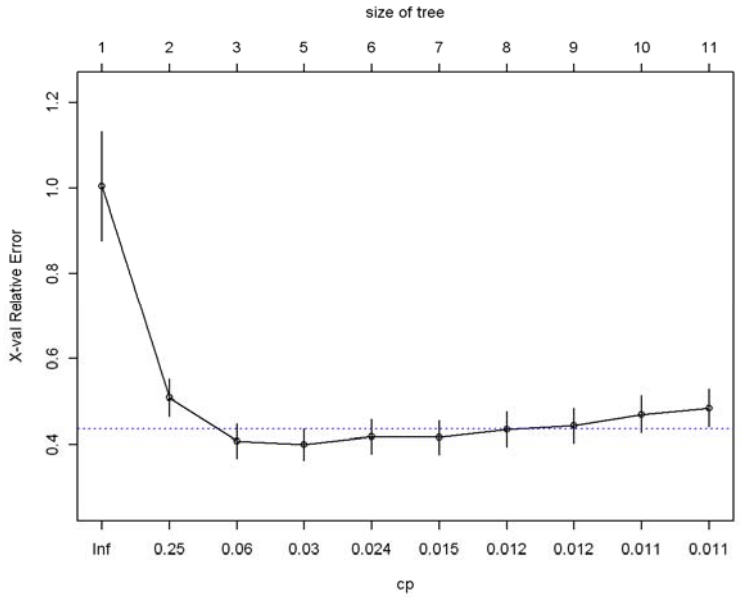
Strom, který nám vytvoří funkce *rpart* je obvykle "přerostlý". Některé z jeho větví mají sice své oprávnění pro vysvětlení specifických případů a korelací v našem souboru dat (tzv. training set, ze kterého byl strom sestrojen), nicméně nemáme jistotu, že všechna ta pravidla budou opravdu zvyšovat naši schopnost např. předpovídat voličskou preferenci, pokud je aplikujeme na data nová (která funkce *rpart* "neviděla"). Příliš složitý strom má obvykle horší výkon než strom jednodušší, navíc je jeho interpretace obtížnější. Jak ale zjistit kvalitu stromu ve vztahu k jeho složitosti? Pokud budeme postupně odřezávat krátké (tj. nejméně významné) koncové větvičky a zjišťovat změnu residuální sumy čtverců (tj. odlišnost mezi skutečnými a předpovídanými hodnotami vysvětlované proměnné), zjistíme jen, že residuální suma čtverců s postupujícím klestěním roste. To je obdobné situaci např. u (zobecněných) lineárních modelů, kde přidání prediktoru libovolné kvality vždy vede ke snížení nevysvětlené variability. U parametrických modelů jsme tento problém řešili buď pomocí testu signifikance nebo pomocí odhadu AIC, u metody CART postupujeme odlišně, za pomocí tzv. krosvalidace (cross-validation).

Při ní nejsou pro hodnocení toho, jak kvalitně strom dané složitosti předpovídá hodnoty vysvětlované proměnné, použita stejná data jako ta, ze kterých byl strom sestrojen (tj. ve kterých byla rozhodovací pravidla nalezena). Kde ale sebrat druhý, nezávislý soubor dat? Rozdělení našich dat na dvě části, z nichž jednu použijeme pro sestrojení stromu a druhou pro hodnocení, jak ho optimálně proklestit, si obvykle nemůžeme dovolit, vzhledem k finanční či časové náročnost sběru dat. Krosvalidace zde postupuje metodou "chytré horákyně". Naše pozorování jsou rozdělena do K (obvykle K=10) skupin a sestrojování a následné prořezávání (kombinované s vyhodnocováním) stromu probíhá také K-krát. Pokaždé se jedna z těchto skupin nepoužije k sestrojení stromu (tj. použije se zbylých $K-1$ skupin) a po jeho konstrukci se pomocí této skupiny vyhodnotí přesnost predikcí, odřízne se nejméně významné větvička, opět se strom vyhodnotí, a s klestěním se pokračuje až do stádia "pařez" (tj. nulového modelu, kterým předpovídáme jen průměrnou hodnotou vysvětlované proměnné, žádné větvení nám nezbylo). Vzhledem k tomu, že pro různé složitosti stromu máme několik (K) hodnocení kvality, můžeme je zprůměrovat a také stanovit přesnost tohoto odhadu.

Funkce *rpart* automaticky krosvalidaci provádí (pokud jí v tom nezabráníme použitím argumentu *xval=0*, parametr vlastně určuje hodnotu K), jen si její výsledek musíme vyžádat:

```
> plotcp(rp.1,col="blue")
```

Výsledný diagram je v Obr. 43.



Obr. 43

Vidíme v něm vynesené hodnoty chyb v předpovídání procenta voličů (na svislé ose) pro stromy postupně se zvětšující složitosti. Složitost stromu je popsána jak počtem koncových větviček (size of tree, na horní vodorovné ose), tak pomocí parametru složitosti (complexity parameter, *cp*) na ose spodní. Kvalita stromu rychle roste až do velikosti 3, velikosti 5, 6 a 7 mají velmi podobnou kvalitu a s dále rostoucí složitostí stromu jeho výpovědní hodnota klesá, tj. nezávisle (krosvalidačně) odhadnutá chyba roste. Referenční modrá čára představuje nejmenší hodnotu krosvalidované chyby (pro velikost 5), s přičtenou standardní chybou odhadu (není tedy náhodou, že se svislá čárka vynášející tuto chybu kolem průměru pro *cp*=0.03 modré čáry dotýká). Tato zvláštní hodnota je vynesena proto, že v Obr. 43 patrný pattern (rychlé zlepšování, relativně mělká a široká oblast optima a za ní pozvolný pokles kvality) se vyskytuje dosti často. Simulačními studiemi bylo zjištěno, že je (alespoň pro některé účely) vhodné vybrat největší hodnotu *cp* (tj. nejmenší strom), pro kterou průměrná chyba "podleze" znázorněnou limitu (tzv. 1-SE pravidlo). V našem případě by to tedy bylo *cp=0.06*, odpovídající stromu o velikosti *3*, tj. pouze se dvěma větveními. Takto jednoduchý strom odpovídá principu parsimonie a tato volba je doporučována, pokud fitování stromu mělo za cíl porozumět vztahům a popsat, jaké faktory o preferencích rozhodují. Pokud je ale naším cílem především predikce (předpovídání nových hodnot), je doporučováno se držet složitosti odpovídající skutečnému minimu (tj. zde strom s pěti koncovými větvičkami). Tak budeme postupovat i v našem případě[34], získáme tím jen trochu "upovídanější" stromeček. Než si jej ale vytvoříme (proklestěním *rp.1*) a zobrazíme, podívejme se na výstup funkce *printcp*, která nám v textové podobě poskytuje údaje srovnatelné s předchozím diagramem:

[34] i když náš cíl je spíše porozumění než predikce – byly přeci charakterizovány všechny okrsky a tak není pro prezidentské volby v roce 2000 v Texasu už co předpovídat...

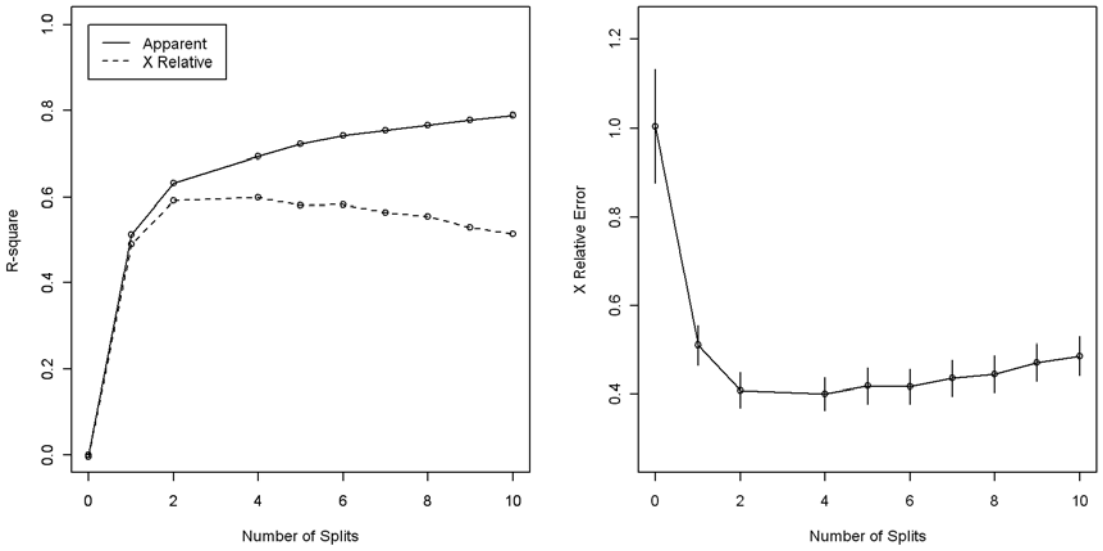
```
> printcp(rp.1)
Regression tree:
rpart(formula = RepPref ~ ., data = TX, minsplit = 7, minbucket = 3)

Variables actually used in tree construction:
[1] EstablRate FamIncome  FarmSize   FedFunds   Hispanic   PovRate    UnempRate
[8] Vehicles   White

Root node error: 35084/236 = 148.66
n= 236

         CP nsplit rel error  xerror     xstd
1  0.513449      0   1.00000 1.00447 0.128256
2  0.119559      1   0.48655 0.50963 0.044402
3  0.030483      2   0.36699 0.40817 0.040768
4  0.029880      4   0.30603 0.39964 0.037620
5  0.018832      5   0.27614 0.41826 0.041047
6  0.012581      6   0.25731 0.41626 0.039534
7  0.012235      7   0.24473 0.43560 0.041249
8  0.011424      8   0.23250 0.44456 0.041330
9  0.011065      9   0.22107 0.47027 0.042964
10 0.010000     10   0.21001 0.48523 0.043870
```

Po počátečních údajích o modelu (včetně skutečně použitých prediktorů) je vypsána tabulka, jejíž obsah sice víceméně odpovídá grafickému zobrazení (viz Obr. 43), ale obsahuje několik "záludností". Zaprvé, hodnoty *CP* jsou odlišně škálovány než v grafu a při redukci složitosti stromu pomocí funkce *prune* (viz dále) musíme zadávat hodnoty z grafu, nikoliv zde zobrazená čísla. Zadruhé, sloupec *nsplit* uvádí počet větvení, nikoliv počet koncových větviček, tj. číslo o jedničku menší než hodnoty na horní vodorovné ose grafu. Hodnoty ve sloupcích *xerror* a *xstd* jsou díkybohu stejné jako ty, které ukazuje diagram (jako body a svislé čáry na obě strany od příslušného bodu – průměru). Navíc zde máme sloupec s označením *rel. error*, který udává relativní pokles chyb ve srovnání s nulovým modelem. Alternativně můžeme tuto hodnotu interpretovat (v případě kvantitativní vysvětlované proměnné) jako hodnotu získanou odečtením koeficientu determinace od jednotky (tato interpretace neplatí pro první řádek, odpovídající nulovému modelu). Jinými slovy, R^2 pro model s jedním větvením je roven *1.0-0.48655*. Jde ale o tzv. "zjevnou" (apparent) hodnotu koeficientu determinace, tj. získanou nikoliv z krosvalidace, ale vytvořením stromu ze všech dat a jeho následným vyhodnocováním pomocí stejných pozorování.

Alternativní diagram, který obsahuje i krosvalidovaný odhad R^2, získáme pomocí funkce *rsq.rpart*. Ve skutečnosti ale vytvoří dva diagramy (druhý z nich ale jen opakuje výstup již dříve použité funkce *plotcp*), kreslící plochu proto musíme rozdělit na dvě poloviny:

```
> par(mfrow=c(1,2))
> rsq.rpart(rp.1)

...

> par(mfrow=c(1,1))
```

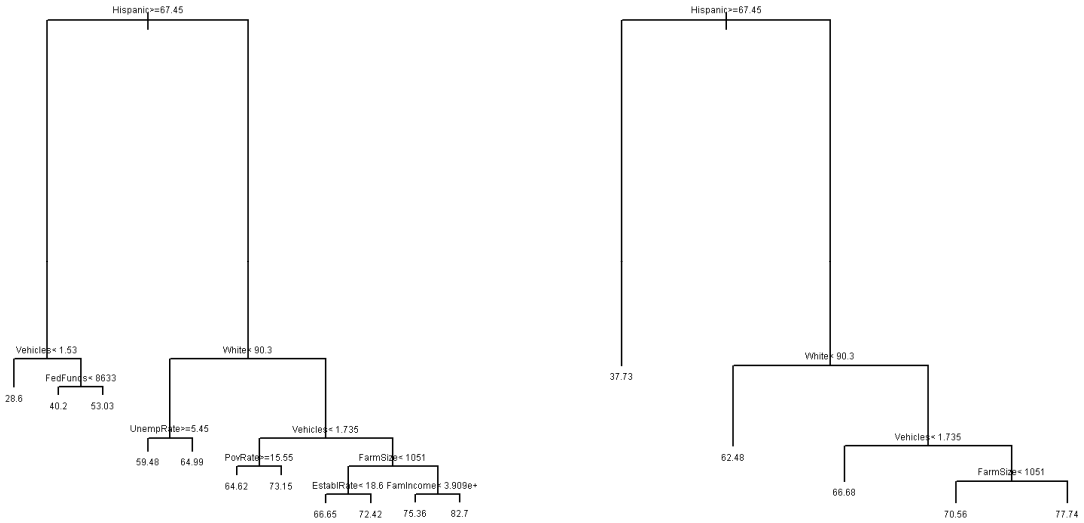
Diagram v levé části Obr. 44 znázorňuje jak zjevnou hodnotu (plná čára), tak krosvalidovaný odhad koeficientu determinace. Vidíme, že pro nová (neznámá) pozorování roste kvalita předpovědi do velikosti stromu 3 či 5, pak opět klesá. Na tomto místě je třeba zdůraznit, že výpovědi obou těchto diagramů nejsou nezávislé, jde o shodnou výpověď popsanou odlišným způsobem.



Obr. 44

Oklestěný strom (*rp.2*) získáme a zobrazíme (spolu s původním) následujícími příkazy:

```
> rp.2<-prune(rp.1,cp=0.03)
> par(mfrow=c(1,2))
> plot(rp.1);text(rp.1,cex=0.5)
> plot(rp.2);text(rp.2,cex=0.5)
> par(mfrow=c(1,1))
```



Obr. 45

Také si můžeme obsah stromu zobrazit v textové podobě:

```
> rp.2
n= 236

node), split, n, deviance, yval
      * denotes terminal node

 1) root 236 35084.470 64.98051
```

```
   2) Hispanic>=67.45 22  2004.048 37.73182 *
   3) Hispanic< 67.45 214 15066.340 67.78178
     6) White< 90.3 88  3705.458 62.48409 *
     7) White>=90.3 126  7166.208 71.48175
      14) Vehicles< 1.735 33  1939.682 66.68485 *
      15) Vehicles>=1.735 93  4197.746 73.18387
        30) FarmSize< 1051 59  1977.520 70.56102 *
        31) FarmSize>=1051 34  1110.018 77.73529 *
```

Kompetující a náhradní prediktory

Při popisu způsobu, kterým funkce *rpart* strom vytváří, jsme zmínili, že potenciální prediktory jsou při každé dělení vyhodnoceny všechny a je vždy vybrán ten, který objasněnou variabilitu zvětší nejvíce. Často se ale stává, že několik dalších "kandidátů" není o mnoho horších, a pro lepší pochopení vztahů v našich datech bychom se o nich také rádi něco dozvěděli. Funkce *rpart* tuto informaci uchovává pro zvolený (a parametrem změnitelný) počet nejlepších kandidátů v každém místě větvení.

Musíme si uvědomit, že ačkoliv tato alternativní pravidla mají stejný cíl (vytvoření podskupin s co nejvíce homogenními hodnotami vysvětlované proměnné), nebývá cesta k jeho dosažení u všech prediktorů stejná (pokud spolu nejsou významně korelované), tj. obsah dvou podskupin, které by alternativní pravidla vytvořila, může být odlišný. Tyto alternativní kandidátní proměnné pro rozdělovací pravidlo se proto v package *rpart* nazývají všechny *primary splits*, ačkoliv se z nich vybere nakonec jen jeden (ten nejlepší).

Poté, co jsme nejlepší prediktor vybrali, by nás ale mohlo také zajímat, jak jsou vybrané pravidlo (tj. rozdělení do dvou podskupin, podle hodnot vybraného prediktoru) schopny reprodukovat jiné vysvětlující proměnné. Odpovídající pravidla můžeme použít místo vybraného primárního pro ta pozorování, u kterých údaj o hodnotě primárního prediktoru chybí. Je to podobná situace jako v botanickém určovacím klíči, ve kterém se dva druhy rozliší nejlépe podle znaků na květech, nicméně rádi bychom rostlinu určili (byť s větší nejistotou) také v případě, že právě nekvete. Informaci o náhradních pravidlech (*surrogate splits*) získáme spolu s údaji o kompetujících primárních prediktorech pomocí funkce *summary*. Její výstup je značně rozsáhlý a v následující ukázce zobrazuji jen jeho část:

```
> summary(rp.2)
Call:

...

Node number 1: 236 observations,    complexity param=0.5134489
  mean=64.98051, MSE=148.663
  left son=2 (22 obs) right son=3 (214 obs)
  Primary splits:
      Hispanic  < 67.45   to the right, improve=0.5134489, (0 missing)
      PovRate   < 23.2    to the right, improve=0.4478818, (0 missing)
      FamIncome < 29672.5 to the left,  improve=0.4090632, (0 missing)
      Vehicles  < 1.545   to the left,  improve=0.4043338, (0 missing)
      HiSchl    < 58.35   to the left,  improve=0.3728125, (0 missing)
  Surrogate splits:
      PovRate    < 23.2    to the right, agree=0.983, adj=0.818, (0 split)
      HiSchl     < 58.9    to the left,  agree=0.979, adj=0.773, (0 split)
      FamIncome  < 29450   to the left,  agree=0.975, adj=0.727, (0 split)
      Vehicles   < 1.545   to the left,  agree=0.966, adj=0.636, (0 split)
```

```
      HousIncome < 24336.5 to the left,  agree=0.953, adj=0.500, (0 split)

Node number 2: 22 observations
  mean=37.73182, MSE=91.09308

Node number 3: 214 observations,    complexity param=0.1195593
  mean=67.78178, MSE=70.40345
  left son=6 (88 obs) right son=7 (126 obs)
  Primary splits:
      White     < 90.3    to the left,  improve=0.2784136, (0 missing)
      Black     < 7.3     to the right, improve=0.2776230, (0 missing)
      UnempRate < 4.65    to the right, improve=0.2196997, (0 missing)
      LandValue < 535.5   to the right, improve=0.1914020, (1 missing)
      Vehicles  < 1.745   to the left,  improve=0.1912031, (0 missing)
  Surrogate splits:
      Black     < 7.75    to the right, agree=0.967, adj=0.920, (0 split)
      FarmSize  < 386.5   to the left,  agree=0.804, adj=0.523, (0 split)
      LandValue < 655.5   to the right, agree=0.757, adj=0.409, (0 split)
      UnempRate < 5.05    to the right, agree=0.710, adj=0.295, (0 split)
      Vehicles  < 1.745   to the left,  agree=0.706, adj=0.284, (0 split)
```

...

Pro první dělení ("Node number 1") do dvou skupin (s 22 a 214 pozorováními) bylo vybráno pravidlo "Hispanic < 67.45". Pozor, je zde zobrazeno s opačnou "polaritou" a proto také říká, že při splnění nerovnosti musíme postoupit do pravé větve[35]. Funkce *summary* ale ukazuje i další kompetující pravidla, a to v sekci "Primary splits". Vidíme, že poměrně velkou predikční hodnotu má i pravidlo založené na procentu lidí pod hladinou chudoby. Relativní kvalitu pravidel udává parametr "improve", představující snížení hodnoty residuální sumy čtverců. V následující sekci "Surrogate splits" jsou uvedena náhradní pravidla, vybraná tak, aby reprodukovala co nejlépe vybrané primární pravidlo (tj. "Hispanic < 67.45"). Často se zde opakují prediktory zobrazené již v předchozí sekci, ale hraniční hodnoty nemusí být vždy stejné: v tomto kontextu již není hlavním cílem maximalizovat kvalitu skupin. Míra shody s predikcí pomocí primárního pravidla je v podobě korelace udávána parametrem "agree".

Klasifikační stromy

Jako příklad vhodný pro tvorbu klasifikačního stromu použijeme data, která jsou součástí package MASS. Ukážeme si zde i podobu výstupu v případě, kdy jako prediktor používáme faktor[36].

```
> data(shuttle,package="MASS")
> summary(shuttle)
 stability   error   sign       wind         magn     vis
 stab :128   LX:64   nn:128   head:128   Light :64   no :128
 xstab:128   MM:64   pp:128   tail:128   Medium:64   yes:128
             SS:64                       Out   :64
             XL:64                       Strong:64
```

[35] V grafech mají všechna pravidla stejnou polaritu: je-li odpověď ano, jdeme vždy do levé větve, kde je také (v případě regresních stromů) nižší hodnota vysvětlované proměnné.

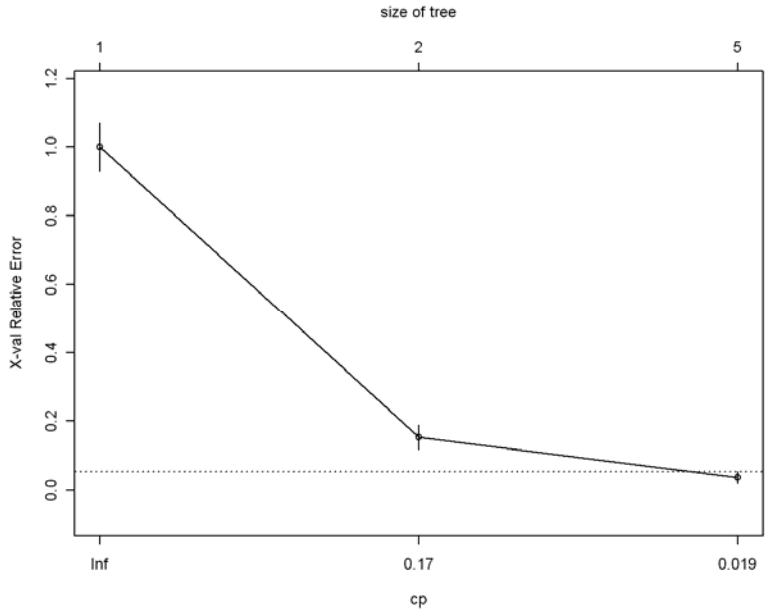
[36] To, že v příkladu regresního stromu byly všechny prediktory kvantitativní a teď, v příkladu klasifikačního stromu, jsou naopak všechny faktory, je jen nešťastnou náhodou. Ve skutečnosti lze jak pro regresní, tak pro klasifikační stromy používat směs kvantitativních a kvalitativních prediktorů.

```
    use
 auto  :145
 noauto:111
```

Jde o reálná data, představující shromážděná doporučení expertní komise při přípravě letů raketoplánu. Tato doporučení popisují okolnosti, za kterých by měla posádka nechat přistávací manévr na počítači (faktor *use* má pak hodnotu *auto*) a za kterých má být přistání provedeno ručně (*use* s hodnotou *noauto*). Rozhodování je ovlivněno stabilitou raketoplánu na předem naplánované dráze sestupu (*stability*), velikostí (*error*) a směrem (*sign*) odchylky od dráhy, směrem větru v místě přistání (*wind* – čelní resp. v zádech), sílou větru (*magn*) a také viditelností v oblasti přistání (*vis*). Ačkoliv poskytnutá data popisují vpodstatě všechny možné kombinace podmínek ($2^8 = 256$ kombinací), pro jejich efektivní použití, a také pro ověření jejich konsistentnosti, byl tento návod převeden do soustavy pravidel a my si můžeme postup převodu ukázat pomocí funkce *rpart*.

```
> rp.3<-rpart(use~.,data=shuttle,minsplit=2,minbucket=1)
> plotcp(rp.3)
```

Diagram nám ukazuje (Obr. 46), že v tomto případě není pro klestění žádný prostor.

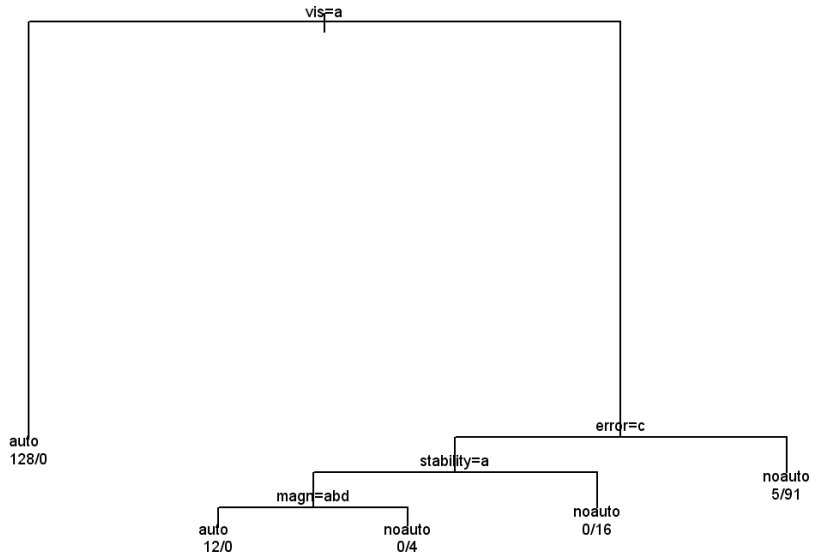


Obr. 46

Další větvení stromu ale také nepřipadá v úvahu – funkci *rpart* jsme sdělili, že i dělení skupiny se dvěma pozorováními je pro nás přijatelné, nicméně vytvořený strom má více omezené větvení, protože výsledné skupiny byly dostatečně homogenní. Jinými slovy, výsledný strom je "té pravé" velikosti, aby reprezentoval expertní vyjádření ohledně přistávacích pravidel. Nyní si ještě vytvořený strom znázorníme graficky (Obr. 47) a textově:

```
> plot(rp.3,margin=0.05)
> text(rp.3,cex=3/4, use.n=T)

> rp.3
n= 256
```

```
node), split, n, loss, yval, (yprob)
      * denotes terminal node

 1) root 256 111 auto (0.56640625 0.43359375)
   2) vis=no 128   0 auto (1.00000000 0.00000000) *
   3) vis=yes 128  17 noauto (0.13281250 0.86718750)
     6) error=SS 32  12 noauto (0.37500000 0.62500000)
      12) stability=stab 16   4 auto (0.75000000 0.25000000)
        24) magn=Light,Medium,Strong 12   0 auto (1.00000000 0.00000000) *
        25) magn=Out 4   0 noauto (0.00000000 1.00000000) *
      13) stability=xstab 16   0 noauto (0.00000000 1.00000000) *
     7) error=LX,MM,XL 96   5 noauto (0.05208333 0.94791667) *
```



Obr. 47

Rozhodovací pravidla používající jako vysvětlující proměnnou faktor odkazují na jeho jednotlivé hladiny pořadovými písmenky (*a*,*b*,*c*,...), a to může interpretaci obrázku ztížit. Musíme totiž často použít funkce *levels*, abychom určili, v jakém pořadí jsou jednotlivé hladiny označeny. Textový výstup je úsporný, a proto je i lépe čitelný. Nicméně, pokud máme v obrázku dost místa pro delší popisky, můžeme dosáhnout pěkného grafu úpravou použití funkce *text*, s parametrem *pretty* s hodnotou *0* (implicitní hodnotou je *NULL*, a ta vede k náhradě písmenky).

Všimněme si ještě odlišné podoby textového zobrazení objektu představujícího klasifikační strom (výstup z funkce *summary* výše). Za vlastním pravidlem je (shodně s regresním stromem) počet pozorování, která jsou dělena nebo patří do dané koncové skupiny, pak následuje míra nepřesnosti predikce (počet špatně klasifikovaných pozorování) a nakonec hodnota vysvětlované proměnné (faktoru), která je v daném místě stromu predikována (to je ale zajímavé hlavně pro koncové větve). Predikována je samozřejmě ta hladina, která je (nej)více pravděpodobná. Pravděpodobnosti, že pozorování "určené" podle daných pravidel bude mít určitou hladinu vysvětlovaného faktoru, jsou udávány čísly v závorkách. V našem případě (faktor se dvěma hladinami) jsou tato čísla jen dvě, ale může jich být obecně i více a samozřejmě sčítají do celkové hodnoty *1.0*. V našem příkladě je vidět hned v prvém větvení jednoznačné pravidlo, že

v případě nízké viditelnosti je třeba přistání nechat na počítači, v ostatních případech závisí rozhodnutí na míře chyby (je-li velká, je třeba ruční přistání – v tomto pravidlu je určitá nejistota, asi pěti-procentní: *0.05208*). V případě menší chyby závisí rozhodnutí jednoznačně na stabilitě a rozsahu odchylky.

V tomto příkladu, ve kterém jsme se snažili získat výstižný obraz "modelu" v myslích expertů, se také ukazuje[37], že pravidla jsou natolik jednoznačná (a faktory vzájemně nekorelované), že nelze najít náhradní prediktory, takže v případě absence určitého údaje je třeba rozhodnutí založit na doporučení v tom místě větvení, do kterého jsme se propracovali.

[37] čtenář si to může ověřit pomocí *summary(rp.3)*

8 Lineární a nelineární modely se smíšenými efekty

Lineární (a nelineární) smíšené modely (linear [non-linear] mixed-effects models, dále LME a NLME modely) rozšiřují klasické lineární a nelineární regresní modely stejným způsobem, jako hierarchická ANOVA nebo ANOVA s opakovanými měřeními rozšiřují standardní modely analýzy variance. Kromě tzv. pevných efektů (faktory nebo kvantitativní prediktory) přibývají v (N)LME modelech i faktory s náhodným efektem, které umožňují popsat náhodnou variabilitu i na jiných úrovních než pro jednotlivá pozorování. Můžeme tak analyzovat data, ve kterých nejsou jednotlivá pozorování úplně nezávislá, např. růstové křivky fitované z opakovaných měření určitých jedinců. LME a NLME modely ovšem nabízejí i další rozšíření, například přesnější popis vlastností stochastické variability nebo přesnějšího popis charakteru závislosti mezi jednotlivými pozorováními (prostorová či časová korelace). Těchto pokročilejších rozšíření se ale v tomto úvodu dotkneme jen okrajově.

Motivační příklad

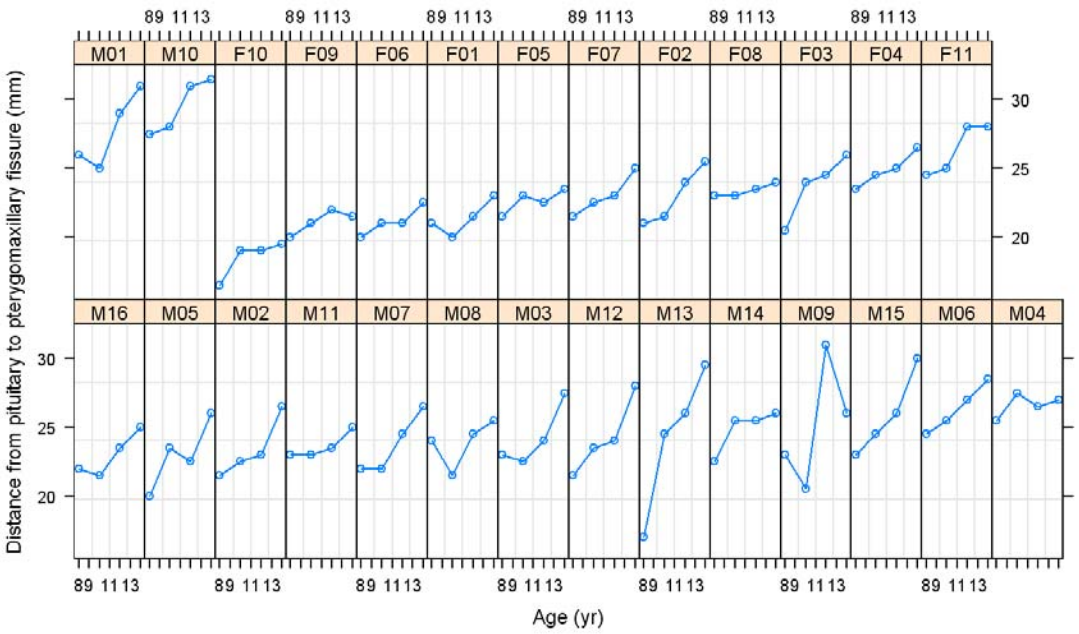
Při seznámování s LME modely budeme pracovat s příkladovými daty, která jsou součásti package *nlme*, což je knihovna funkcí implementujících jak tyto, tak NLME modely.

```
> library(nlme)
> summary(Orthodont)
    distance          age          Subject       Sex
 Min.   :16.50   Min.   : 8.0   M16    : 4   Male  :64
 1st Qu.:22.00   1st Qu.: 9.5   M05    : 4   Female:44
 Median :23.75   Median :11.0   M02    : 4
 Mean   :24.02   Mean   :11.0   M11    : 4
 3rd Qu.:26.00   3rd Qu.:12.5   M07    : 4
 Max.   :31.50   Max.   :14.0   M08    : 4
                                (Other):84
> Or<-Orthodont
```

Poslední příkaz nám umožní odkazovat na příkladová data kratším jménem. Data obsahují údaje o anatomicky definované vzdálenosti (*distance*) odrážející velikost lebky chlapců a děvčat, změřené z rentgenových snímků. Každý jedinec byl měřen celkem čtyřikrát (v 8, 10, 12 a 14 letech) – stáří je zaznamenáno v proměnné *age*, faktor *Subject* udává, která čtyři pozorování patří vždy k jednomu jedinci, proměnná *Sex* udává pohlaví (hodnota se, ovšem, nemění v rámci jedince).

Data můžeme shrnout i graficky (Obr. 48):

```
> plot(Or)
```



Obr. 48

Jednoduchost zadání a elegance grafického výsledku vypadá magicky, je zde ale "více, než oči vidí" – krásný graf je výsledkem akcí "za oponou", jak si ukážeme později.

Data pro LME a NLME

Data, která popisujeme LME modely, mají obvykle definovanou vnitřní strukturu: jednotlivá pozorování patří do skupin, které představují opakovaně pozorované jedince, opakovaně popisované plochy, prostorově sblížené skupiny ploch a podobně. Informaci o tom, která z proměnných definuje tyto skupiny (může jich být i více u hierarchicky uspořádaných dat), ale také o tom, co je hlavní vysvětlovaná proměnná a co je primární prediktor (vysvětlující proměnná), ukládáme v případě dat pro (N)LME modely jako **atribut** datového rámce. Výsledný datový objekt už pak není "obyčejný" datový rámec, je to objekt třídy *groupedData*:

```
> class(Or)
[1] "nfnGroupedData" "nfGroupedData"  "groupedData"    "data.frame"
```

Výše uvedenou základní informaci můžeme z objektu *groupedData* získat pomocí extrakční funkce *formula*:

```
> formula(Or)
distance ~ age | Subject
```

Tento vzorec nám (a funkcím knihovny *nlme*, které s daty pracují) říká, že vysvětlujeme hodnoty proměnné *distance* především pomocí věku (*age*) osoby, a že skupiny pozorování jsou definovány proměnnou *Subject*. Uchování této informace pohromadě s vlastními daty nám pak umožňuje zjednodušit zadávané regresní modely – funkci *lme* pak není třeba dodávat informaci o proměnné definující skupiny, najde si ji v datech sama. Takový postup ale čtenáři spíše nedoporučuji – jednodušší příkazy jsou draze zaplaceny tím, že se náš postup stává při jeho pozdějším zkoumání neprůhledným.

Informací o datech je ale k vlastnímu datovému rámci přidáno obvykle více – ke vzorci lze přidat i informaci o plných názvech proměnných a názvech jednotek, ve kterých byly vysvětlovaná proměnná a primární prediktor měřeny. Objekt *Or* (původně *Orthodont*) tyto informace již obsahuje, ale v následujícím kódu si předvedeme, jak bychom takový objekt sami vytvořili z "obyčejného" datového rámce (který z něj – čistě pro tento didaktický účel – vyextrahujeme):

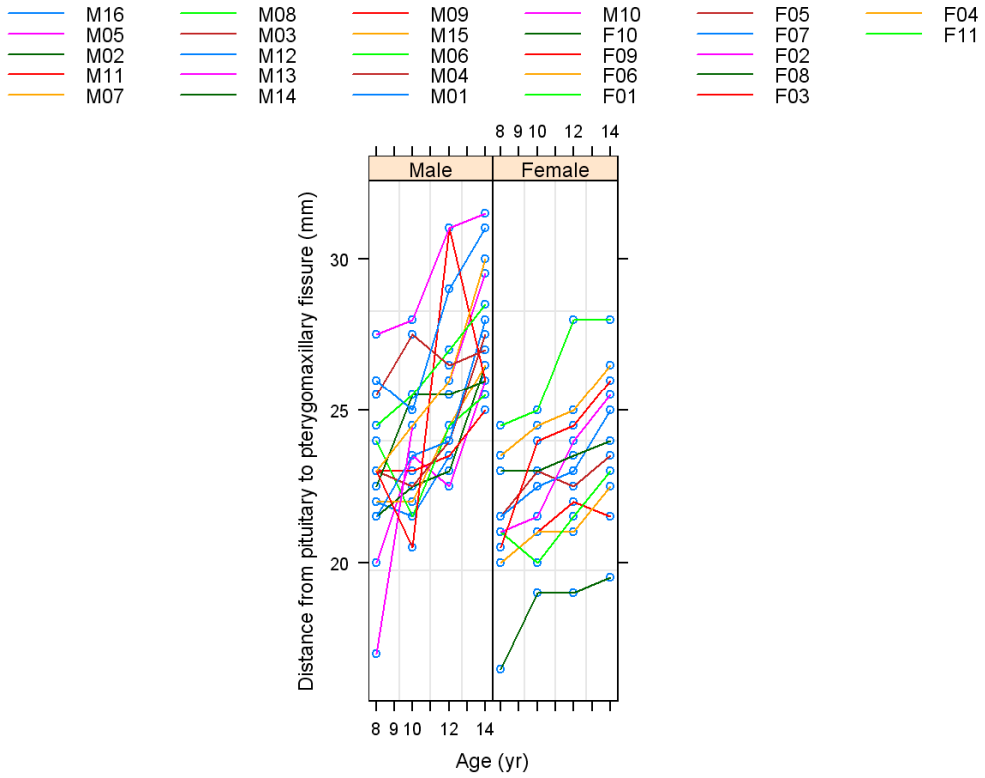
```
> Ort.df<-data.frame(Or)
> Orth.new <- groupedData( distance ~ age | Subject, data = Ort.df,
+   FUN = mean, outer = ~ Sex,
+   labels = list(x = "Age",
+                 y = "Distance from pituitary to pterygomaxillary fissure"),
+   units = list( x = "(yr)", y = "(mm)") )
```

Význam většiny parametrů bude asi zřejmý. Parametr *FUN* určuje funkci, jejíž výsledek (aplikovaný na hodnoty vysvětlované proměnné v jednotlivých skupinách) udává, v jakém pořadí budou jednotlivé skupiny v grafech vynášeny (zde například osoby v pořadí rostoucího průměru hodnot *distance*). Parametr *outer* udává faktor (nebo faktory, oddělené hvězdičkou), které představují proměnné stojící "nad jednotlivými skupinami", tj. obecně faktory, jejichž hodnota se uvnitř skupin (pro jedince) nemění. Existuje i podobný parametr *inner*, pro faktory odlišující různé kategorie pozorování uvnitř skupin.

Parametr *outer* můžeme použít při grafické sumarizaci dat k tomu, abychom oddělili kategorie příslušného faktoru do samostatných panelů. Například pro naše data takto můžeme použít faktor *Sex* (hodnota *T* pro parametr *outer* znamená "použij tu proměnnou (ty proměnné), která byla označena jako *outer* v definici datového objektu"):

```
> plot(Or,outer=T)
```

Výsledný diagram (v Obr. 49) má křivky rozdělené do dvou panelů, podle pohlaví osob.



Obr. 49

Nezvyklý poměr šířky a výšky panelů byl funkcí *plot* zvolen proto, aby byl vytvořen průměrný sklon křivek 45 stupňů, který je považován za optimální pro jejich porovnávání.

Dílčí lineární modely a volba náhodných efektů

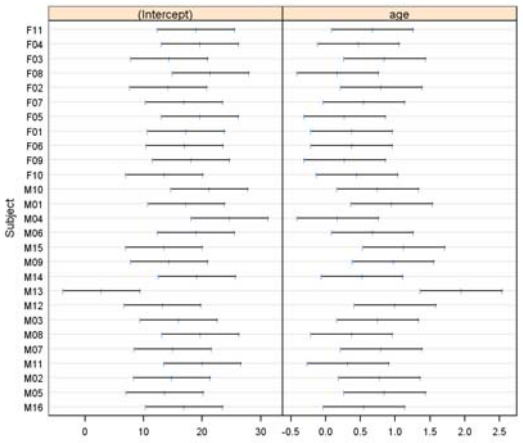
Naším prvním cílem bude popsat lineární závislost zkoumané vzdálenosti na věku dítěte (bez ohledu na pohlaví) a rozšířit tento model o náhodné efekty, popisující odlišnosti mezi dětmi. Tyto individuální odlišnosti se mohou týkat jak průměrné hodnoty (tj. různé děti se liší velikostí své lebky bez ohledu na věk), tak sklonu přímky (různé děti se liší rychlostí, kterou se jejich lebka zvětšuje). Zda je náhodný efekt vhodné do modelu zahrnout pro jeden nebo oba tyto koeficienty, můžeme zjistit porovnáním přímek fitovaných pro jednotlivé osoby. Knihovna *nlme* nám pro tento účel nabízí funkci *lmList*:

```
> lmlist.1<-lmList(distance~age|Subject,data=Or)
> summary(lmlist.1)
Call:
  Model: distance ~ age | Subject
   Data: Or

Coefficients:
   (Intercept)
    Estimate Std. Error   t value     Pr(>|t|)
M16    16.95   3.288173 5.1548379 3.695247e-06
M05    13.65   3.288173 4.1512411 1.181678e-04
...
F04    19.65   3.288173 5.9759625 1.863600e-07
```

```
F11    18.95   3.288173 5.7630783 4.078189e-07
   age
    Estimate Std. Error   t value     Pr(>|t|)
M16    0.550  0.2929338 1.8775576 6.584707e-02
M05    0.850  0.2929338 2.9016799 5.361639e-03
...
F04    0.475  0.2929338 1.6215270 1.107298e-01
F11    0.675  0.2929338 2.3042752 2.508117e-02

Residual standard error: 1.310040 on 54 degrees of freedom
```

Na případný vliv náhodných efektů (tj. rozdílů mezi jedinci v průměrné hodnotě nebo v rychlosti růstu) lze usoudit, spíše než z takovéto dlouhé tabulky čísel, prozkoumáním grafu. Pro porovnání vlivu náhodné variability mezi jedinci na jednotlivé parametry modelu se nejlépe hodí graf, který vytvoříme vynesením konfidenčních intervalů pro jednotlivé koeficienty a tyto intervaly nám vypočte funkce *intervals*:

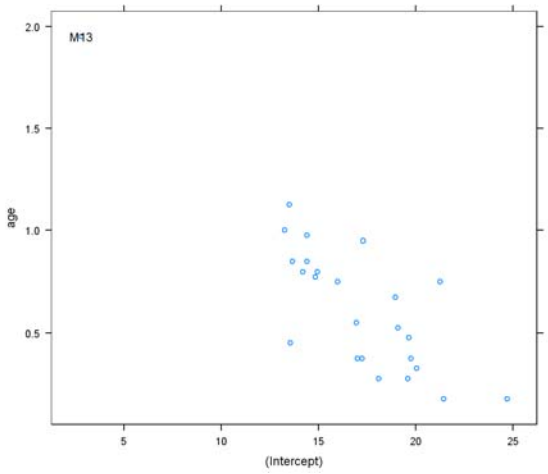
```
> plot(intervals(lmlist.1))
```



Obr. 50

Výsledný graf v Obr. 50 má zvláštní vlastnost: hodnoty obou koeficientů jsou uspořádány zrcadlově kolem čáry oddělující oba panely. Je-li hodnota průsečíku nízká, sklon přímky je zase vyšší, a naopak. Povahu vztahu mezi oběma koeficienty uvidíme lépe, použijeme-li jinou funkci (viz též Obr. 51):

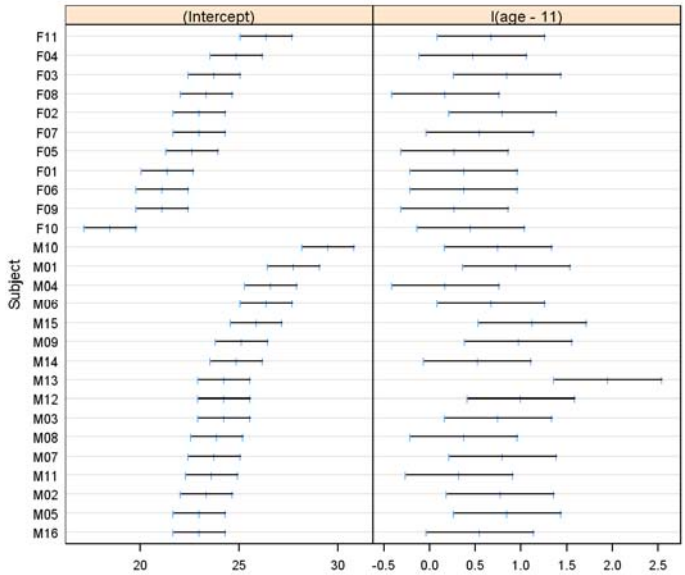
```
> pairs(lmlist.1,id=0.01)
```



Obr. 51

Tato negativní korelace mezi oběma parametry vzniká proto, že hodnota průsečíku předpovídá hodnotu vysvětlované proměnné pro věk 0, který leží daleko mimo rozsah věku studovaných osob. Této korelace se proto můžeme zbavit (a dosáhnout tak větší informativnosti obou koeficientů) tím, že centrujeme hodnoty vysvětlující proměnné. Střední hodnotou proměnné *age* je 11 a model proto změníme následovně:

```
> lmlist.2<-update(lmlist.1,distance~I(age-11))
> plot(intervals(lmlist.2))
```



Obr. 52

Vidíme (Obr. 52), že vzájemná závislost průsečíků a sklonů přímek vymizela (čtenář si to může ověřit také použitím funkce *pairs*), a můžeme se proto soustředit na informaci, která nás v tomto diagramu především zajímá. Ačkoliv se odhady sklonu přímky (v pravé části Obr. 52) liší, přeci jen se jejich konfidenční intervaly pro většinu pozorovaných osob značně překrývají. Jinak je tomu ale u průsečíku, tj. absolutního členu regresního

modelu. Odlišnost mezi jedinci je pro něj natolik velká, že tomuto pevnému parametru musí odpovídat v našem LME modelu také parametr s náhodným efektem. K otázce, zda by i sklonu přímky měl odpovídat náhodný efekt osoby, se vrátíme později.

Jednoduchý LME model

Náš první LME model nafitujeme takto:

```
> lme.1<-lme(distance~I(age-11), random=~1|Subject, data=Or)
```

Zadání vypadá podobně jako v případě klasického lineárního modelu (který vytváříme pomocí funkce *lm*), hlavní odlišnost je v parametru *random*, který popisuje strukturu náhodných efektů. Parametr *random* musí být přítomen v každém volání funkce *lme*. Pokud jej nepoužijeme, musí být objekt předávaný v parametru *data* typu *groupedData* a funkce *lme* si pak hodnotu parametru *random* odvodí ze vzorce uloženého spolu s daty. V našem příkladě používáme pro parametr *random* jednu z jednodušších podob a způsob zadání nám říká, že náhodný efekt bude modelován jen pro průsečík (proto je před svislou čárou hodnota *1*, nikoliv název proměnné) a že povaha náhodného efektu (jakých skupin pozorování se týká) je dána hodnotami faktoru *Subject*.

Podívejme se nyní na to, jak je nafitovaný LME model shrnut funkcí *summary*:

```
> summary(lme.1)
Linear mixed-effects model fit by REML
 Data: Or
       AIC      BIC    logLik
  455.0025 465.6563 -223.5013

Random effects:
 Formula: ~1 | Subject
        (Intercept) Residual
StdDev:    2.114724 1.431592


Fixed effects: distance ~ I(age - 11)
                Value Std.Error DF  t-value p-value
(Intercept) 24.023148 0.4296605 80 55.91193       0
I(age - 11)  0.660185 0.0616059 80 10.71626       0

 Correlation:
            (Intr)
I(age - 11) 0

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
-3.66453932 -0.53507984 -0.01289591  0.48742859  3.72178465

Number of Observations: 108
Number of Groups: 27
```

Ve výstupu stojí za pozornost hned první řádka, která nám sděluje, že LME model byl fitován s použitím metody REML (restricted maximum likelihood). Alternativní metodou je maximum likelihood (zadáme parametrem *type="ML"*), ale tato metoda často podceňuje velikost náhodných efektů. Musíme ji ale použít, pokud chceme porovnávat dva nebo více modelů, pokud se liší svými pevnými efekty.

Výstup funkce *summary* také udává hodnoty AIC, BIC a logaritmu dosažené věrohodnosti (likelihood).

Pak následuje část popisující nafitované náhodné efekty: prvý z nich (Intercept) představuje variabilitu mezi jedinci v hodnotě průsečíku přímkové závislosti, druhý (Residual) pak představuje nevysvětlenou variabilitu na úrovni jednotlivých měření. Odpovídá tedy residuálnímu střednímu čtverci (residual mean square) klasického regresního modelu.

Pevné efekty jsou popsány v tabulce způsobem, na který jsme již zvyklí z výsledků klasických lineárních modelů. V následující tabulce korelací mezi odhady regresních koeficientů si všimneme, že v důsledku centrování prediktoru *age* jsme dosáhli nulové korelace obou odhadů.

Poslední zajímavou informací, kterou nám funkce *summary* poskytne pro *lme* objekt, je počet skupin pozorování (v našem případě tedy jedinců), které v datech rozpoznala.

Na rozdíl od klasického regresního modelu nebo ANOVA modelu můžeme konfidenční intervaly pro odhady parametrů LME modelu získat pohodlně pomocí funkce *intervals*. Implicitní pokrytí těmito intervaly (interval coverage) je 0.95 (tj. 95%), ukážeme si ale, jak zadat odlišnou hodnotu, například 99% intervaly:

```
> intervals(lme.1,level=0.99)
Approximate 99% confidence intervals

 Fixed effects:
                 lower       est.      upper
(Intercept) 22.8894071 24.0231481 25.1568892
I(age - 11)  0.4976262  0.6601852  0.8227441
attr(,"label")
[1] "Fixed effects:"

 Random Effects:
  Level: Subject
                   lower     est.    upper
sd((Intercept)) 1.419220 2.114724 3.151067

 Within-group standard error:
   lower     est.    upper
1.167818 1.431592 1.754944
```

Každý parametr (jak s pevným, tak s náhodným efektem) je popsán třemi odhady – dolní hranicí, vlastním odhadem parametru a horní hranicí intervalu spolehlivosti.

Testy náhodných efektů

Náš jednoduchý LME model obsahuje jen jeden parametr s náhodným efektem. Chceme-li otestovat jeho významnost, musíme jej porovnávat s klasickým lineárním modelem, tj. se shodnými pevnými efekty jako má *lme.1*, ale bez parametru s náhodným efektem:

```
> lm.1<-lm(distance~I(age-11),data=Or)
> anova.lme(lm.1,lme.1)
      Model df      AIC      BIC    logLik   Test  L.Ratio p-value
lm.1      1  3 515.1695 523.1598 -254.5848
lme.1     2  4 455.0025 465.6563 -223.5013 1 vs 2 62.16698  <.0001
```

Vhodnou metodu pro funkci *anova* jsme si explicitně vyžádali (použitím názvu *anova.lme*) proto, že jazyk S určuje typ metody podle prvého parametru, a ten byl v tomto případě objektem třídy *lm*. Z výsledné tabulky vidíme, že jak z porovnání úspornosti (parsimony), tak z likelihood-ratio testu vyplývá, že přidání náhodného efektu výrazně zvýšilo kvalitu modelu. Vraťme se ještě k naší dřívější myšlence, a to zda by nebylo vhodné předpokládat variabilitu mezi jedinci i pokud jde o růstovou rychlost (tj. sklon nafitované regresní přímky). Řešení bude analogické naším dřívějším postupům při hledání správné podoby regresního modelu:

```
> lme.2<-update(lme.1,random=~I(age-11)|Subject)
> anova(lme.1,lme.2)
      Model df      AIC      BIC    logLik   Test L.Ratio p-value
lme.1     1  4 455.0025 465.6563 -223.5013
lme.2     2  6 454.6367 470.6173 -221.3183 1 vs 2 4.36583  0.1127
```

Výpověď dvou různých charakteristik parsimony (AIC a BIC) je protichůdná, ale protože ani likelihood-ratio test není příliš přesvědčivý, zůstaneme u modelu jednoduššího (*lme.1*).

Testy parametrů s pevným efektem

Dosud jsme se nezabývali rozdílem v růstu lebky mezi chlapci a dívkami. Tento rozdíl se, podobně jako u dosud diskutovaných náhodných efektů, může projevovat buď jen v odlišné průměrné velikosti lebky mezi oběma pohlavími (v takovém případě půjde o dvě paralelně běžící přímky) nebo i v dynamice růstu lebky (pak by měly tyto dvě přímky odlišný sklon). Začněme první, jednodušší možností:

```
> lme.3<-update(lme.1,.~.+Sex)
> anova(lme.2,lme.3)
      Model df      AIC      BIC    logLik   Test  L.Ratio p-value
lme.2     1  6 454.6367 470.6173 -221.3183
lme.3     2  5 447.5125 460.7823 -218.7562 1 vs 2 5.124178  0.0236
Warning message:
Fitted objects with different fixed effects. REML comparisons are not
meaningful in: anova.lme(lme.2, lme.3)
>
```

Varovná zpráva na konci výstupu funkce *anova* je důležitá. Jak jsem zmiňoval již výše, nelze odhady věrohodnosti, získané metodou REML, porovnávat mezi modely lišícími se strukturou pevných efektů. Musíme proto oba porovnávané modely přefitovat, s použitím metody maximum likelihood:

```
> lme.2ml<-update(lme.2,method="ML")
> lme.3ml<-update(lme.3,method="ML")
> anova(lme.2ml,lme.3ml)
        Model df      AIC      BIC    logLik   Test  L.Ratio p-value
lme.2ml     1  6 451.2116 467.3044 -219.6058
lme.3ml     2  5 444.8565 458.2671 -217.4282 1 vs 2 4.355116  0.0369
```

Závěr se nám sice nezměnil (je průkazný rozdíl mezi chlapci a děvčaty ve velikosti lebky), nicméně výrazné zvýšení pravděpodobnosti chyby I. druhu nám ukazuje, že na použití správné metody si musíme dávat pozor.

Je mezi chlapci a dívkami rozdíl také v růstové rychlosti?

```
> lme.4ml<-update(lme.3ml,.~Sex*I(age-11))
```

```
> anova(lme.3ml,lme.4ml)
        Model df      AIC      BIC    logLik   Test  L.Ratio p-value
lme.3ml     1  5 444.8565 458.2671 -217.4282
lme.4ml     2  6 440.6391 456.7318 -214.3195 1 vs 2 6.217427  0.0126
```

Rozdíl zde zjevně je, jeho směr zjistíme takto:

```
> fixef(lme.4ml)
          (Intercept)             SexFemale           I(age - 11)
           24.9687500            -2.3210227             0.7843750
SexFemale:I(age - 11)
           -0.3048295
```

Lebky dívek jsou tedy nejen celkově menší (o *2.32*), ale i jejich růst s věkem je pomalejší (roční přírůstek je menší o *0.305*).

Zobrazení LME modelu

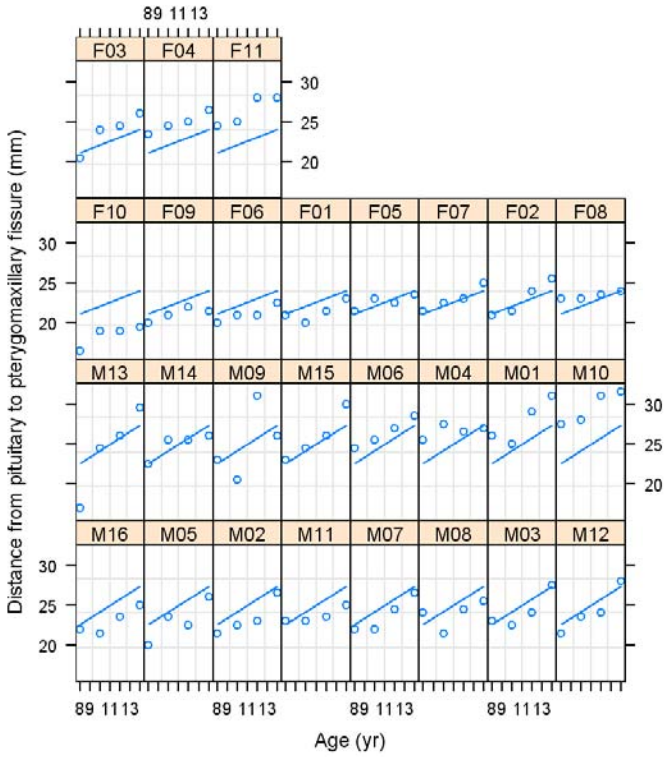
Jak si nafitovaný model zobrazíme? Pokud použijeme (na základě výsledků z předchozí sekce) model *lme.4ml*, měli bychom jej nejprve znovu nafitovat metodou REML:

```
> lme.4<-update(lme.4ml, method="REML")
```

Vzhledem k tomu, že v LME modelu se kombinují dva typy parametrů – s pevným a s náhodným efektem – může být pro čtenáře zajímavé si graficky porovnat vlivy těchto dvou skupin parametrů:

```
> plot(augPred(lme.4,level=0,length.out=2),grid=T,aspect=2)
```

Predikce hodnot vysvětlované proměnné je prováděna funkcí *augPred* na hladině struktury náhodných efektů dané parametrem *level*. Pokud nejsou pro daný model náhodné efekty v sobě hierarchicky vnořeny, např. v našem modelu, připadá v úvahu jen hodnota 1 – kdy jsou náhodné efekty patřící k faktoru *Subject* zahrnuty do vypočtených fitovaných hodnot – nebo hodnota 0, kdy jsou náhodné efekty ignorovány. Ostatní parametry použité ve funkci *augPred* resp. *plot* ovlivňují jen vzhledové vlastnosti obrázku (Obr. 53).

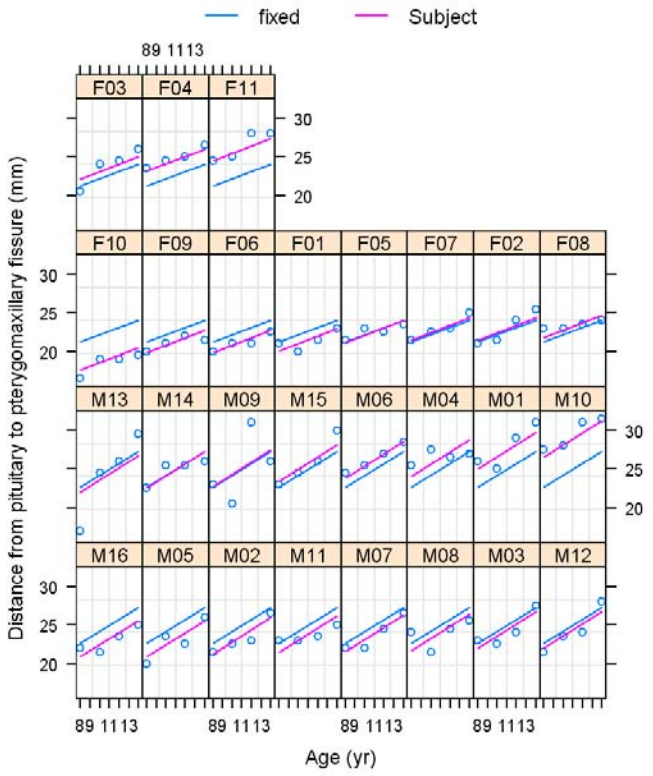


Obr. 53

V obrázku vidíme, že jsou data popsána dvěmi odlišně vysoko umístěnými přímkami (toto umístění je dáno hlavním efektem faktoru *Sex*), které se také liší svým sklonem, menším pro dívky (dáno interakcí mezi faktorem *Sex* a proměnnou *age*). Můžeme si ale vynést také diagram, ve kterém je vidět, o kolik lépe fituje pozorovaná data plný LME model, tj. model zahrnující i náhodný efekt jedinců:

```
> plot(augPred(lme.4,level=0:1,length.out=2),grid=T,aspect=2,layout=c(8,4))
```

Výsledek je v Obr. 54.



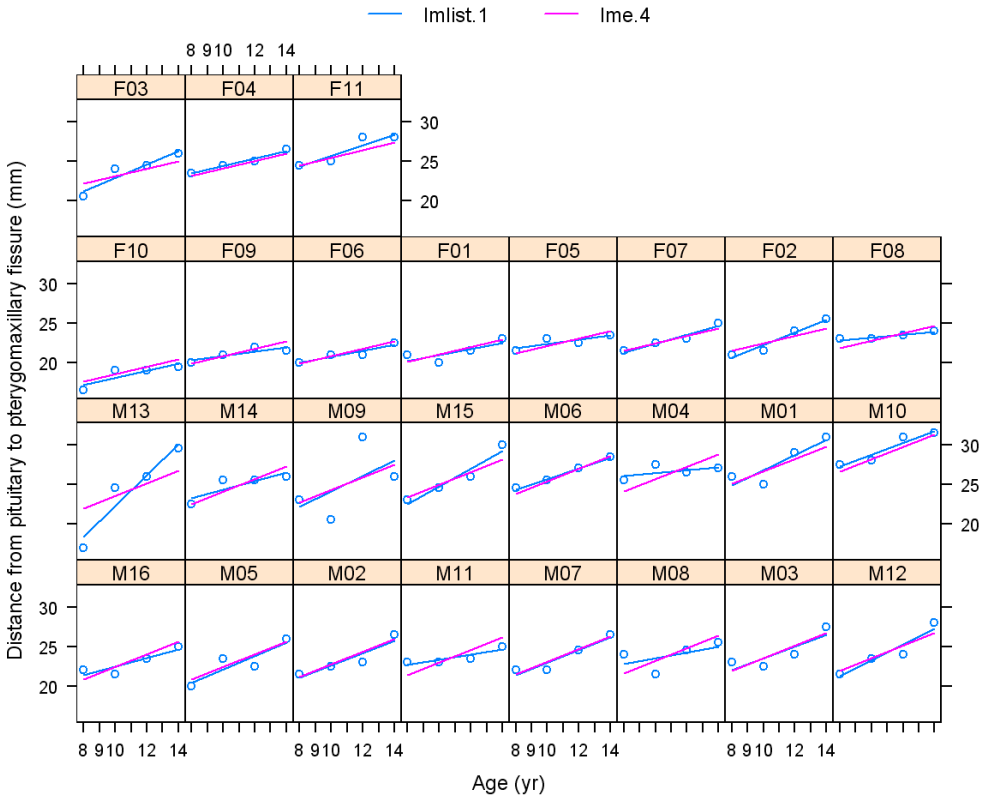
Obr. 54

Z toho, že jsou přímky zahrnující náhodný efekt (červené) rovnoběžné s přímkami založenými jen na pevných efektech, je také vidět, že se náhodný efekt týkal jen průsečíku přímky.

Pro pochopení toho, že náš LME model stojí někde mezi klasickým lineárním modelem a postupem, kdy je fitován samostatný lineární model pro každou skupinu pozorování (tj. zde pro každý *Subject*), může být užitečné porovnat náš model *lme.4* také s dřívějším *lmlist.1* modelem. Použijeme k tomu funkci *comparePred*:

```
> plot(comparePred(lmlist.1,lme.4,length.out=2),layout=c(8,4))
```

Výsledek (který vidíme v Obr. 55) nám ukazuje, že model *lme.4* u některých jedinců skutečnou růstovou rychlost podceňuje (např. u chlapce M13), zatímco u jiných ji nadhodnocuje (například M04 nebo M11). Pro většinu případů si ale sklony přímek dobře odpovídají, jak vyplývá i z testu, který použití náhodného efektu pro sklon závislosti na věku zamítl.



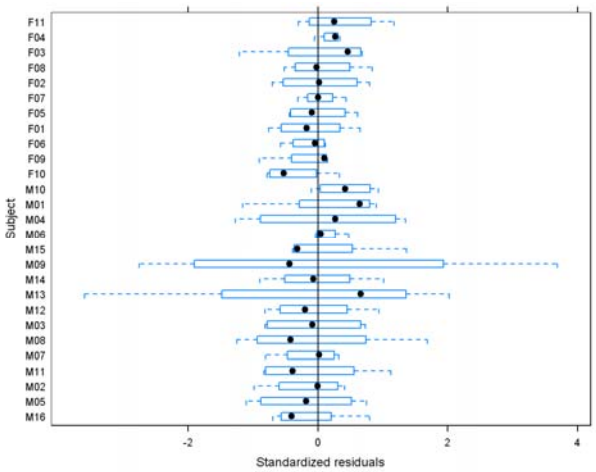
Obr. 55

Residuály a modelování variability

Podobně jako v klasických regresních modelech bychom i zde neměli zapomenout na regresní diagnostiku. V situaci, kdy náhodnou variabilitu modelujeme pomocí faktorů s náhodným efektem, je její správný popis v našem modelu naopak ještě důležitější.

Podívejme se nejdříve na distribuci residuálů, jak se projevuje uvnitř skupin pozorování, odpovídajících konkrétním jedincům:

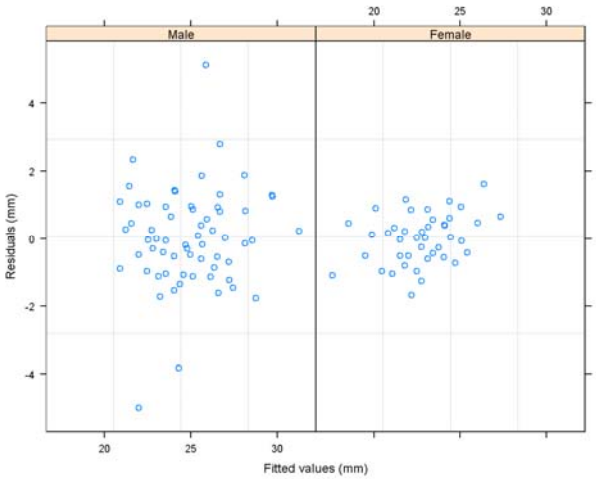
```
> plot(lme.4,Subject~resid(.,type="p"),abline=0)
```



Obr. 56

V grafu (Obr. 56) vidíme rozdíly nejen mezi jednotlivými osobami, ale také obecnější trend nižší variability mezi děvčaty, ve srovnání s variabilitou mezi chlapci. To není, zdá se, způsobeno jen těmi několika případy, kdy se růstová rychlost výrazně odlišovala od společného modelu. Jde spíše o obecnější případ nehomogenity variancí při porovnání chlapců a dívek a lépe to uvidíme pokud stejné standardizované residuály (standardizaci zajišťuje parametr *type="p"* ve volání funkce *resid*) vyneseme proti fitovaným hodnotám, zvlášť pro chlapce a pro dívky:

```
> plot(lme.4,resid(.)~fitted(.)|Sex)
```



Obr. 57

Tento poměrně jednoduchý případ, ve kterém odhadujeme residuální variabilitu zvlášť pro dvě skupiny definované faktorem *Sex*, můžeme popsat v LME modelu snadno:

```
> lme.5<-update(lme.4, weights=varIdent(form=~1|Sex))
```

Odhady pevných efektů se nezměnily:

```
> fixef(lme.4)
          (Intercept)             SexFemale          I(age - 11)
           24.9687500            -2.3210227              0.7843750
```

```
SexFemale:I(age - 11)
           -0.3048295
> fixef(lme.5)
          (Intercept)             SexFemale           I(age - 11)
           24.9687500            -2.3210227             0.7843750
SexFemale:I(age - 11)
           -0.3048295
```

a můžeme tedy pochybovat, zda tato změna měla nějaký vliv na kvalitu modelu. Přesně na tuto otázku odpovídá likelihood-ratio test:

```
> anova(lme.4,lme.5)
      Model df      AIC      BIC    logLik   Test  L.Ratio p-value
lme.4     1  6 445.7572 461.6236 -216.8786
lme.5     2  7 429.2205 447.7312 -207.6102 1 vs 2 18.53677  <.0001
```

Jeho výsledek jasně ukazuje, že modelování nevysvětlené variability zvlášť pro chlapce a pro dívky bylo dobrou volbou. Ještě se podívejme na informaci, o kterou se rozšířil text vypisovaný funkcí *summary:*

```
> summary(lme.5)

...
Variance function:
 Structure: Different standard deviations per stratum
 Formula: ~1 | Sex
 Parameter estimates:
     Male    Female
1.0000000 0.4678944

...
```

Variance (odmocněná, tj. prezentovaná jako směrodatná odchylka) v první skupině (v tomto případě chlapci) má referenční hodnotu 1.0, pro ostatní skupiny (zde skupina dívky) jsou uváděny relativní hodnoty. Nevysvětlená variabilita je tedy zhruba poloviční (46.8%) u děvčat ve srovnání s chlapci.

Nelineární závislosti

Nelineární modely se smíšenými efekty užíváme nejčastěji v těch situacích, kdy potřebujeme daty, která představují změny veličiny v čase, proložit nějakou křivku, která není ve svých parametrech lineární a jejíž volba je dána zvyklostmi v určitém vědním oboru. Pro popis růstu populace můžeme použít logistické modely různé složitosti, při studiu enzymatické kinetiky rovnici Michaelise a Mentenové, v oboru farmako-kinetiky například kompartmentový model prvního řádu, pokud jde např. o lék podávaný ústy, vstřebávající se do krevního oběhu a vychytávaný jedním orgánem.

My si zde ukážeme jen nejjednodušší použití, na příkladu modelování změny rychlosti fotosyntézy s rostoucí koncentrací oxidu uhličitého (zde tedy čas jako explicitní vysvětlující proměnná nevystupuje). Od určité koncentrace se fotosyntetická rychlost již nezvyšuje, protože se přiblížila asymptotické hodnotě. Různé podmínky mohou změnit jak dynamiku nárůstu této rychlosti, tak limitní hodnotu (asymptotu). Tyto vlivy můžeme obvykle popsat faktory s pevným efektem, nicméně objevují se i rozdíly mezi studovanými jedinci, které se týkají různých parametrů fitované křivky, a tyto rozdíly je lepší popsat jako náhodné efekty.

Data, která zde budeme používat, jsou měření prováděná na 12 jedincích C_4 trávy *Echinochloa crus-galli*. Protože cílem bylo studovat a porovnat efekty adaptace a aklimace, 6 rostlin pocházelo z oblasti Quebecu, zatímco zbylých 6 z jihu USA (Mississippi). Polovina z každé šestice byla kontrolní, rostoucí při 26 °C, zatímco druhá polovina byly rostliny vystavené chladu (chilled) –teplotě 7 °C po dobu 14 hodin. Na každé rostlině bylo provedeno měření spotřeby oxidu uhličitého při sedmi různých koncentracích CO_2. Tato koncentrace je tedy naším primárním prediktorem.

```
>
> summary(CO2)
     Plant             Type         Treatment       conc          uptake
 Qn1    : 7   Quebec     :42   nonchilled:42   Min.   :  95   Min.   : 7.70
 Qn2    : 7   Mississippi:42   chilled   :42   1st Qu.: 175   1st Qu.:17.90
 Qn3    : 7                                    Median : 350   Median :28.30
 Qc1    : 7                                    Mean   : 435   Mean   :27.21
 Qc3    : 7                                    3rd Qu.: 675   3rd Qu.:37.13
 Qc2    : 7                                    Max.   :1000   Max.   :45.50
 (Other):42
```

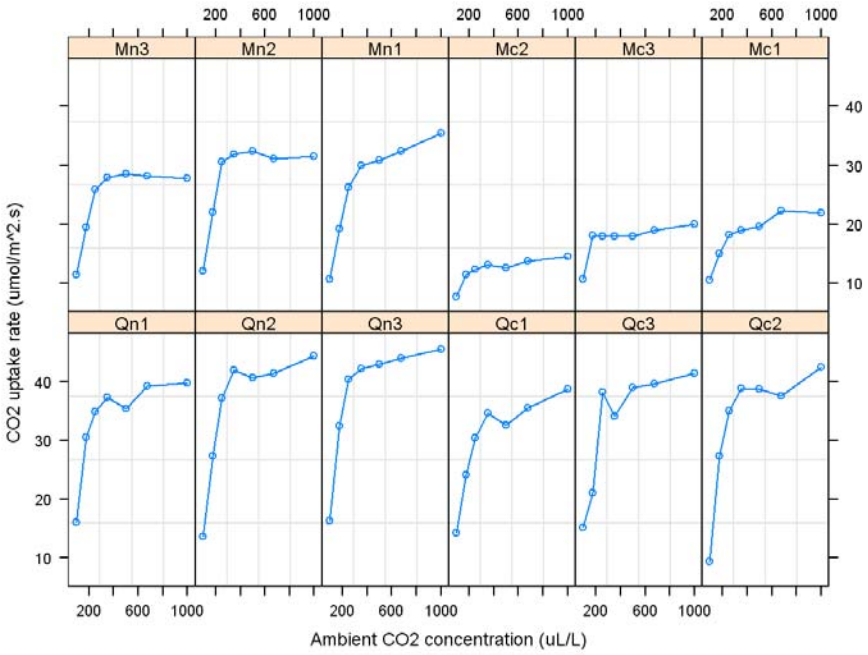
Na rozdíl od našich předchozích dat je objekt *CO2* "obyčejný" datový rámec, není typu *groupedData*. Velmi by se nám ale hodilo, aby ho měl, proto si vytvoříme novou verzi:

```
> co2.gd<-groupedData( uptake~conc|Plant, data=CO2, FUN=mean,
+  outer = ~ Type*Treatment,
+  labels = list( x = "Ambient CO2 concentration", y = "CO2 uptake rate"),
+  units = list( x = "(uL/L)", y="(umol/m^2.s)"))
```

Data si nyní můžeme poměrně snadno vynést:

```
> plot(co2.gd,aspect=2)
```

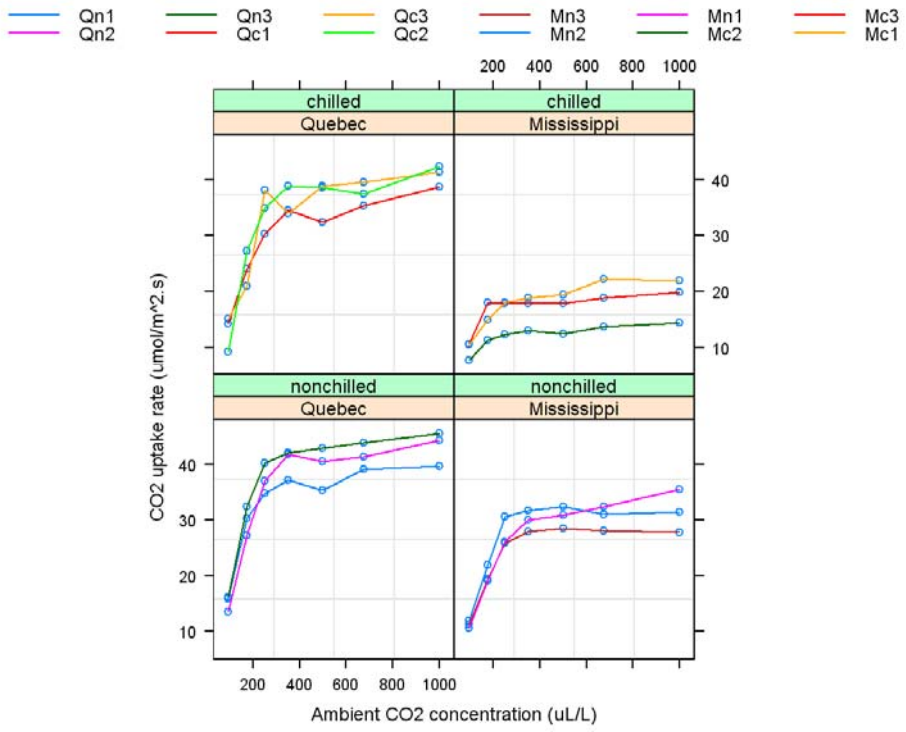
Výsledný diagram (Obr. 58) ukazuje křivky pro jednotlivé experimentální rostliny, jejich pořadí uvnitř každé ze čtyř skupin (definovaných kombinacemi hodnot dvou "outer" proměnných, *Type* a *Treatment*) je dáno rostoucí průměrnou hodnotou rychlosti příjmu.



Obr. 58

Pokud bychom se chtěli zaměřit na rozdíly mezi křivkami pro rostliny patřící do různých skupin, můžeme obrázek změnit pomocí parametru *outer*:

```
> plot(co2.gd,aspect=1,outer=T)
```



Obr. 59

Rostliny jsou v diagramu (Obr. 59) seskupeny podle původu (Quebec v levém sloupci a Mississippi v pravém) a podle experimentálního zásahu (*nonchilled* v dolním řádku a *chilled* v horním). Vidíme i bez fitování modelu, že vliv ochlazení je výraznější u rostlin jižního původu.

Model asymptotického růstu

Fitování modelu začneme volbou správného typu křivky. Obvykle (ale ne nutně) rovnice těchto křivek popisují závislost vysvětlované proměnné na jedné (primární) vysvětlující proměnné. Obvykle to bývá čas, v našem případě je to ale ambientní koncentrace CO2. Každá taková rovnice obsahuje dva nebo více parametrů, jejichž hodnoty musí být odhadnuty. Takovouto křivku můžeme proložit daty pro každou skupinu pozorování (typicky to znamená pro každého jedince) nebo naopak lze fitovat společnou křivku pro všechna pozorování. Celý postup ale začne být zajímavý v okamžiku, kdy definujeme, které z parametrů rovnice se mohou lišit mezi jedinci, a to buď díky náhodné variabilitě (náhodné efekty) nebo díky tomu, že se jedinci liší v hodnotách dalších vysvětlujících proměnných s pevnými efekty. V tom druhém případě popíšeme vliv dalších prediktorů tak, že hodnoty parametrů nelineárního modelu definujeme jako lineární kombinaci těchto dalších vysvětlujících proměnných.

Začněme ale pro naše data nejprve se základní nelineární rovnicí. Růst rychlosti příjmu CO_2 popíšeme tzv. asymptotickou křivkou s posunem (asymptotic regression with an offset)[38]. Package *nlme* nám již nabízí pro tuto i jiné nelineární rovnice předdefinované funkce, které se označují jako self-starting, protože jsou z dat schopny určit počáteční odhady pro své parametry. Pokud bychom si zvolili typ rovnice, která není v package *nlme* předdefinována, museli bychom pro fitování dodat počáteční hodnoty odhadů.

My ale máme štěstí, můžeme použít předdefinovanou funkci *SSasympOff*, která má celkem tři parametry:

$$y = \phi_1(1 - e^{-e^{\phi_2}(x-\phi_3)})$$

Parametr ϕ_1 představuje asymptotickou hodnotu vysvětlované proměnné, ke které fitovaná křivka roste. Parametr ϕ_2 představuje logaritmus rychlosti růstu y s hodnotou x a parametr ϕ_3 je hodnotou x, ve které začne hodnota y růst (tj. do této hodnoty je, pro naše data, rychlost příjmu oxidu uhličitého nulová – nebo negativní).

Výběr náhodných efektů pro NLME model

Podobně jako pro LME model se i zde rozhodneme, který ze tří parametrů modelu vykazuje výraznou variabilitu mezi jedinci, tak, že nafitujeme asymptotickou křivku pro každého měřeného jedince zvlášť, pomocí funkce *nlsList*.

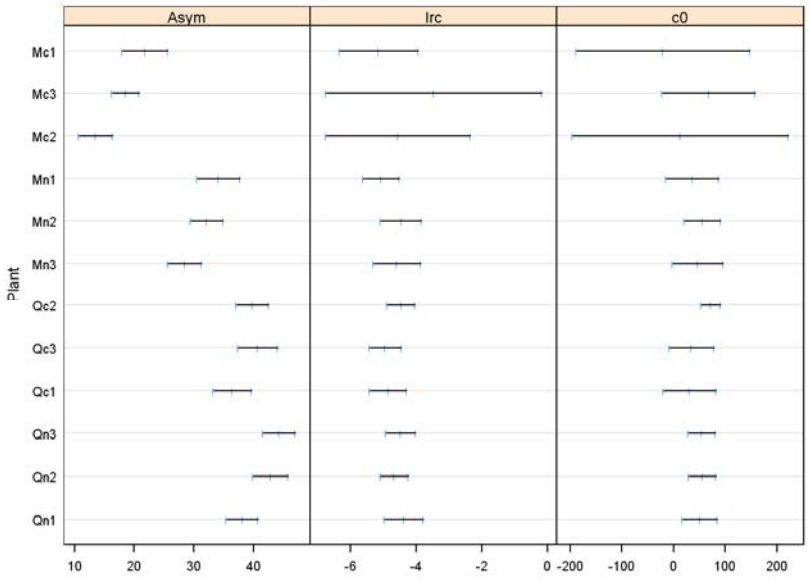
```
> nlslist.1<-nlsList(uptake~SSasympOff(conc, Asym, lrc, c0)|Plant,data=co2.gd)

> nlslist.1
Call:
```

[38] Posun zde vyjadřuje skutečnost, že pozitivní hodnota fotosyntézy nenastává při nulové, ale až při vyšší koncentraci CO_2.

```
  Model: uptake ~ SSasympOff(conc, Asym, lrc, c0) | Plant
   Data: co2.gd

Coefficients:
        Asym       lrc        c0
Qn1 38.13978 -4.380647  51.22324
Qn2 42.87169 -4.665728  55.85816
...
Mc3 18.53506 -3.465158  67.84877
Mc1 21.78723 -5.142256 -20.39998
Degrees of freedom: 84 total; 48 residual
Residual standard error: 1.79822
```

Podobně jako u LME, i zde ale získáme lepší představu vynesením konfidenčních intervalů pro jednotlivé parametry:

```
> plot(intervals(nlslist.1,level=0.99))
```



Obr. 60

V Obr. 60 vidíme, že nejvýraznější odlišnosti jsou mezi rostlinami v parametru *Asym*, méně přesvědčivé pak ve zbylých dvou parametrech. Měli bychom ovšem vzít v úvahu i to, že odchylky mezi jedinci pocházejícími z odlišných experimentálních skupin mohou být dány pevnými efekty experimentálního zásahu a/nebo původu rostliny, případně jejich interakcí. Pro začátek nafitujeme základní model s náhodným efektem jen v parametru pozice asymptoty (tj. parametr Asym = ϕ_1).

V implementaci funkce *nlme* v rámci programu R je nejjednodušší definovat první NLME model na základě výsledku vráceného funkcí *nlsList*:

```
> nlme.1<-nlme(nlslist.1,random=Asym~1)
```

Přestože by si self-starting funkce měla určit počáteční hodnoty parametrů sama, není tomu tak, a tak musíme funkci *nlme*, která začíná "z ničeho", předat i počáteční odhady pro hodnotami parametrů *fixed* a *random* (nic nám neodpustí ☺). To by vypadalo takto:

```
> nlme.1<-nlme(uptake~SSasympOff(conc,Asym,lrc,c0),data=co2.gd,
```

```
+  fixed=Asym+lrc+c0~1,random=Asym~1,
+ start=c(Asym=30,lrc=-5,c0=50))
```

Počáteční hodnoty, předané jako parametr *start*, jsem zjistil odečtením průměrných hodnot z Obr. 60.

Parametr *fixed* nám ve své výše uvedené podobě říká, že všechny tři parametry ϕ_i asymptotické křivky mají být konstanty, které nezávisí na žádné další vysvětlující proměnné s pevným efektem, a hodnota parametru *random* zase určuje, že náhodný efekt jedince předpokládáme pouze u křivkového parametru *Asym*.

Podívejme se ještě, zda by nebylo vhodné přidat náhodný efekt i k parametru *lrc*, tj. logaritmu rychlosti, se kterou se fotosyntetická rychlost zvyšuje s rostoucí koncentrací CO2:

```
> nlme.2<-update(nlme.1,random=Asym+lrc~1)
> anova(nlme.1,nlme.2)
       Model df      AIC      BIC    logLik   Test  L.Ratio p-value
nlme.1     1  5 422.3691 434.5232 -206.1846
nlme.2     2  7 419.5166 436.5323 -202.7583 1 vs 2 6.852543  0.0325
```

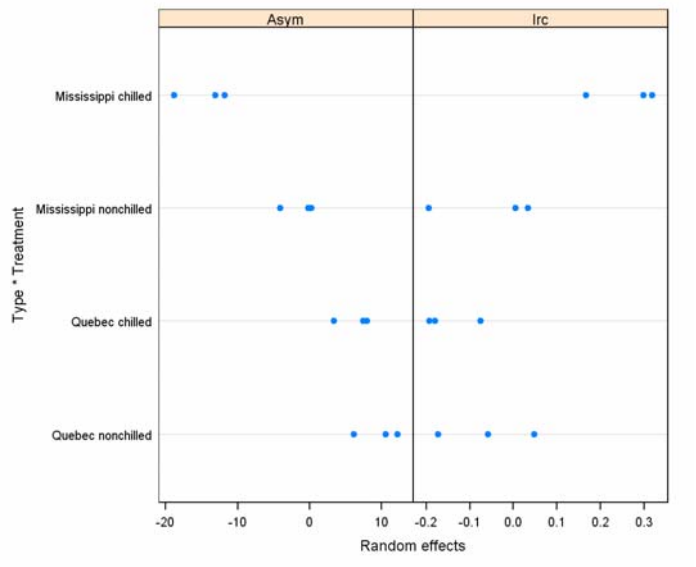
Rozdíl není nijak impresivní, ale je průkazný, a tak zůstaneme u nového, složitějšího modelu *nlme.2*.

Modelování pevných efektů u NLME modelu

Nejprve se musíme rozhodnout, které z našich dvou vysvětlujících proměnných budou ovlivňovat který ze tří parametrů asymptotické křivky. Při rozhodování nám nejspíše pomůže následující diagram:

```
> nlme.2re<-ranef(nlme.2,augFrame=T)
> plot(nlme.2re,form=~Type*Treatment)
```

Parametr *form* zajistil, že jsou v Obr. 61 vynášeny hodnoty náhodných efektů (pro dva parametry asymptotické křivky) do společných linií pro všechny tři rostliny, sdílející hodnoty faktorů *Type* a *Treatment*; bez něj by měla každá rostlina samostatnou horizontální linii. Funkce *plot* našla hodnoty *Type* a *Treatment* pro každé pozorování v objektu vráceném funkcí *ranef* – ta normálně vrací jen vlastní odhady náhodných efektů, ale argument *augFrame* zajistil, že sem byly ze zdrojových dat překopírovány i hodnoty vysvětlujících proměnných.



Obr. 61

Z diagramu vidíme, že se odchylky jednotlivých rostlin od průměrných hodnot parametrů liší především podle geografického původu, nicméně i efekt chladu je výrazný, zejména pro rostliny z Mississippi. To naznačuje existenci interakce mezi faktory *Type* a *Treatment*. Zkusíme si proto rovnou nafitovat model s pevnými, kombinovanými efekty obou faktorů na parametry *Asym* a *lrc*:

```
> nlme.3<-update(nlme.2,fixed=list(Asym+lrc~Type*Treatment,c0~1))
Error in nlme.formula(model = uptake ~ SSasympOff(conc, Asym, lrc, c0),  :
        starting values for the fixed component are not the correct length
```

Jenže to nefunguje – počáteční hodnoty jsou k dispozici jen pro tři původní parametry, a nyní nám pro každý parametr, který vysvětlujeme hlavními efekty vektorů *Type* a *Treatment* (a jejich interakcí), přibyly tři další parametry (oba faktory mají totiž jen dvě hladiny). Pro tyto nové parametry (vsunuté za jim odpovídající původní parametry) můžeme použít počáteční hodnotu nulovou (představující absenci vlivu faktorů):

```
> fixef(nlme.2)
     Asym       lrc        c0
32.411849 -4.560327 49.342305
> nlme.3<-update(nlme.2,fixed=list(Asym+lrc~Type*Treatment,c0~1),
+ start=c(32.412,0,0,0, -4.56,0,0,0,49.34))
> fixef(nlme.3)
                      Asym.(Intercept)
                             41.8175806
                   Asym.TypeMississippi
                            -10.5304916
                 Asym.Treatmentchilled
                             -2.9694093
Asym.TypeMississippi:Treatmentchilled
                            -10.8992725
                       lrc.(Intercept)
                             -4.5572679
                    lrc.TypeMississippi
                             -0.1041054
                  lrc.Treatmentchilled
                             -0.1712432
 lrc.TypeMississippi:Treatmentchilled
```

```
 0.7412328
        c0
50.5076531
```

Tentokrát to tedy fungovalo a z výstupu funkce *fixef* vidíme, že např. hodnota asymptoty je pro kontrolní rostliny z Quebecu rovna *41.818*, zatímco pro rostliny z Mississippi je obecně od *10.53* nižší, a třeba pro chlazené rostliny z Mississippi je nižší o dalších (*2.969+10.899*). Velikosti pevných efektů jsou méně výrazné pro parametr *lrc*, jejich průkaznosti můžeme prozkoumat pomocí funkce *anova*:

```
> anova(nlme.3)
                    numDF denDF   F-value p-value
Asym.(Intercept)        1    64  1905.529  <.0001
Asym.Type               1    64   221.172  <.0001
Asym.Treatment          1    64    47.254  <.0001
Asym.Type:Treatment     1    64   128.299  <.0001
lrc.(Intercept)         1    64 12972.224  <.0001
lrc.Type                1    64     0.359  0.5513
lrc.Treatment           1    64     0.000  0.9870
lrc.Type:Treatment      1    64     3.042  0.0859
c0                      1    64   133.897  <.0001
```

Výsledky pro *lrc* parametr asymptotické rovnice ukazují, že se jeho hodnota obecně neliší mezi oběma geografickými oblastmi ani mezi chlazenými a kontrolními rostlinami. To je také dobře vidět pokud budeme společně testovat jen hlavní efekty obou faktorů:

```
> anova(nlme.3,Terms=c(6,7))
F-test for: lrc.Type, lrc.Treatment
  numDF denDF  F-value p-value
1     2    64 1.188622  0.3113
```

Naproti tomu interakce je průkazná, pokud ji testujeme společně s hlavním efektem kteréhokoliv z obou faktorů, například:

```
> anova(nlme.3,Terms=c(6,8))
F-test for: lrc.Type, lrc.Type:Treatment
  numDF denDF  F-value p-value
1     2    64 6.297781  0.0032
```

To odpovídá obsahu Obr. 61, ve kterém se hodnoty *lrc* odlišují jen pro ochlazované rostliny z oblasti Mississippi.

Zobrazení nafitovaného NLME modelu

Ještě si ukážeme, jak fitovaný model zobrazit. Postup je obdobný jako u LME modelů, opět použijeme funkci *augPred* pro získání fitovaných hodnot s a bez náhodných efektů. Čeká nás ale jeden technický problém:

```
> plot(augPred(nlme.3,level=0:1))
Error in predict.nlme(object, value[1:(nrow(value)/nL), , drop = FALSE],  :
        Levels Quebec,Mississippi not allowed for Type
```

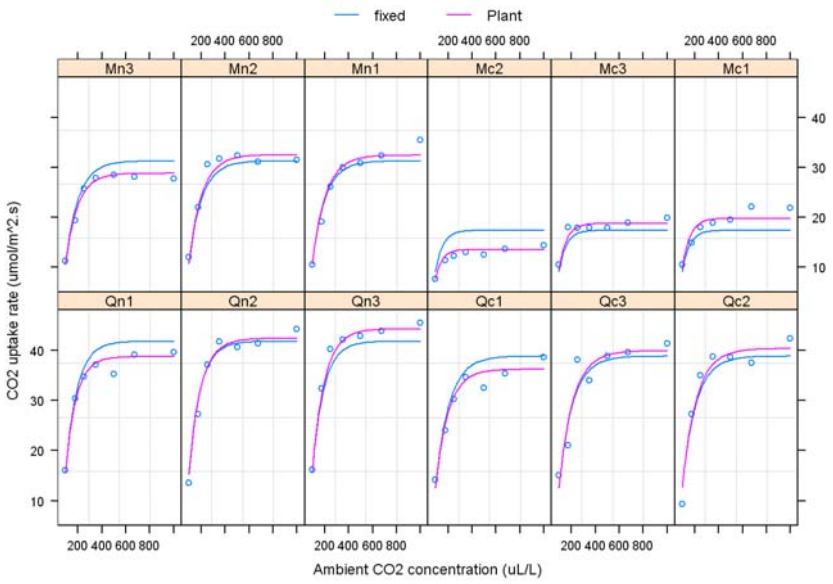
Jde o známou chybu v implementaci package *nlme* v programu R, která nebyla za několik let opravena (nicméně je známo, jak ji obejít). Musíme se vrátit k volání funkce *update*, která nám nafitovala model *nlme.3* a uzavřít odkazy na faktory *Type* a *Treatment* do volání funkce *factor*:

```
> nlme.3<-update(nlme.2,fixed=list(Asym+lrc~factor(Type)*factor(Treatment),
+  c0~1), start=c( 32.412, 0, 0, 0, -4.56, 0, 0, 0, 49.34))
```

Teď nám již funkce *plot* bude fungovat (přidáme jí ještě dva parametry, ovlivňující vzhled panelů i jejich uspořádání):

```
> plot(augPred(nlme.3,level=0:1),grid=T,layout=c(6,2))
```

A zde je výsledný diagram:



Obr. 62

Zobecněné LME modely

Také LME modely lze zobecnit stejným způsobem jako klasické lineární, tj. volbou typu distribuce pro náhodnou variabilitu a také vhodné link funkce. To je např. výhodné, pokud chceme modely zahrnující náhodné efekty používat pro binární vysvětlovanou proměnnou. Konkrétní příklad uvádět nebudu, ale čtenáři prozradím, že potřebná funkce se jmenuje *glmmPQL*, je součástí package *MASS* a ve svém použití se od funkce *lme* liší jen v tom, že v ní přibyl navíc parametr *family*, který se zde užívá stejně, jako ve funkci *glm*.

9 Práce s částečně závislými údaji: fylogenetická korekce a bodová uspořádání

V předchozí kapitole jsme si ukázali, že jednotlivá pozorování ve skupinách definovaných v lineárním modelu se smíšenými efekty jsou spolu korelována. Buď můžeme považovat tuto jejich korelaci za symetrickou (tj. stejnou pro všechna pozorování ve skupině) nebo můžeme její konkrétní podobu přesněji modelovat – např. jako časovou autokorelaci nebo korelaci prostorovou.

Velmi často ale potřebujeme analyzovat data, která představují jen jednu velkou "skupinu" pozorování, tj. vztah částečné závislosti je mezi všemi našimi pozorováními. Pokud tuto částečnou závislost ignorujeme, můžeme použít standardní statistické metody, naše závěry o testovaných hypotézách ale budou povětšinou příliš optimistické (tj. skutečná pravděpodobnost chyby prvního druhu bude obecně vyšší, než je náš odhad).

U biologických dat se nejčastěji setkáváme se dvěma speciálními případy korelovaných dat – fylogenetické závislosti mezi taxony porovnávanými v rámci srovnávacích studií a prostorové korelace mezi pozorováními.

Fylogenetická závislost – motivační příklad

Biologické srovnávací studie porovnávají taxony s různou mírou "příbuznosti". Ačkoliv bychom tuto příbuznost ideálně mohli popsat fylogenetickým stromem se spolehlivými odhady časů divergence jednotlivých vývojových linií, v praxi máme jen nepříliš spolehlivé odhady, někdy ani to ne. Základní postupy metod fylogenetické korekce ve srovnávacích studiích proto popíšeme na příkladu, ve kterém je naše znalost evoluční spřízněnosti jednotlivých taxonů chabá. Jde o příkladová data z package *ape*, popisující různé charakteristiky jednotivých druhů šelem:

```
> library(ape)
> summary(carnivora)
       Order         SuperFamily         Family        Genus
 Carnivora:112   Caniformia:57   Viverridae :32   Mustela : 9
                 Feliformia:55   Mustelidae :30   Herpetes: 8
                                 Felidae    :19   Panthera: 5
                                 Canidae    :18   Canis   : 4
                                 Hyaenidae  : 4   Martes  : 4
                                 Procyonidae: 4   Felis   : 3
                                 (Other)    : 5   (Other) :79
                   Species          FW               SW               FB
 Acinonyx jubatus       :  1   Min.   :  0.050   Min.   :  0.050   Min.   :  1.00
 Ailuropoda melanoleuca :  1   1st Qu.:  1.245   1st Qu.:  1.400   1st Qu.: 15.25
 Alopex lagopus         :  1   Median :  3.400   Median :  3.895   Median : 33.00
 Aonyx capensis         :  1   Mean   : 18.099   Mean   : 20.084   Mean   : 53.40
 Arctictis binturong    :  1   3rd Qu.: 10.363   3rd Qu.: 11.592   3rd Qu.: 57.38
 Arctogalidia trivirgata:  1   Max.   :320.000   Max.   :365.000   Max.   :365.00
 (Other)                :106
       SB               LS              GL               BW               WA
 Min.   :  1.00   Min.   :1.000   Min.   : 23.50   Min.   :   0.01   Min.   : 21.0
 1st Qu.: 15.68   1st Qu.:2.500   1st Qu.: 53.80   1st Qu.:  41.88   1st Qu.: 54.5
 Median : 33.75   Median :3.000   Median : 63.00   Median : 116.25   Median : 70.0
 Mean   : 56.43   Mean   :3.232   Mean   : 65.79   Mean   : 249.31   Mean   :104.0
 3rd Qu.: 57.17   3rd Qu.:3.800   3rd Qu.: 73.50   3rd Qu.: 286.88   3rd Qu.:117.0
 Max.   :459.50   Max.   :8.800   Max.   :168.00   Max.   :1650.00   Max.   :730.0
                  NA's   :2.000   NA's   : 21.00   NA's   :  50.00   NA's   : 49.0
...
```

Naše informace o evoluční spřízněnosti jednotlivých druhů se zde omezují na jejich zařazení do klasických taxonomických jednotek (*Order*, *SuperFamily*,...).

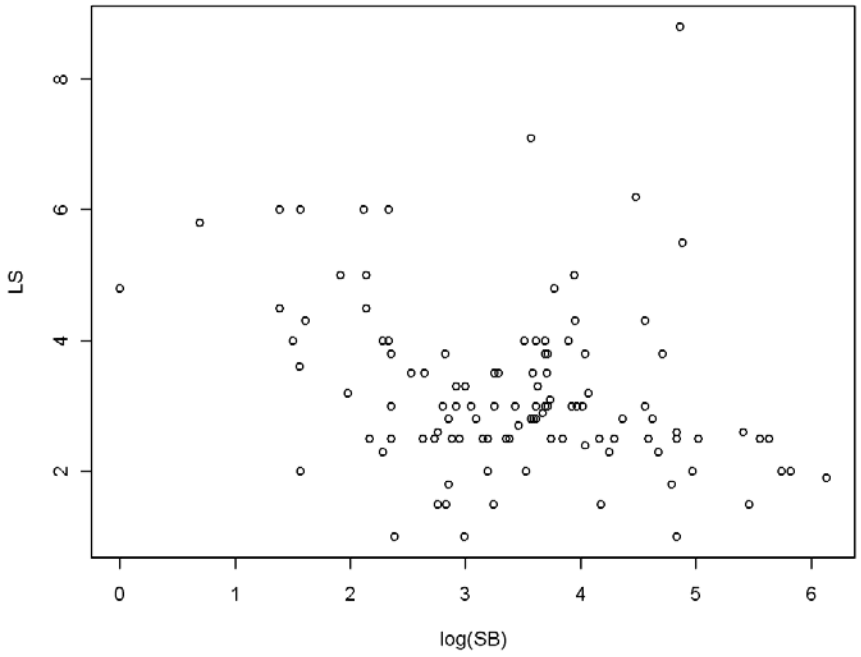
Ptáme se na to, zda se průměrná velikost vrhu (proměnná *LS*) mění podle průměrné velikosti mozku (proměnná *SB*). Nejprve můžeme vztah těchto dvou proměnných zhodnotit graficky:

```
> plot(LS~SB,data=carnivora)
```

Diagram (zde vynechán) i úvaha[39] nám naznačují, že by náš prediktor (*SB*) měl být používán na logaritmické škále. Podobně můžeme uvažovat i o vysvětlované proměnné, ale vzhledem k omezenému rozsahu hodnot velikosti vrhu zde logaritmická transformace asi není nutná[40]:

```
> plot(LS~log(SB),data=carnivora)
```

Výsledný diagram (Obr. 63) ukazuje obecný pokles velikosti vrhu s rostoucí velikostí mozku.



Obr. 63

Stejný závěr by nám dala i analýza, ve které bychom příbuznost mezi druhy ignorovali:

```
> summary(lm(LS~log(SB),data=carnivora))
...
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   4.3392     0.3819  11.362  < 2e-16 ***
log(SB)      -0.3229     0.1057  -3.055  0.00283 **

...
```

[39] Nebudu se asi ptát "jak se velikost vrhu změní, když se hmotnost mozku zvýší o 5 g?". Spíš mne bude zajímat, jaký efekt má zvýšení hmotnosti o například 10%.
[40] tj. to, zda transformuji nebo ne, příliš neovlivní výsledné modely.

```
Multiple R-Squared: 0.07955,    Adjusted R-squared: 0.07103
F-statistic: 9.334 on 1 and 108 DF,  p-value: 0.002834
```

Výsledek nám naznačuje průkazný ($p<0.003$) pokles velikosti vrhu s rostoucí hmotností mozku.

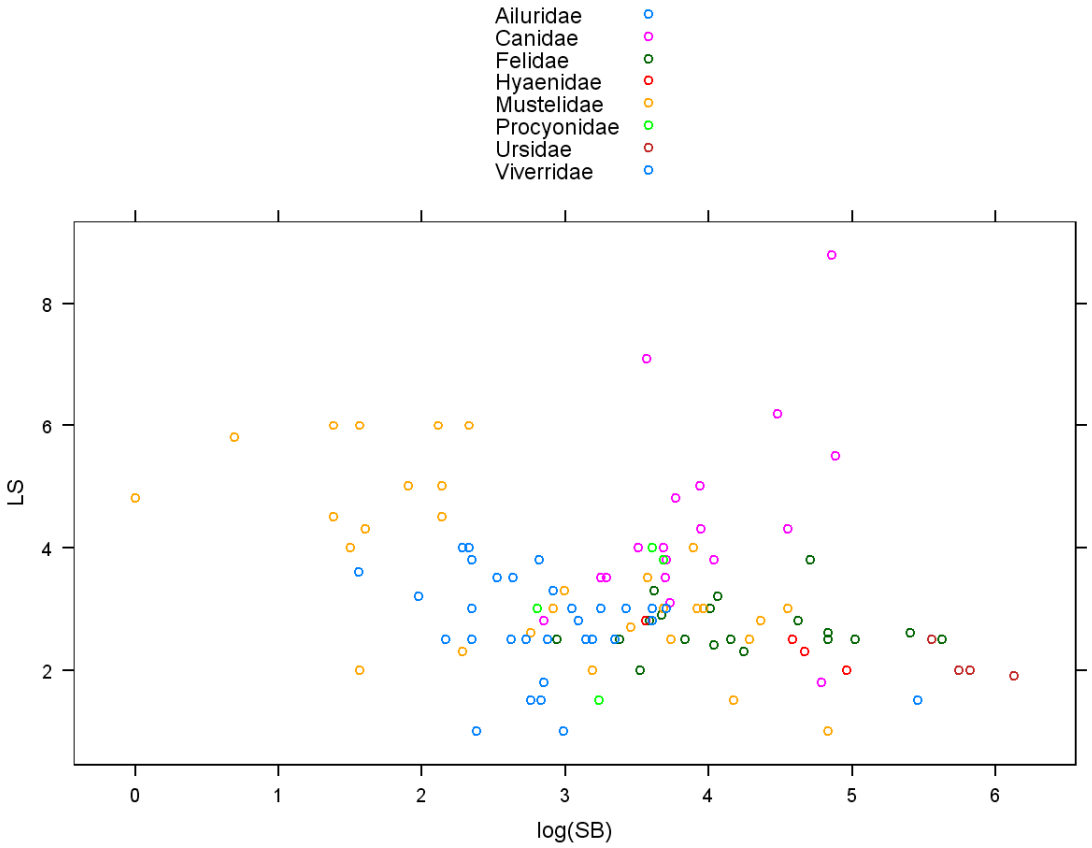
Evoluční setrvačnost

Jenomže ignorovat spřízněnost studovaných taxonů bychom neměli. Protože je v našem souboru větší počet blízce příbuzných druhů, je docela dobře možné, že podobné hodnoty velikosti mozku či počtu mláďat ve vrhu měl i jejich společný předek a porovnávané znaky vlastně ani "neměly čas" se mezi příbuznými taxony rozrůznit (tzv. evoluční setrvačnost, evolutionary inertia). Z tohoto pohledu mohou být mnohé z porovnávaných taxonů považovány za pseudo-replikace, tj. nepředstavují nezávislé příspěvky (důkazy) pro testovanou hypotézu.

V případě, kdy studujeme vztah mezi charakteristikami druhů a vlastnostmi prostředí, ve kterém tyto druhy žijí, se kromě evoluční setrvačnosti můžeme setkávat i s tzv. nikovou setrvačností, často nazývanou i nikový konservatismus (niche conservatism). Jestliže společný předek dvou nebo více druhů žil v určitém prostředí, pak je docela pravděpodobné, že i z něj vyvinuté (evolučně se postupně odlišující) druhy zůstaly v daném typu prostředí nebo se přesunuly do prostředí svými vlastnostmi podobného. Tento efekt může působit nepřímo i v našich datech – pokud je například velikost vrhu ovlivňována vlastnostmi prostředí, ve kterém daný druh šelmy žije.

Podobnost hodnot pro příbuzné druhy je vidět i na vyšších taxonomických úrovních, například v čeledích, jak ukazuje Obr. 64 vytvořený následujícím kódem:

```
> library(lattice)
> xyplot(LS~log(SB),groups=Family,data=carnivora, auto.key=TRUE)
```



Obr. 64

Variabilitu v hodnotách vysvětlovaných i vysvětlujících proměnných, kterou lze vysvětlit evoluční setrvačností, bychom měli odstranit (zohlednit) dříve, než tyto proměnné použijeme v našich modelech. Toho lze dosáhnout různými postupy, z nichž alespoň některé (v tomto textu zmiňované) jsou víceméně ekvivalentní, všechny ale vyžadují popis vztahů mezi jednotlivými taxony, tj. odhad jejich evoluční spřízněnosti. Tyto údaje lze nejlépe odvodit z fylogenetického stromu popisujícího hypotézy vývoje aktuálně žijících (a v našem datovém souboru zahrnutých) taxonů ze společných předků.

Fylogenetický strom

Pro vytváření fylogenetických stromů ze známé informace o jednotlivých taxonech (v dnešní době obvykle ze sekvence bazí nebo sekvence aminokyselin v proteinech) existuje velká paleta postupů a specializovaných programů a také v rámci programu R existují funkce (např. v package *ape*) vhodné ke konstrukci a porovnávání fylogenetických stromů. Nicméně toto téma nepatří do oblasti moderních regresních metod a proto budu předpokládat, že čtenář má takový strom již k dispozici nebo si vytvoří jeho nedokonalou náhražku z hierarchické taxonomické informace způsobem popsaným níže.

Fylogenetický strom, který chceme použít ke "korekci" našeho statistického modelu (často se tento postup označuje jako fylogenetická korekce, phylogenetic correction), musí mít nejen definovanou strukturu větvení, ale i odhadnuté délky jednotlivých větví.

Tyto délky by měly představovat dobu, která uplynula od okamžiku, ve kterém začal společný předek (představovaný uzlem – nodem, ve které se daná větev spojuje s jinou) divergovat do dvou nebo více nezávislých vývojových linií. Z pohledu metod fylogenetické korekce ale není ideálem odhad absolutního času (např. miliony let ode dneška), protože rychlost evolučních změn se mohla v různých obdobích měnit, například s měnícími se vlastnostmi prostředí, ve kterých se druhy vyvíjely[41]. Podstata odhadů délek větví (založených na míře odlišnosti sekvencí bazí či aminokyselin) ale naštěstí vyhovuje předpokladu, že délka větví odpovídá míře, ve které se vlastnosti organismů mohly za dané období změnit.[42]

Pokud máme strom zkonstruovaný ze sekvenčních informací (nebo alespoň z morfometrických údajů – nemělo by jít ale o stejné údaje, které zamýšlíme analyzovat pomocí fylogeneticky korigovaných modelů!), budeme mít jeho definici uloženou nejspíše v tzv. Newick-ově formátu[43], který vypadá například takto:

```
(
  SangMino:30,
  ( PoteArge:5,
    PoteErec:5
  ):25
);
```

Výše uvedený text popisuje fylogenetický ministrom (příliš malý pro reálné analýzy), ve kterém jsou propojeny tři taxony, z nichž dva (z rodu ***Pote****ntilla*) se oddělily ze společného předka před 5 časovými jednotkami, a společný předek, kterého sdílely se třetím druhem (*Sanguisorba minor*, také z čeledi růžovitých), žil zhruba před 30 časovými jednotkami. Soubor s takto popsaným fylogenetickým stromem načteme do programu R (pokud jsme již zpřístupnili package *ape*) pomocí funkce *read.tree*. Soubory popisující fylogenetický stromeček pomocí alternativních formátů lze načíst funkcemi *read.caic* nebo *read.nexus*.

V případě našich dat ale fylogenetický strom popisující vztahy mezi jednotlivými druhy šelem nemáme k dispozici a naší jedinou možností bude spoléhat se na klasickou hierarchickou taxonomii, i když ta může být v rozporu s fylogenetickými principy. Na taxonomii založený strom vytvoříme takto:

```
> carn.tree<-as.phylo(~SuperFamily/Family/Genus/Species,data=carnivora)
```

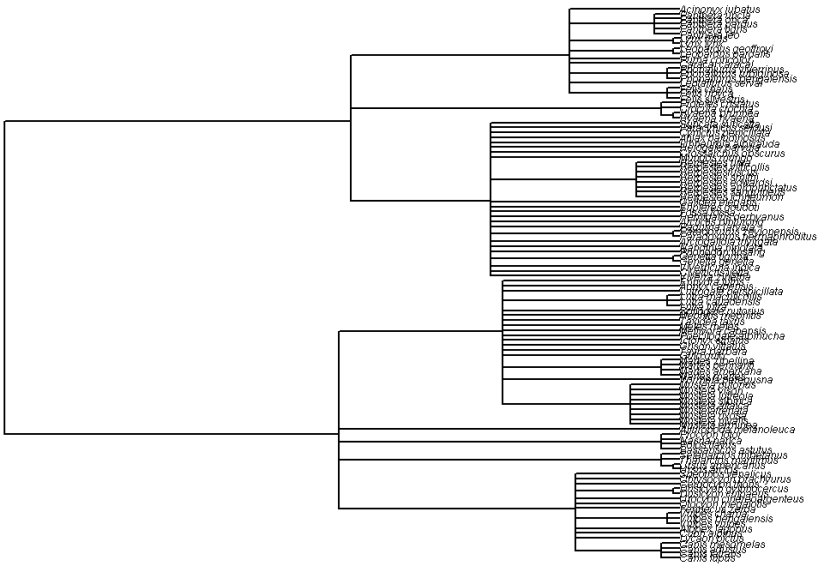
Obsah takto vytvořeného stromu nám vyjadřuje vpodstatě jen skutečnost, že druhy patřící do jednoho rodu si jsou blíže než druhy z rodu jiného, a podobně že druhy ze stejné čeledi (či nadčeledi) jsou si podobnější, i když méně než druhy stejného rodu. S obsahem stromu se můžeme blíže seznámit pomocí funkcí *summary* a *plot*:

```
> summary(carn.tree)
Phylogenetic tree: carn.tree
```

[41] Bohužel je docela dobře možné, že tato variabilita v rychlosti vývoje je různá pro různé, v modelech porovnávané znaky, ale musíme se v tomto případě smířit s absencí přesnějších informací.
[42] Proti tomuto požadavku ovšem stojí obvyklý metodický požadavek, aby byl příslušný strom ultrametrický, tj. aby součet délek větví od určitého koncového bodu (tj. aktuálně žijícího druhu) k libovolnému předku (včetně kořene fylogenetického stromu) byl pro všechny koncové body stejný.
[43] Newick nebyl fylogenetický badatel, ale zakladatel restaurace, ve které byly dohodnuty základy tohoto formátu.

```
  Number of tips: 112
  Number of nodes: 26
  No branch lengths.
  No root edge.
  First ten tip labels: Canis lupus
                        Canis latrans
                        Canis adustus
                        Canis mesomelas
                        Lycaon pictus
                        Cuon alpinus
                        Alopex lagopus
                        Vulpes vulpes
                        Vulpes bengalensis
                        Vulpes chama
  No node labels.
> plot(carn.tree,cex=0.5)
```

Všimněme si nejprve, že strom má sice 112 koncových bodů (špiček – tips), odpovídajících 112 srovnávaným druhům šelem, nicméně jen 26 vnitřních nód (míst větvení). To je důsledkem skutečnosti, že mnohá větvení jsou polytomická – všechny větve odpovídající druhům jednoho rodu vycházejí ze stejného bodu, podobně pro všechny rody z dané čeledi, atd. Pro plně dichotomicky větvený strom je počet vnitřních uzlů roven počtu koncových bodů zmenšenému o jednotku (tj. byl by 111 v našem příkladě). Dále nás funkce *summary* informuje, že pro tento strom nejsou známy délky větví. Funkce *plot* ale nemůže strom bez znalosti délek nakreslit, a tak si délky sama "vymyslela". Používá k tomu tzv. Grafenovu metodu, pomocí které délky větví vypočteme níže sami. Vzdálenost větvení od koncových bodů (na pravé straně diagramu v Obr. 65) je v této metodě určena počtem koncových bodů, které daný uzel nese. To je – v případě absence doplňujících znalostí – asi nejlepší možné řešení[44].



Obr. 65

[44] odpovídá předpokladu, že taxony byly z dané skupiny vybírány náhodně, všechny mají stejnou hodnotu a četnost divergencí byla v průběhu evoluce konstantní

Abychom přidali takto "vypočtené" ad hoc délky do objektu, popisujícího náš "fylogenetický" strom, použijeme funkci *compute.brlen*:

```
> carn.tree.2<-compute.brlen(carn.tree)
```

Nyní můžeme takto vytvořený strom použít v některé z vícero existujících metod fylogenetické korekce. Zde si ukážeme nejjednodušší způsoby použití tří z nich – zobecněných nejmenších čtverců (generalized least squares, GLS), nezávislých kontrastů (obvykle nazývaných phylogenetically independent contrasts, PIC) a Desdevisovy metody – existuje jich ale větší množství.

Metoda GLS

Metoda GLS je obecnější, nejde o specifickou metodu jen pro práci s fylogenetickými daty. Představuje zobecnění klasické lineární metody, ve kterém se může jednak lišit přesnost (spolehlivost) hodnot jednotlivých pozorování (tato vlastnost GLS se při fylogenetické korekci obvykle nepoužívá), jednak (a to je právě zde důležité) můžeme v GLS modelech popsat korelaci (kovarianci) mezi jednotlivými pozorováními. V package *ape* jsou tyto modely nabízeny ve více obecné podobě, nazývané GEE (generalized estimating equations), ve které je možné zvolit i typy distribuce a link funkce, podobně jako ve zobecněných lineárních modelech. My ale budeme funkci *compar.gee* používat v té nejjednodušší formě:

```
> compar.gee(LS~log(SB),data=carnivora,phy=carn.tree.2)
Error in compar.gee(LS ~ log(SB), data = carnivora, phy = carn.tree.2) :
     the present method cannot (yet) be used directly with missing data: you
may consider removing the species with missing data from your tree with the
function `drop.tip'.
In addition: Warning message:
the rownames of the data.frame and the names of the tip labels do not match:
```

...

Funkce *compar.gee* odmítá pracovat, vysvětluje nám ale, v čem je problém: v porovnávaných proměnných (konkrétně v proměnné *LS*, představující velikost vrhu, viz výstup z funkce *summary* na začátku kapitoly) máme chybějící hodnoty. Musíme tedy jak z fylogenetického stromu, tak z datového rámce s proměnnými *LS* a *SB*, odstranit ty druhy, pro které údaje o *LS* chybí. Nejprve zjistíme, jaké indexy mají druhy s chybějícími údaji, a pak tyto indexy vynecháme při tvorbě nového datového rámce a nového stromu:

```
> (1:112)[is.na(carnivora$LS)]
[1] 63 70
> carnivora.2<-carnivora[c(-63,-70),]
> carn.tree.3<-drop.tip(carn.tree.2, c(63,70))
```

Nyní již můžeme fitovat GLS model. Funkce *compar.gee* vyprodukuje výstup, který je poměrně málo informativní a zde je vynechán[45].

```
> cg.1<-compar.gee(LS~log(SB), data=carnivora.2, phy=carn.tree.3)
```

...

```
> cg.1
Call:
```

[45] Ukazuje, že autoři package *ape* pilně bádají a nemají čas uhlazovat detaily ☺

```
  formula: LS ~ log(SB)
Number of observations:  110

Model:
 Link:                       identity
 Variance to Mean Relation: gaussian

Summary of Residuals:
       Min         1Q     Median         3Q        Max
-1.8625505 -0.3380243  0.1662743  0.9956809  5.9365567

Coefficients:
             Estimate       S.E.         t Pr(T > |t|)
(Intercept) 2.70570320 0.80144164 3.3760452 0.003482129
log(SB)     0.03245807 0.06668408 0.4867439 0.632493886

Estimated Scale Parameter:  1.933533
"Phylogenetic" df (dfP):  19.45946
```

Primárním výsledkem jsou odhady regresních koeficientů, zejména sklonu přímky popisující závislost *LS* na logaritmu *SB*. Po fylogenetické korekci není vztah velikosti vrhu k velikosti mozku průkazný. Za zmínku stojí i poslední řádek s hodnotou statistiky *dfP*, která představuje odhad počtu stupňů volnosti, zohledňující vzájemnou závislost mezi jednotlivými pozorováními. Tento odhad je založen výlučně na struktuře vývojového stromu, který metoda používala.

Metoda PIC

Jde asi o nejvíce používanou metodu fylogenetické korekce, kterou lze použít především pro kvantitativní data. Původních *n* (v našem případě 110) pozorování pro každou z porovnávaných proměnných[46] je nahrazeno *n-1* kontrasty – rozdíly mezi hodnotami, předpovídanými pro dva taxony (nejen srovnávané druhy, ale i hypotetické společné předky, odpovídající vnitřním nódům fylogenetického stromu), jejichž větve se ve stromu bezprostředně spojují. Tyto rozdíly jsou dále standardizovány očekávanou variabilitou jejich odhadů, a ta je vypočtena z délek větví. Vzhledem ke způsobu výpočtu kontrastů jako rozdílu dvou hodnot je nezbytné, aby byl použitý strom plně dichotomický (tj. v jednom nodu se spojovaly vždy dvě větve). Že to není náš případ uvidíme hned při prvním pokusu o výpočet PIC:

```
> LS.pic<-pic(carnivora.2$LS,carn.tree.3)
Error in pic(carnivora.2$LS, carn.tree.3) :
        "phy" is not fully dichotomous
```

Funkce *multi2di* arbitrárně "rozštěpí" polytomické větvení[47] na sérii dichotomických, přičemž délka uměle vytvořených větví (oddělující jednotlivé dichotomie v sérii) je nastavena na nulovou hodnotu.

```
> LS.pic<-pic(carnivora.2$LS,multi2di(carn.tree.3))
> logSB.pic<-pic(log(carnivora.2$SB),multi2di(carn.tree.3))
```

[46] Je důležité si uvědomit, že PIC metoda "nezávisle" (ovšem s použitím téhož fylogenetického stromu) "přežvýká" každou z proměnných tak, aby se daly použít v nějaké standardním (nejspíše lineárním) modelu.

[47] v našich datech jde např. o skupiny více druhů ze stejného rodu

V případě velikosti mozku tedy opět používáme logaritmované hodnoty. Při užití kontrastů je důležité pamatovat na to, že jde o rozdíly původních hodnot (a také hodnot vypočtených pro hypotetické předky, představované vnitřními nódy stromu) a že odchylku od nuly v hodnotě vysvětlované proměnné korelujeme s odchylkou v hodnotách proměnné (proměnných) vysvětlujících. Je proto nezbytné, abychom v našich modelech neměli absolutní člen, nulová hodnota kontrastu vysvětlující proměnné pak bude odpovídat nulové hodnotě kontrastu proměnné vysvětlované.[48]

```
> summary(lm(LS.pic~logSB.pic-1))

...

Coefficients:
          Estimate Std. Error t value Pr(>|t|)
logSB.pic -0.02237    0.15322  -0.146    0.884

Residual standard error: 4.619 on 108 degrees of freedom
Multiple R-Squared: 0.0001973,  Adjusted R-squared: -0.00906
F-statistic: 0.02132 on 1 and 108 DF,  p-value: 0.8842
```

Závěr získaný metodou PIC je tedy shodný s tím, který jsme získali metodou GLS: velikost mozku nemá po fylogenetické korekci vliv na průměrnou velikost vrhu jednotlivých druhů šelem.

Desdevisova metoda

Metoda, kterou publikovali Desdevises et al. v roce 2003, opět vychází z informace obsažené ve fylogenetickém stromu. Tuto informaci ale transformuje do podoby proměnných, které v modelu představují vliv fylogenetické spřízněnosti mezi jednotlivými pozorováními (taxony) na hodnoty vysvětlované i vysvětlujících proměnných. Obvykle tedy představují tyto proměnné v našem modelu kovariáty. Na druhou stranu ale také můžeme zaměřit naši pozornost přímo na ně a studovat, například, v jaké míře se vysvětlující schopnost evoluční minulosti překrývá s vlivem prostředí, protože takový překryv může být s jistou opatrností[49] interpretován jako nikový konservatismus.

Postup, kterým jsou z fylogenetického stromu tyto "fylogenetické proměnné" vytvářeny, je poměrně jednoduchý. Informace o evoluční vzdálenosti jednotlivých párů taxonů (tj. jak daleko je, měřeno délkou větví, ke společnému předku zvolené dvojice taxonů) je převedena na souřadnice jednotlivých taxonů na osách analýzy hlavních koordinát (principal coordinate analysis, PCO či PCoA). Problém ale nastává s tím, že obecně potřebujeme k převodu vzdáleností mezi *n* taxony celkem *n-1* takových os a v takovém případě nám těchto *n-1* proměnných vysvětlí nezbytně veškerou variabilitu hodnot vysvětlované proměnné. Postup, který doporučují Desdevises et al., je vybrat jen ty fylogenetické prediktory, které mají průkazný vztah k vysvětlované proměnné. Výběr je

[48] Pokud bychom tedy testovali mnohorozměrnou hypotézu, s více vysvětlovanými proměnnými, například metodou RDA, neměli bychom vysvětlované proměnné centrovat, protože centrováním vlastně deklarujeme existenci absolutního členu, který takto z modelu odstraňujeme.
[49] protože i vlastnosti prostředí se v čase měnily

zjednodušen tím, že jednotlivé osy PCO jsou vzájemně nekorelované (lineárně nezávislé).

Nejprve tedy převedeme náš vývojový strom na matici vzdáleností:

```
> carn.dist<-cophenetic(carn.tree.3)
> dim(carn.dist)
[1] 110 110
```

Funkcí *dim* jsme si ověřili, že byla vytvořena symetrická čtvercová matice pro 110 druhů šelem. Faktorizaci metodou PCO můžeme provést pomocí funkce *cmdscale*:

```
> carn.pco<-cmdscale(carn.dist,k=109,eig=T)
Warning messages:
1: some of the first 109 eigenvalues are < 0 in: cmdscale(carn.dist, k = 109,
eig = T)
2: NaNs produced in: sqrt(ev)
```

To, že některá charakteristická čísla jsou záporná, není tak nečekané. To se stává v případě, že vzdálenosti v matici obsažené nejsou zcela metrické, ve smyslu Eukleidovského prostoru (není např. splněna podmínka tzv. trojúhelníkové nerovnosti). Takové vzdálenosti se nedají zcela přesně zobrazit v *n*-rozměrném eukleidovském prostoru a tento rozpor popisují osy se zápornými charakteristickými čísly (na těchto osách mají souřadnice jednotlivých taxonů komplexní hodnoty). Proto se při výběru průkazných hodnot omezíme na osy PCO, které mají kladná charakteristická čísla:

```
> names(carn.pco)
[1] "points" "eig"    "x"      "ac"     "GOF"
> carn.pco$eig
  [1]  8.122080e+04  8.695949e+03  2.013655e+03  2.013655e+03  2.013655e+03
  [6]  2.013655e+03  2.013655e+03  2.013655e+03  2.013655e+03  2.013655e+03
 [11]  2.013655e+03  2.013655e+03  2.013655e+03  2.013655e+03  2.013655e+03
 [16]  2.013655e+03  2.013655e+03  2.013655e+03  2.013655e+03  1.117095e+03
 [21]  1.065576e+03  1.065576e+03  1.065576e+03  8.359125e+02  2.946993e+01
 [26]  1.598831e+01  1.598831e+01  1.598831e+01  1.598831e+01  1.598831e+01
...
 [86]  6.492979e-04  1.623245e-04  1.623245e-04  0.000000e+00  0.000000e+00
 [91]  0.000000e+00 -3.637979e-12 -3.637979e-12 -3.637979e-12 -3.637979e-12
 [96] -3.637979e-12 -3.637979e-12 -3.637979e-12 -3.637979e-12 -1.091394e-11
[101] -1.091394e-11 -1.091394e-11 -1.091394e-11 -1.091394e-11 -1.455192e-11
[106] -1.455192e-11 -1.818989e-11 -1.818989e-11 -2.268348e+02
```

Vidíme, že poslední kladné (byť poměrně malé) charakteristické číslo odpovídá 88. ose. V tomto okamžiku nás z výsledků budou zajímat jen souřadnice bodů (taxonů), přesněji souřadnice na prvých 88 osách:

```
> carn.pco<-carn.pco$points
> dim(carn.pco)
[1] 110 109
> carn.pco<-carn.pco[,1:88]
> dim(carn.pco)
[1] 110  88
```

Postupný výběr signifikantních prediktorů z nabízených 88 os je poněkud zdlouhavý (automatizovaná metoda založená na hodnotě AIC je v případě velkého množství prediktorů zvláště liberální, vybírá jich zbytečně mnoho), takže je zde zobrazena jen jeho část:

```
> carn.pco.x<-data.frame(carn.pco)
```

```
> names(carn.pco.x)
 [1] "X1"  "X2"  "X3"  "X4"  "X5"  "X6"  "X7"  "X8"  "X9"  "X10" "X11" "X12"
[13] "X13" "X14" "X15" "X16" "X17" "X18" "X19" "X20" "X21" "X22" "X23" "X24"
[25] "X25" "X26" "X27" "X28" "X29" "X30" "X31" "X32" "X33" "X34" "X35" "X36"
[37] "X37" "X38" "X39" "X40" "X41" "X42" "X43" "X44" "X45" "X46" "X47" "X48"
[49] "X49" "X50" "X51" "X52" "X53" "X54" "X55" "X56" "X57" "X58" "X59" "X60"
[61] "X61" "X62" "X63" "X64" "X65" "X66" "X67" "X68" "X69" "X70" "X71" "X72"
[73] "X73" "X74" "X75" "X76" "X77" "X78" "X79" "X80" "X81" "X82" "X83" "X84"
[85] "X85" "X86" "X87" "X88"
> xnam<-paste("X",1:88,sep="")
> xnam
 [1] "X1"  "X2"  "X3"  "X4"  "X5"  "X6"  "X7"  "X8"  "X9"  "X10" "X11" "X12"
[13] "X13" "X14" "X15" "X16" "X17" "X18" "X19" "X20" "X21" "X22" "X23" "X24"
[25] "X25" "X26" "X27" "X28" "X29" "X30" "X31" "X32" "X33" "X34" "X35" "X36"
[37] "X37" "X38" "X39" "X40" "X41" "X42" "X43" "X44" "X45" "X46" "X47" "X48"
[49] "X49" "X50" "X51" "X52" "X53" "X54" "X55" "X56" "X57" "X58" "X59" "X60"
[61] "X61" "X62" "X63" "X64" "X65" "X66" "X67" "X68" "X69" "X70" "X71" "X72"
[73] "X73" "X74" "X75" "X76" "X77" "X78" "X79" "X80" "X81" "X82" "X83" "X84"
[85] "X85" "X86" "X87" "X88"
> carn.form<-as.formula(paste("~",paste(xnam,collapse="+")))
> carn.form
~X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8 + X9 + X10 + X11 + X12 +
    X13 + X14 + X15 + X16 + X17 + X18 + X19 + X20 + X21 + X22 +
    X23 + X24 + X25 + X26 + X27 + X28 + X29 + X30 + X31 + X32 +
    X33 + X34 + X35 + X36 + X37 + X38 + X39 + X40 + X41 + X42 +
    X43 + X44 + X45 + X46 + X47 + X48 + X49 + X50 + X51 + X52 +
    X53 + X54 + X55 + X56 + X57 + X58 + X59 + X60 + X61 + X62 +
    X63 + X64 + X65 + X66 + X67 + X68 + X69 + X70 + X71 + X72 +
    X73 + X74 + X75 + X76 + X77 + X78 + X79 + X80 + X81 + X82 +
    X83 + X84 + X85 + X86 + X87 + X88
```

Výše uvedené příkazy vytvořily vzorec, který obsahuje všech 88 potenciálních prediktorů a který můžeme použít během postupného výběru, ve funkci *add1*:

```
> lm.0<-lm(carnivora.2$LS~+1,data=carn.pco.x)
> add1(lm.0,carn.form,test="F")
Single term additions

Model:
carnivora.2$LS ~ +1
       Df Sum of Sq     RSS     AIC   F value     Pr(F)
<none>              186.759  60.227
X1      1    33.712 153.047  40.329   23.7894 3.723e-06 ***
X2      1     4.346 182.413  59.637    2.5730  0.111620
X3      1     6.687 180.072  58.216    4.0105  0.047723 *

...

X87     1     0.392 186.367  61.996    0.2270  0.634709
X88     1     0.033 186.725  62.207    0.0192  0.889948
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> lm.1<-update(lm.0,.~.+X1)
> add1(lm.1,carn.form,test="F")

...

> lm.2<-update(lm.1,.~.+X71)
> add1(lm.2,carn.form,test="F")

...
```

Postupné rozšiřování modelu dále pokračovalo až k modelu, pro který přidání jakéhokoliv dalšího prediktoru nevedlo k průkaznému zvýšení objasněné variance, posuzovánému F testem:

```
> lm.11<-update(lm.10,.~.+X27+X43+X58)
> add1(lm.11,carn.form,test="F")
```

...

Model *lm.11* lze považovat za výsledný, obsahující relevantní část informace o fylogenetické zpřízněnosti srovnávaných druhů. K tomuto referenčnímu ("nulovému" - baseline) modelu tedy můžeme přidat prediktor, jehož efekt nás zajímá, a otestovat změnu jeho kvality oproti referenčnímu modelu:

```
> lm.final<-update(lm.11,.~.+log(carnivora.2$SB))
> anova(lm.11,lm.final)
Analysis of Variance Table

Model 1: carnivora.2$LS ~ X1 + X71 + X20 + X6 + X68 + X3 + X8 + X9 + X2 +
    X7 + X17 + X28 + X63 + X84 + X69 + X56 + X4 + X14 + X27 +
    X43 + X58
Model 2: carnivora.2$LS ~ X1 + X71 + X20 + X6 + X68 + X3 + X8 + X9 + X2 +
    X7 + X17 + X28 + X63 + X84 + X69 + X56 + X4 + X14 + X27 +
    X43 + X58 + log(carnivora.2$SB)
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1     88 32.281
2     87 32.216  1     0.065 0.1755 0.6763
```

Náš závěr je tedy opět stejný jako u PIC a GLS metod, po fylogenetické korekci není efekt hmotnosti mozku průkazný.

Bodová data – motivační příklad

Naše příkladová data popisují rozmístění hnízd dvou druhů mravenců na ploše o velikosti zhruba 250 x 230 metrů (délkovou jednotkou jsou ale "půl-stopy"). Protože pro tento typ dat je v programu R nejužitečnější asi knihovna *spatstat*, začneme jejím otevřením.

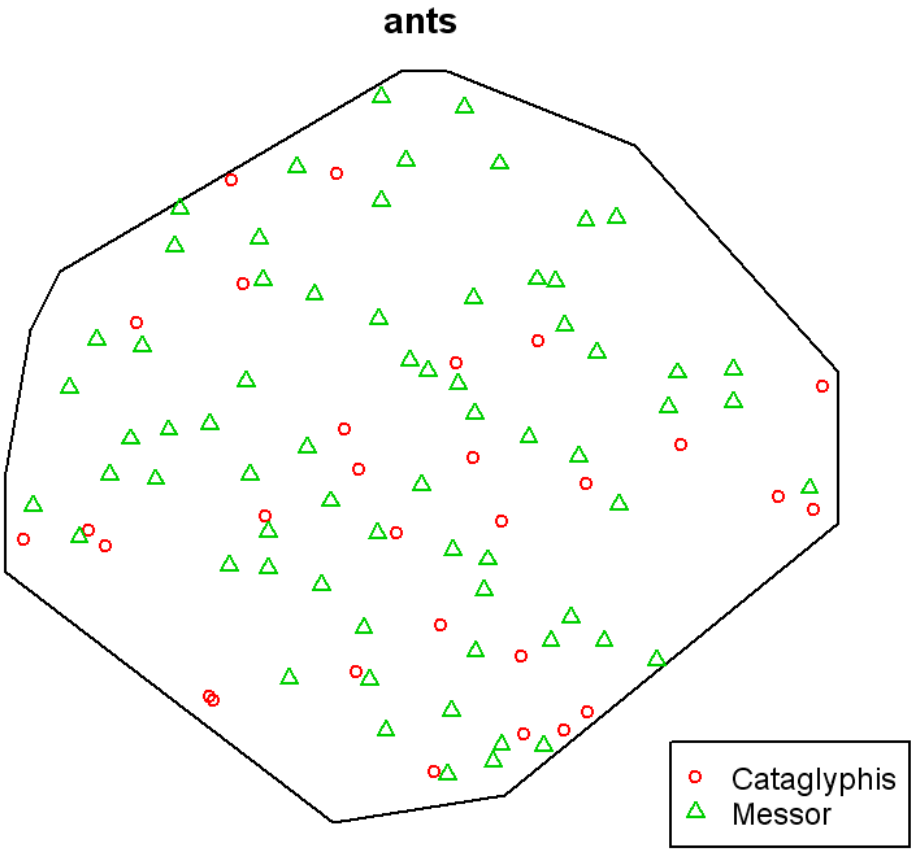
```
> library(spatstat)
> data(ants)
> summary(ants)
Marked planar point pattern: 97 points
Average intensity 0.000226 points per unit area
Marks:
           frequency proportion intensity
Cataglyphis        29      0.299  6.76e-05
Messor             68      0.701  1.59e-04

Window: polygonal boundary
single connected closed polygon with 11 vertices
enclosing rectangle: [ -25 , 803 ] x [ -49 , 717 ]
Window area =  428922
```

```
> plot(ants,cols=2:3)
```



```
> legend("bottomright",pch=1:2,col=2:3, legend=levels(ants$marks))
```

Text, který nám zobrazila funkce *summary* se dost liší od toho, který představuje shrnutí dat například v datových rámcích. Je to proto, že datový objekt *ants* je typu "bodové uspořádání" (point pattern, v R je tento typ dat označován jako *ppp*). Vidíme (též ve vytvořeném grafu – viz Obr. 66), že celkem bylo zaznamenáno 97 hnízd, ta ale patří dvěma druhům mravenců – *Cataglyphis bicolor* s 29 hnízdy a *Messor wasmannii* s 68 hnízdy. Data tedy představují tzv. značkované bodové uspořádání (marked point pattern), ve kterém nejsou všechny zaznamenané pozice (body v rovině) ekvivalentní, protože jsou rozlišeny hodnotou své značky. V našem případě jde o identifikaci druhu, kterému mraveniště patří, nicméně značkované uspořádání může mít i kvantitativní atribut (například údaje o rozmístění jednotlivých stromů na ploše mohou být doplněny jejich výškou nebo průměrem kmene).



Obr. 66

Pro jednoduchost začneme ale s neznačkovaným patternem, který pro naše data získáme jejich rozdělením na dvě bodová uspořádání – každé odpovídající jednomu z druhů. Pozornost ale zaměříme jen na jeden z nich, odpovídající v datech více zastoupenému druhu *Messor* (druhová jména pro zestručnění v následujícím textu vynechávám):

```
> ants.M<-split(ants)$Messor
> summary(ants.M)
Planar point pattern: 68 points
Average intensity 0.000159 points per unit area

Window: polygonal boundary
```

```
single connected closed polygon with 11 vertices
enclosing rectangle: [ -25 , 803 ] x [ -49 , 717 ]
Window area =  428922
```

Všimněme si, že součástí informace o bodovém patternu, která je nezbytná pro jeho následnou analýzu, je znalost tvaru a velikosti plochy, na kterou se vztahují údaje o pozicích jednotlivých jevů (events), v našem případě přítomnosti hnízd. Tato informace je uložena v atributu bodového uspořádání nazvaném *window*. Jak ještě uvidíme, je očekávaná densita (intensita) jevů λ (tj. počet případů vztažený na jednotku plochy) základní informací, kterou budeme studovat a v našich modelech bodových procesů předpovídat. Je proto důležité, abychom skutečnou uvažovanou (při terénním záznamu prohledávanou) plochu správně zaznamenali. Například pro tato data lze sice spočítat obdélník obklopující těsně všechna pozorování (viz "enclosing rectangle" výše, ve výstupu funkce *summary*), ale skutečný tvar plochy, na které byla hnízda zaznamenávána, je polygonální a vyplňuje jen část tohoto obdélníku. Ještě extrémnější situaci si lze představit v případě, že bychom zaznamenávali např. výskyt jedinců jednoho nebo více druhů rostlin, omezených na luční biotopy v podmínkách, ve kterých jde o nepravidelné, částečně propojené plochy, obklopené a prostoupené jiným typem biotopu (např. lesem). Obklopující obdélník by nám o intensitě výskytu rostlin v takovém případě nic zajímavého neřekl. Knihovna *spatstat* je ale ochotna přijmout i složitý polygonální popis skutečně analyzované plochy, který se může skládat z několika oddělených polygonů, a uvnitř každého z nich také mohou být "díry", ve kterých nebyly výskyty zaznamenávány. Polygony se ale nesmí protínat a jejich vrcholy musí být zaznamenávány proti směru hodinových ručiček[50].

Základní shrnutí bodového uspořádání

Výše uvedená funkce *summary* nám o bodech, představujících hnízda druhu *Messor*, poskytla důležitou (ač obsahem jednoduchou) informaci, a tou je průměrná intensita jevu λ, tj. hustota hnízd na jednotku plochy. Hodnota *0.000159* označuje průměrný počet mravenišť na jednu čtvrtinu čtverečné stopy (protože délkovou jednotkou je zde půl stopy) a pokud známe celkový počet jevů (mravenišť) a velikost záznamové plochy (v našem případě polygonu s 11 vrcholy), není její výpočet složitý:

```
> 68/428922
[1] 0.0001585370
```

Pokud by umístění každého hnízda v ploše bylo zcela náhodné, nezávislé na pozici hnízd jiných, a každý bod této plochy měl stejnou očekávanou intensitu výskytu[51], popisoval by rozmístění hnízd nejlépe tzv. homogenní[52] Poissonův proces. Pokud bychom do studované plochy umísťovali čtverce o ploše P a zaznamenávali počet mravenišť, která do jednotlivých ploch padnou, získané hodnoty by pocházely z Poissonovy distribuce s průměrem λP. Jde o obvyklý nulový model, který při studiu uspořádání bodů v ploše

[50] Podrobnosti o zadání polygonu zde probírat nebudeme, ale čtenáře odkazuji na nápovědu pro funkce *ppp* a *owin*.

[51] nešlo by například o gradient od okraje lesa do pole, podle kterého by se měnila dostupnost potravy, a tím i zájem mravenců o umístění hnízda

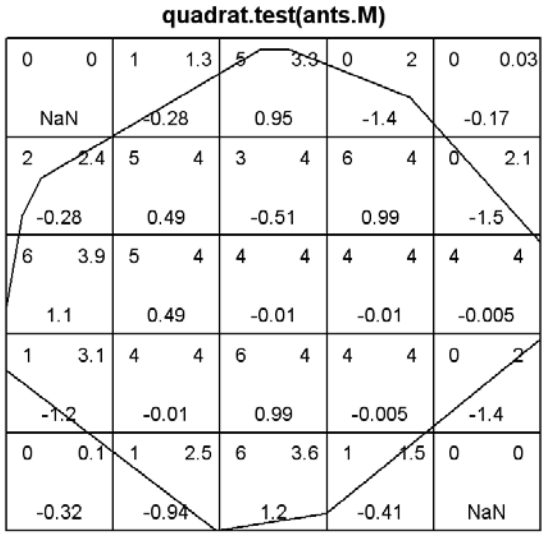
[52] Slovo "homogenní" odkazuje právě na neměnnost očekávané intensity v rámci studované plochy.

používáme jako referenční. Tento model se v literatuře často označuje zkratkou CSR (z Complete Spatial Randomness). Pokud by byl pro naše data správný, nedalo by se říci o uspořádání bodů, navíc k informaci o λ, již nic dalšího. Většinou proto naši práci s bodovým uspořádáním začínáme jeho porovnáním s Poissonovým procesem. Nejjednodušším postupem je překrýt plochu pravidelnou sítí čtverců a porovnat pozorované počty případů v jednotlivých čtvercích s očekávanými počty, pomocí χ^2 testu. Jeho síla ale závisí na počtu pozorování a také na počtu čtverců (řádků a sloupců), do kterých plochu rozdělíme. Naše příkladová data (zejména jen s jedním druhem) jsou malá:

```
> quadrat.test(ants.M)
        Chi-squared test of CSR using quadrat counts
data:  ants.M
X-squared = NaN, df = 24, p-value = NA
Warning message:
Some expected counts are small; chi^2 approximation may be inaccurate in:
quadrat.test(ants.M)
> quadrat.test(ants.M)$expected
 [1] 0.10050251 2.47236181 3.63819095 1.50753769 0.00000000 3.08542714
 [7] 4.02010050 4.02010050 4.01005025 1.95979899 3.85929648 4.02010050
[13] 4.02010050 4.02010050 4.01005025 2.44221106 4.02010050 4.02010050
[19] 4.02010050 2.11055276 0.00000000 1.32663317 3.28643216 2.00000000
[25] 0.03015075
```

Druhým příkazem jsme zobrazil očekávané počty hnízd v jednotlivých 25 čtvercích, na které byl rozdělen obdélník obklopující studovanou plochu. Krom toho, že jsou všechny očekávané počty příliš nízké na to, aby byla χ^2 distribuce dobrou aproximací distribuce testovací statistiky, vidíme také, že v jednom případě je očekávaná hodnota dokonce nulová. To je primární příčinou neúspěchu testu, kde je hodnotou testovací (X^2) statistiky NaN (tj. "nečíslo"): při výpočtu příspěvku tohoto čtverce dělíme nulou! Jak je vůbec možné, že nejsou pro stejně velké čtverce všechny očekávané počty stejné? Čtenáři, který na to zatím ještě nepřišel, napoví hezký obrázek (Obr. 67), který z výsledku testu vytvoří funkce *plot*:

```
> plot(quadrat.test(ants.M))
```



Obr. 67

Každý čtverec má v levém horním rohu uveden počet pozorovaných událostí *O* (počet mravenišť) a v pravém pak (zaokrouhlený, srovnej s textovou informací výše) počet očekávaný (*E*). Z nich je pak vypočten "příspěvek" čtverce do celkové X^2 statistiky (zde vypočteno jako (O-E)/√E) a zobrazen nad dolním okrajem čtverce. Vidíme, že očekávané hodnoty jsou rovny 4.02 jen pro čtverce plně obsažené v polygonu představujícím skutečnou studovanou plochu. U ostatních čtverců je tato hodnota násobena podílem jejich plochy obsažené uvnitř polygonu. Jednoduchým "řešením" našeho problému je snížit počet čtverců, ale oslabujeme tím také výsledný test:

```
> quadrat.test(ants.M,3)
        Chi-squared test of CSR using quadrat counts
data:  ants.M
X-squared = 5.7929, df = 8, p-value = 0.6704

Warning message:

...
```

Pokud bychom se na výsledek tohoto testu spoléhali (např. pro podstatně větší data), závěr by byl, že rozmístění hnízd mravenců *Messor* nelze odlišit od zcela náhodného.

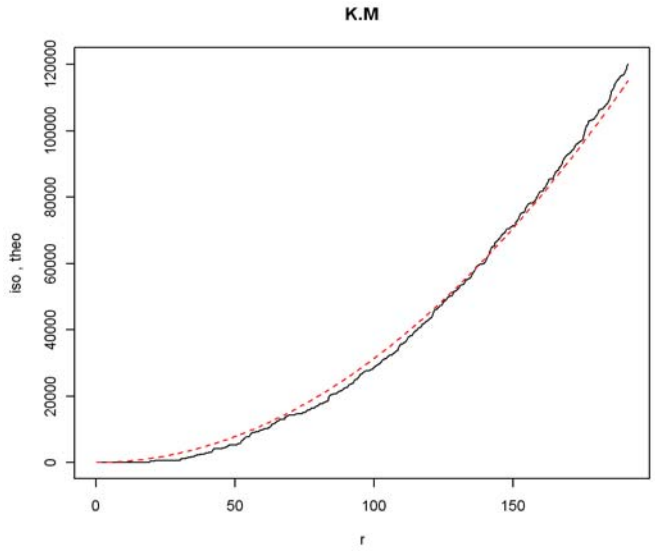
Funkce K a její příbuzenstvo

Případné odchylky od náhodné distribuce bodů v prostoru lze sice také popsat jedním číslem (např. poměrem mezi variabilitou a průměrem počtů pozorování ve čtvercích umístěných přes studovanou plochu), nicméně jen velmi hrubě, tj. jako shlukovité nebo pravidelné rozmístění bodů.

Takové zjištění ale příliš nepohne naším poznáním biologických jevů, které za prostorovým uspořádáním stojí. U mravenců *Messor* bychom očekávali, že existuje určitá minimální hodnota, pod kterou se vzdálenosti mezi jejich hnízdy nemohou dostat. Přinejmenším je to hodnota odpovídající dvojnásobku poloměru typického hnízda (mraveniště nejsou bezrozměrné body a asi se nebudou navzájem překrývat). Nicméně, nižší pravděpodobnost existence dalšího mraveniště můžeme předpokládat i pro větší vzdálenosti, například jako dvojnásobek typické délky potravních cest, vycházejících radiálně z každého mraveniště. Tyto fyzikální (velikost mraveniště) a biologické (kompetice o potravu) procesy by se měly odrazit v celkovém rozmístění mravenišť a my bychom rádi zjistili zda a na jak velké prostorové škále ve skutečnosti působí.

K tomu nám může pomocí tzv. K funkce, kterou v osmdesátých letech 20. století navrhl prof. Brian Ripley. Klasickou K funkci můžeme vynést jako rostoucí křivku, která nám pro různé vzdálenosti od "průměrného" jevu (zde mraveniště) – vynášené na osu X, ukazuje na ose Y očekávaný počet mravenišť vyskytující se do této vzdálenosti. Ve skutečnosti ale K funkce nepředpovídá skutečné počty případů – její hodnoty jsou standardizovány průměrnou intensitou jevů, tj. očekávané počty bychom získali vynásobením hodnoty funkce K odhadem λ. Funkci K si znázorníme pro naše data poměrně jednodušše:

```
> K.M<-Kest(ants.M)
> plot(K.M,cbind(iso,theo)~r)
```

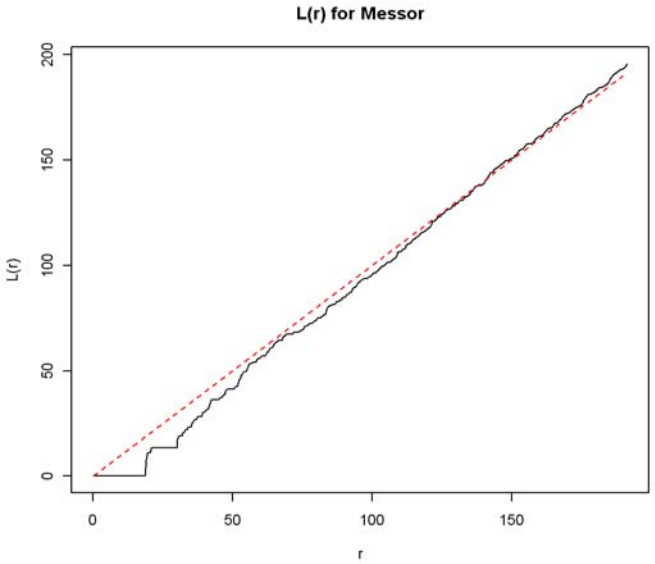


Obr. 68

Teoretický průběh pro náhodný Poissonův proces (v proměnné *K.M* jej představuje proměnná *theo*) je v grafu znázorněn červenou přerušovanou křivkou, zatímco hodnotu K funkce odhadnutou z našeho bodového uspořádání představuje černá křivka. Odpovídající název *iso* odkazuje na typ korekce na okrajový efekt: tuto korekci je vždy třeba provést (protože počet sousedů pro body blíže k okrajům studované plochy je podceněn), názory na nejvhodnější metodu se ale dost liší. Obecně lze říci, že výše zvolená metoda (tzv. isotropická korekce) patří ke kvalitním, její výpočet je ale pro data většího rozsahu časově náročný. Proměnná *K.M* nabízí i další alternativní odhady funkce K, založené na odlišném způsobu korekce okrajového efektu.

Z diagramu také vidíme, že hodnota funkce *K.M* rychle roste: není divu, s rostoucím poloměrem okolí *r* se očekávaný počet sousedů zvyšuje s jeho druhou mocninou (pokud je λ rovno 1, je očekávaná hodnota K rovna πr^2. Proto se také často místo funkce K vynáší proti hodnotám *r* její transformace, $\sqrt{(K/\pi)}$. Takto transformovaná funkce se občas označuje písmenem L a pro naše data si ji vyneseme následujícím způsobem:

```
>  plot(K.M, sqrt(cbind(iso,theo)/pi) ~ r, ylab="L(r)",
+         main="L(r) for Messor")
```



Obr. 69

Vidíme (v Obr. 69), že pozorovaný počet sousedů je nižší než očekávaný až do vzájemné vzdálenosti kolem 120, mraveniště blíže než 20 jednotek (tj. asi 3 metry) v tomto souboru vůbec nejsou. Tento poznatek využijeme později, při fitování statistického modelu pro toto bodové uspořádání.

Problém s diagramem v Obr. 69 je v tom, že významnost odchylky naší K funkce (černá čára) od teoretické hodnoty (červená přímka) z něj nedokážeme posoudit. Můžeme k tomu ale použít tzv. obálky (*envelopes*). Pokud bychom vytvořili bodové uspořádání *n* bodů (kde *n* by odpovídalo počtu bodů v našich datech), které by opravdu bylo výsledkem Poissonova procesu (tj. nezávislého umisťování jednotlivých bodů) a pro toto uspořádání spočítali K funkci, představovaly by její hodnoty jakousi referenci, se kterou můžeme K funkci spočtenou z našich dat porovnávat. Ovšem pokud bychom vytvořili další takové náhodné uspořádání *n* bodů, funkce K by pro něj běžela trochu jinudy, přestože by uspořádání bylo vytvořeno ze stejného náhodného procesu. Když takových uspořádání ale vytvoříme dostatečný počet (například 99), rozsah hodnot z nich spočítaných K funkcí pro danou hodnotu *r* nám dává přibližnou představu o intervalu, ve kterém bude ležet "většina" K křivek, které odpovídají uspořádáním, vzniklým zcela náhodným (Poissonovým) procesem. Význam slova "většina" můžeme upřesnit tak, že pokud opravdu použijeme vždy nejmenší a největší hodnotu daných 99 křivek, představuje jejich rozsah (obálka, envelope) obdobu 98-procentní konfidenční oblasti.[53] Jak si takovou obálku sestrojit a pro danou funkci ji vynést? Ukážeme si to na složitějším příkladu grafu L funkce, který již máme vytvořený v Obr. 69 – čáry 98%-ní obálky do něj přidáme takto:

```
> env.1<-envelope(ants.M, Kest, 199, 5)
> lines(env.1$r, sqrt(env.1$hi/pi), col="red")
> lines(env.1$r, sqrt(env.1$lo/pi), col="red")
```

[53] Pokud bychom při konstrukci obecněji používali *x*-tou nejmenší hodnotu a také *x*-tou největší, a to vždy z *N* sestrojených křivek, byl by pokrytý interval roven hodnotě 100*(1-(2 * *x*)/ (*N*+1))

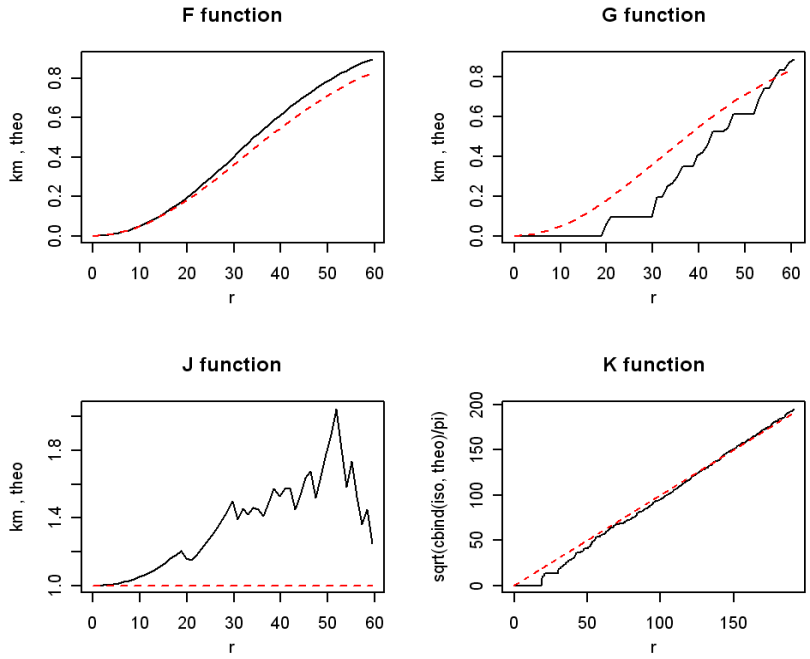
Vidíme, že v rozsahu vzdáleností zhruba 15 až 50 vybočuje K funkce pro naše uspořádání z 95% rozsahu K funkcí zcela náhodných procesů.

Kromě K funkce, která vyjadřuje změnu počtu sousedů s rostoucí vzdáleností od náhodně zvoleného případu (mraveniště), se používají i další funkce, které pracují se vzdálenostmi v bodovém uspořádání. Zde si uvedeme F, G a J funkce. Jejich smysl a použití lépe pochopíme, pokud je uvidíme v grafické podobě. To zvládneme relativně lehce následujícím příkazem:

```
> plot(allstats(ants.M),formule=list(cbind(km,theo)~r,
+   cbind(km,theo)~r,cbind(km,theo)~r,sqrt(cbind(iso,theo)/pi)~r))
```

Jedinou složitější částí je parametr *formule*, pomocí kterého vybíráme charakteristiky, které jsou v každém ze 4 grafů vynášeny.

V pravém dolním rohu výsledného obrázku (Obr. 70) je nám již známá L funkce. Rozsah vzdáleností na horizontální ose tohoto grafu je větší než pro grafy zbývající, protože ostatní vynášené funkce pracují se vzdálenostmi vždy jen k nejbližším sousedům, ne ke všem možným bodům, jak uvidíme v následujících odstavcích.



Obr. 70

F funkce se také nazývá funkcí volného prostoru (empty space) a velmi přibližně můžeme říci, že popisuje vlastnosti volného místa (mezer, gapů) v daném bodovém uspořádání. Popisuje pro libovolné souřadnice (pozor, v tomto případě nejde o existující jev, ale opravdu jakýkoliv bod na ploše) distribuci vzdáleností k nejbližšímu případu ("jak daleko musím z náhodně zvoleného bodu urazit, než dojdu k mraveništi?"). Skutečný průběh ukazuje, že je tato vzdálenost o trochu menší, než bychom předpokládali u zcela náhodného patternu (odhadnutá kumulativní distribuční křivka roste rychleji než teoretická, označená červenou čárou).

Další zajímavou funkcí je tzv. *G* funkce, která popisuje distribuci hodnot vzdáleností mezi určitým případem (jevem) a jeho nejbližším sousedem ("jak daleko musím jít od určitého mraveniště, než dorazím k dalšímu?"). V tomto případě vidíme, že k nejbližším sousedům je v našich datech dále, než bychom očekávali u skutečně náhodného uspořádání.

Funkce *F* a *G* popisují dva komplementární aspekty náhodného resp. shlukovitého či více pravidelného uspořádání bodů. Tyto dva aspekty se také dají poměrně jednoduše zkombinovat jako *(1-G)/(1-F)*. To je *J* funkce ve třetím obrázku. Pro zcela náhodné bodové uspořádání má konstantní hodnotu 1 a je menší než 1 pro shlukovitá a větší než 1 pro více pravidelná uspořádání na dané prostorové škále, dané hodnotou *r*. Graf funkce *J* nám naznačuje nejvyšší pravidelnost na vzdálenosti zhruba 50 jednotek mezi nejbližšími sousedy.

Modelování bodových uspořádání

Dosavadní výsledky nám naznačují, že uspořádání mravenišť druhu *Messor* není tak náhodné, jak by vyplývalo z výsledku χ^2 testu. Výše jsme spíše teoreticky navrhli (a některá z grafických hodnocení bodového patternu to potvrzují), že jednotlivá mraveniště jednak mají určitou minimální vzdálenost, která musí existovat mezi body představujícími jejich středy, jednak že určité biologické jevy vedou k tomu, že i do větší vzdálenostní limity (mezi 50 a 120 jednotkami), existuje určitá "odpudivá síla". Takové uspořádání můžeme popsat jako výsledek tzv. "hard core" procesu, někdy nazývaného Straussův "hard core" proces. Výraz "hard-core" odkazuje na onen menší disk výlučného prostoru kolem každého existujícího bodu, ale společně se základním Straussovým uspořádáním má i toto druhý parametr, kterým je vzdálenost, pod kterou začínají působit interakce mezi body.

Uspořádání bodů modelujeme v knihovně *spatstat* pomocí funkce *ppm*, která v jistém smyslu odpovídá obecným modelovacím funkcím typu *glm*. Pokud bychom chtěli např. *ppm* a *glm* modely porovnávat, musíme si uvědomit, že zde modelujeme očekávanou intensitu procesu (očekávanou densitu mravenišť). Ta se mění jednak podle hrubšího plánu (na větší škále), například systematicky přes výzkumnou plochu z jednoho konce na druhů, nebo mezi lokalitami, nebo v závislosti na faktoru prostředí (např. půdní vlhkost), jednak se mění na drobné škále, odpovídající interakcím mezi jednotlivými jevy (zde mraveništi). První typ variability jsme často schopni popsat pomocí nějakých vysvětlujících proměnných (prostorové souřadnice, faktory, kvantitativní prediktory), zatímco projev té druhé složky se mění mezi vzorky, vždy podle konkrétního uspořádání bodů. V tomto smyslu můžeme o těchto dvou složkách hovořit jako o systematické a stochastické složce modelu, který funkce *ppm* fituje.

Nejjednodušší model (který jsme mezitím již trochu zpochybnili) je homogenní Poissonův, ve kterém se intenzita procesu nemění ani systematicky, ani v závislosti na umístění jednotlivých případů. Nafitujeme jej takto:

```
> ppm.0<-ppm(ants.M,trend=~1, interaction=Poisson())
```

Parametr *trend* představuje systematickou část modelu, změny intenzity vyvolané vnějšími faktory. Hodnota *~1* (tj. konstanta) tedy určuje, že jde o stacionární

(homogenní) uspořádání bez trendu. Stochastická složka (*interaction*) je modelována pomocí funkce *Poisson*, představující absenci interakce mezi jednotlivými body (hnízdy).

Nyní můžeme pokročit k modelu Straussova "hard core" procesu:

```
> ppm.1<-ppm(ants.M,trend=~1,interaction=StraussHard(r=50,hc=10))
> ppm.1
Stationary Strauss - hard core process

Trend:
First order term:
        beta
0.0002386591

Interaction:
Pairwise interaction family
Interaction: Strauss - hard core process
interaction distance:   50
 hard core distance:    10
Fitted interaction parameter gamma:
[1] 0.6679
```

Tento model má, kromě dvou parametrů *r* a *hc*, které musí být předem zadány, ještě jeden – nazvaný gamma, který popisuje povahu interakce mezi body se vzdáleností menší než *r* (tj. méně než 50 pro náš model). Parametr *gamma* menší než jedna udává inhibitivní interakci (vedoucí k uspořádání pravidelnějšímu než je náhodné) – a to je případ našich dat, zatímco hodnota gamma větší než jedna odpovídá tendenci ke shlukování.

S takto nafitovaným modelem můžeme testovat hypotézu, že tento model nepopisuje bodové uspořádání lépe než model *ppm.0*, který předpokládá zcela náhodné rozmístění. Nejde to ale bez jistých obtíží:

```
> anova(ppm.0,ppm.1)
Error in anova.ppm(ppm.0, ppm.1) : Some of the fitted models are not Poisson
processes: p-values are not supported by any theory
```

Autoři se tímto způsobem zříkají odpovědnosti za případné použití parametrického testu[54]. My se ale odvážíme použít test porovnávající deviance modelů, které z *ppm.0* a *ppm.1* vydobudeme odlišnými způsoby:

```
> anova(ppm.0)
Analysis of Deviance Table
Model: quasi, link: log
Response: .mpl.Y
Terms added sequentially (first to last)
     Df Deviance Resid. Df Resid. Dev
NULL                  524     343.52

> anova(ppm.1$internal$glmfit)
Analysis of Deviance Table
Model: quasi, link: log
Response: .mpl.Y
Terms added sequentially (first to last)
            Df Deviance Resid. Df Resid. Dev
NULL                          503     339.65
Interaction  1     8.23       502     331.42
```

[54] a jinde slibují, že budoucí verze knihovny *spatstat* umožní lépe testovat modely i jejich jednotlivé parametry ...

```
> pchisq((343.52-331.42),(524-502))
[1] 0.04472064
```

Výsledek (p=0.0447) je poměrně dobrý, s ohledem na omezený počet bodů v našich datech, ovšem statistická teorie podporující jeho použití není dosud publikována.

Alternativní způsob (méně kontroverzní z pohledu statistické teorie bodových uspořádání), jak porovnat příhodnost dvou alternativních modelů (*ppm.0* a *ppm.1*), je ten, že budeme pomocí simulace vytvářet na základě nafitovaných modelů nová bodová uspořádání stejné velikosti jako má to naše. Pro každé z takto vytvořených uspořádání pak vypočteme např. K funkci a posoudíme, zda K funkce spočtená z našeho výchozího bodového uspořádání může "být jednou z těchto K funkcí". Jinými slovy - tímto způsobem testujeme, zda námi pozorované uspořádání bodů mohlo vzniknout procesem, který popisuje nafitovaný model. Opět zde použijeme funkci *envelope*, se kterou jsme se seznámili již v předchozí sekci:

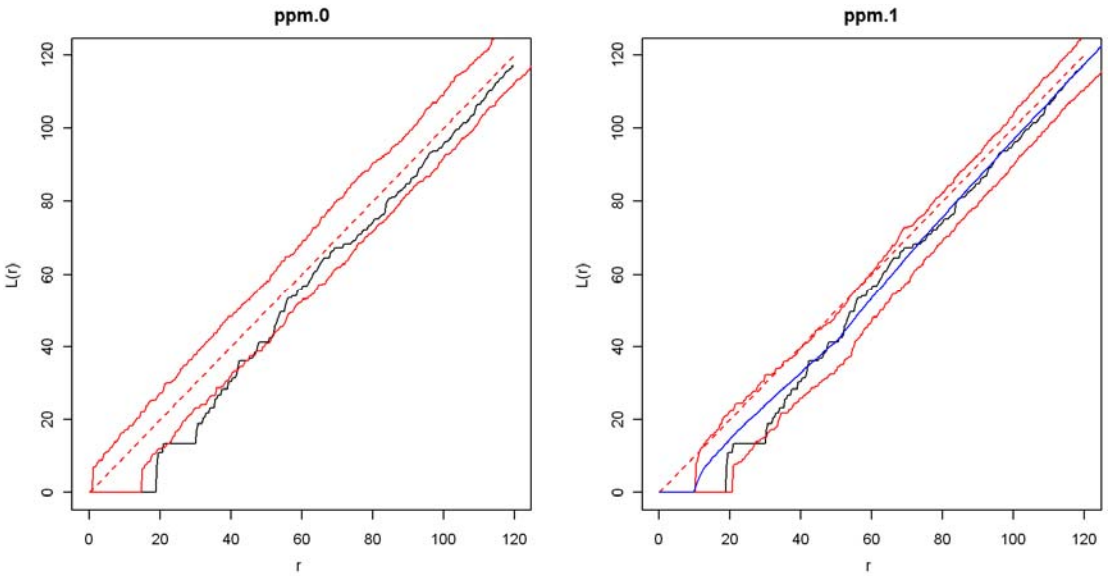
```
> env.0<-envelope(ppm.0,Kest,199,5)
> env.1<-envelope(ppm.1,Kest,199,5)
```

Všimněme si ještě, že funkce *summary* nám pro takto vytvořené objekty poskytne docela užitečné informace:

```
> summary(env.1)
Pointwise critical envelopes for K(r)
Obtained from 199 simulations of fitted model
Upper envelope: pointwise 195th largest of simulated curves
Lower envelope: pointwise 5th smallest of simulated curves
Significance level of Monte Carlo test: 10/200 = 0.05
Data: ppm.1
```

Výsledek si zobrazíme pomocí následujícího kódu, výsledek je pak vidět v Obr. 71.

```
> par(mfrow=c(1,2))
> plot(K.M, sqrt(cbind(iso,theo)/pi) ~ r,
ylab="L(r)",main="ppm.0",xlim=c(0,120))
```



Obr. 71

Náš bodový pattern tedy modelu *ppm.1* v zásadě neodporuje (možná bychom na základě obrázku zvýšili poloměr "hard core" oblasti z 10 na 20), na rozdíl od modelu *ppm.0*, jehož "chování" je až do *r* kolem 50 neslučitelné s našimi daty.

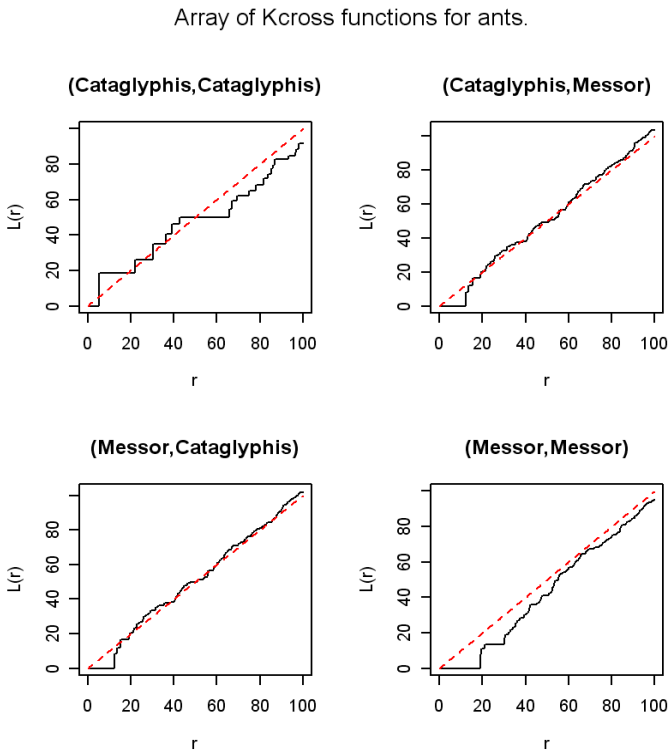
V této sekci jsme si ukázali jen velmi malou část možností funkce *ppm*. Zmiňme jen, že pomocí této funkce můžeme modelovat také trendy v intenzitě procesu. Můžeme buď popisovat prostorové trendy, například parametrem typu *trend=polynom(x,y,2)* popíšeme prostorovou změnu hustoty mravenčích hnízd polynomem druhé stupně, nebo vztah k externím faktorům prostředí (pokryvnost vegetace, pH, vlhkost, apod). U této druhé možnosti ale čtenáře varuji, že pro úspěšné nafitování modelu nepostačuje mít takový faktor prostředí změřen v místě každého jevu (tj. například mravenčího hnízda), musíme mít ještě rozumné množství dalších bodů, ve kterých byly tyto proměnné rovněž změřeny. Pokud se nad tím logicky zamyslíme, dává to smysl – nemohu studovat preference mravenců pro umístění hnízd jenom na základě měření podmínek tam, kde hnízda nakonec opravdu umístili. Jinou možností je, že vysvětlující proměnnou zadáme v podobě rastrové mapy, ve které ji lze získat např. z GIS aplikací. Způsob provedení takovýchto analýz ale vybočuje z rozsahu této učebnice a čtenáře odkazuji na dokumentaci ke knihovně *spatstat*.

Analýza značkovaných bodových uspořádání

V této části se omezíme na jednoduché příklady práce s daty, ve kterých značky představují kategorie určitého faktoru – například biologické druhy. Vrátíme se k původním datům *ants*, obsahujícím údaje o dvou druzích mravenců.

Začněme nejprve u K funkce. V případě porovnání dvou druhů si můžeme představit čtyři různé K funkce: jedna popisuje očekávané počty hnízd druhu *Messor* s rostoucí vzdáleností od hnízda stejného druhu, druhá pak obdobnou charakteristiku pro druh *Cataglyphis*. Pak jsou tu ale také dvě cross-funkce K, ve kterých porovnáváme "napříč" oběma druhy. Všechny tyto funkce může spočítat a zobrazit najednou, pomocí funkce *alltypes*:

```
> plot(alltypes(ants,"K"),formule=sqrt(cbind(iso,theo)/pi)~r,ylab="L(r)",
+  xlim=c(0,100))
```



Obr. 72

Funkci v pravém dolním rohu již známe – je to L funkce pro druh *Messor*. Funkce v levém horním rohu popisuje rozmístění hnízd pro druh *Cataglyphis*, jestliže ignorujeme existenci hnízd druhého druhu. Z průběhu vidíme tendenci k seskupování na malé škále (řádově kolem 10 jednotek, možná jde o dceřinná mraveniště vzniklá rozdělením) a naopak více "inhibiční" uspořádání na větší škále (nad 70 jednotek). Naopak ve dvou cross-funkcích (vpravo nahoře a vlevo dole) není vidět žádný výrazný vztah mezi distribucemi mravenišť obou druhů, samozřejme mimo existence "výlučného prostoru" kolem středu mravenišť.

Na základě těchto výsledků bychom mohli nafitovat například takovýto model:

```
> rr<-matrix(50,2,2)
> hh<-matrix(c(5,10,10,10),2,2)
> types<-levels(ants$marks)
> ppm.2<-ppm(ants,trend=~marks,interaction=MultiStraussHard(types,rr,hh))
```

Hodnota vzorce pro parametr *trend* odpovídá předpokladu, že je intenzita procesu (tj. densita výskytu mravenišť) sice konstantní, ale odlišná pro oba druhy. Interakce je popsána násobným Straussovým procesem, s výlučnými zónami (hard core) a s možnými interakcemi od vzdálenosti 50 jednotek. Oba parametry (*r* a *h*, viz výše pro jednodruhový Straussův hard-core proces) jsou zadány v podobě matice 2x2 (protože máme dva spolu interagující druhy), nižší hodnota pro výlučnou (hard-core) zónu v případě druhu *Cataglyphis* odráží zjištěnou pozitivní interakci na takto krátkou vzdálenost (viz Obr. 72).

Nafitovaný model si můžeme shrnout takto (funkce *summary* poskytuje navíc detaily, které jsou spíše technické povahy):

```
> ppm.2
```

```
Stationary Multitype Strauss Hardcore process
Possible marks:
Cataglyphis Messor
Trend:
Trend formula: ~marks

Fitted first order terms:
beta_Cataglyphis      beta_Messor
    6.525459e-05     2.316562e-04

Interaction:
Pairwise interaction family
Interaction:     Multitype Strauss Hardcore process
2 types of points
Possible types:
[1] "Cataglyphis" "Messor"
Interaction radii:
            Cataglyphis Messor
Cataglyphis          50     50
Messor               50     50
Hardcore radii:
            Cataglyphis Messor
Cataglyphis           5     10
Messor               10     10
Fitted interaction parameters gamma_ij:
            Cataglyphis Messor
Cataglyphis      0.8989 1.1741
Messor           1.1741 0.6648
```

Nejzajímavější jsou asi odhady parametrů *gamma* (viz též výše, u jednodruhového Straussova modelu), které ukazují víceméně pozitivní vazbu mezi hnízdy obou druhů (to by odpovídalo tomu, že se druh *Cataglyphis* živí hlavně mrtvým hmyzem, zde asi hlavně mrtvolami druhu *Messor*). Taková interpretace ovšem závisí na naší apriorní volbě vzdálenosti, pod kterou se interakce začínají objevovat (my jsme zvolili hodnotu 50).

Porovnání s modelem s odlišně zvolenými parametry je sice možné, nicméně knihovna *spatstat* jej příliš neusnadňuje (autoři ale opět slibují vylepšení v budoucích verzích). Porovnávat bychom museli (pomocí funkce *anova*) nikoliv přímo objekty vrácené funkcí *ppm*, ale jejich komponenty, které představují výsledky fitování zástupných zobecněných lineárních modelů (*$internal$glmfit*).