

Frekvenční analýza, čtyřpolní tabulky

V následujícím příkladě nás zajímá, zda sekání má pozitivní vliv na reprodukci studovaného druhu. V experimentu tedy máme dva druhy ošetření (sekané, nesekané) a pro každé máme patnáct ploch. V průběhu sezóny sledujeme, jestli druh vykvete. Nezajímá nás tedy kolik jedinců vykvete, ale zda se na ploše objeví alespoň jedna kvetoucí rostlina.

Výsledná tabulka sledování je zde:

X7	Faktor5	Faktor6
13	1	1
2	1	2
6	2	1
9	2	2

X7 - četnost pozorování, Faktor5 - typ ošetření (sekaná - 1, kontrola - 2), Faktor6 - kvetení (vykvetl - 1, nevykvetl - 2)

Experimenty s četnostmi nějakého jevu v závislosti na působení dvou či více faktorů se řeší pomocí metod frekvenční analýzy. V našem případě máme pouze dva faktory a oba mají pouze dvě hladiny, a tak se jedná o nejjednodušší typ frekvenčních analýz - tzv. čtyřpolní tabulku. Frekvenční analýzy se obvykle řeší pomocí testů Chí-kvadrát nebo Fisherovým exaktním testem (ten pouze pro čtyřpolní tabulky). Předpokladem a nulovou hypotézou testu je nezávislost působení obou faktorů.

Pokud zadáme data do sloupců (sloupec četností, a kódování pro faktory), lze v S+ pouze vytvořit "Crosstabulation" z menu Statistics>Data Summaries>Crosstabulations...

```
*** Crosstabulations ***
Call:
crosstabs(formula = X7 ~ Faktor5 + Faktor6, data = SDF1,
  na.action = na.exclude, drop.unused.levels = T)
30 cases in table
+-----+
|N      |
|N/RowTotal|
|N/ColTotal|
|N/Total  |
+-----+
Faktor5|Faktor6
      |1      |2      |RowTotal|
+-----+-----+-----+
1      |13      | 2      |15      |
      |0.87    |0.13    |0.5      |
      |0.68    |0.18    |         |
      |0.43    |0.067   |         |
+-----+-----+-----+
2      | 6      | 9      |15      |
      |0.4      |0.6      |0.5      |
      |0.32    |0.82    |         |
      |0.2      |0.3      |         |
+-----+-----+-----+
ColTotal|19      |11      |30      |
      |0.63    |0.37    |         |
+-----+-----+-----+
Test for independence of all factors
Chi^2 = 7.033493 d.f.= 1 (p=0.007999917)
Yates' correction not used
```

Tabulka spočte "observed" proporce pro jednotlivé faktory a to jak pro celková data (N/Total) tak i pro jednotlivé řádky a sloupce. Pokud chceme spočítat očekávané "expected" četnosti, musíme počítat v ruce: pro kombinaci 1&1 počítáme z marginálních četností $(15 \cdot 19)/30 = 9.5$.

V menu pro vytvoření tabulky bohužel nelze nastavit podrobnější parametry Chí-kvadrát testu (Yatesova korekce). Pokud bychom chtěli spočítat Fisherův exaktní test či Chí-kvadrát s Yatesovou korekcí, je třeba data zadat buď přímo do kontingenční tabulky, nebo přímo pro každé sledování vlastní řádek (tj. v tomto případě: třináctkrát ve sloupcích Faktor5 a Faktor6 jedna, dvakrát ve sloupci Faktor5 jedna a Faktor6 dva....).

Z menu voláme Statistics>Compare Samples>Counts and Proportions a poté buď Fisherův exaktní test či Chí-kvadrát.

Výsledkem jsou tedy:

```
Fisher's exact test

data:  Faktor5 and Faktor6 from data set frekv
p-value = 0.0209
alternative hypothesis:
two.sided

Pearson's chi-square test with Yates' continuity correction

data:  Faktor5 and Faktor6 from data set frekv
X-square = 5.1675, df = 1, p-value = 0.023
```

V našem případě oba testy zamítly nulovou hypotézu o nezávislosti působení obou jevů. Pokud chceme říci které kombinace faktorů jak ovlivňují kvetení, musíme ještě dopočítat očekávané četnosti a na základě porovnání se skutečnými správně interpretovat. Z výsledků v našem příkladu můžeme říci, že sekání louky pozitivně ovlivňuje kvetení sledovaného druhu rostliny. Yatesova korekce (dostupná pouze pro tabulky 2×2) je zde použita, protože frekvenční tabulky obsahují diskrétní data, avšak Chí-kvadrát je rozdělení spojitě. Zejména by se měla použít, pokud se jedná o tabulky kdy pozorované četnosti jsou nízké (< 5). Pro tyto tabulky je vhodnější použít Fisherův exaktní test.

	observed		expected
	pozorované		očekávané
sekaná/vykvetl	13	>	$(15 \cdot 19)/30 = 9.5$
sekaná/nevykvetl	2	<	$(15 \cdot 11)/30 = 5.5$
kontrola/vykvetl	6	<	$(15 \cdot 19)/30 = 9.5$
kontrola/nevykvetl	9	>	$(15 \cdot 11)/30 = 5.5$

Alternativou pro analýzu dat kde vysvětlující proměnné jsou kategoriální a vysvětlovanou počty jsou kromě výše uvedených Chí-kvadrát testu a Fisherova exaktního testu i zobecněné lineární modely. Použijeme log-lineární model s poissonovským rozdělením chyb. V S+ pro tyto výpočty používáme funkci glm. Její parametry jsou obdobné jako u např. lm (lineární regrese). Jen je třeba doplnit typ rozložení chyb, v našem případě: poisson. Model uložíme pod jménem "pokus" a píšeme tedy:

```
pokus<-glm(x7~Faktor1+Faktor2, data=SDF1, poisson)
```

Výsledek vyvoláme a z celého výpisu nás zajímá pouze hodnota residuální deviance.

```
pokus
Call:
glm(formula = x7 ~ f1 + f2, family = poisson, data = SDF1)
```

```
Coefficients:
(Intercept)          f1          f2
    1.97802 -2.279783e-008 -0.2732718
```

```
Degrees of Freedom: 4 Total; 1 Residual
```

```
Residual Deviance: 7.458882
```

Hodnotu residuální deviance srovnáme s hodnotou Chí-kvadrátu s jedním stupněm volnosti.

Pro výpočet hladiny významnosti voláme:

```
1-pchisq(7.46,1)
[1] 0.006308501
```

Což ukazuje obdobný výsledek jako když jsme příklad počítali Fisherovým testem či přímo Chí-kvadrátem. Pozorované a očekávané hodnoty si vypíšeme pro určení, které kombinace faktorů mají pozitivní či negativní efekt.

```
> SDF1$x7
[1] 13  2  6  9
> fitted(model)
      1      2      3      4
9.5 5.500001 9.5 5.500001
```

Logistická regrese

Pokud máme závislou proměnnou kategoriální a ještě k tomu nabývá pouze dvou hodnot (přežil/nepřežil, kvete/nekvete, nakažený/nenakažený) a zajímá nás které faktory a jak je ovlivňují.

V následujícím příkladu nás bude zajímat jak je pravděpodobnost přežití rostliny ovlivněna množstvím vytvořených semen v závislosti na její velikosti. K dispozici máme údaje o 59 jedincích, zda přežili do další sezóny (surv: 0 - nepřežil, 1 - přežil; flow - počet úborů, root - velikost kořene). Data jsou z příkladu GLEX26 programového balíku GLIM.

1. Výpočet

Vlastní analýzu budeme počítat pomocí logistické regrese z menu **Statistics>Regression>Logistic...** Menu, kde zadáváme vlastní analýzu se příliš neliší od klasické regrese. V tomto příkladě využijeme možnosti S+ a uložíme si model analýzy jako samostatný objekt se kterým budeme později pracovat. Pro uložení je potřeba jen zadat jméno analýzy do okna "Save Model Object".

Po zadání analýzy máme obecnou rovnici: `surv~flow+root`.

Ve výsledcích jsou opět uvedeny hodnoty pro rozdělení reziduí. Hlavní výsledek je v části "Coefficients" kde jsou uvedeny regresní koeficienty pro jednotlivé proměnné spolu s jejich standardními chybami a t-hodnotami. Pravděpodobnosti pro jednotlivé proměnné je třeba dopočítat ručně.

```
*** Generalized Linear Model ***

Call: glm(formula = surv ~ flow + root, family = binomial(link =
  logit), data = GLEX26, na.action = na.exclude, control
  = list(epsilon = 0.0001, maxit = 50, trace = F))
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.42963 -0.7899175 -0.1878944  0.7291494  2.359577

Coefficients:
              Value Std. Error  t value
(Intercept)  0.9615073  0.61588600   1.561177
      flow -0.1064166  0.03331434  -3.194318
      root  6.6003244  2.09356086   3.152679

(Dispersion Parameter for Binomial family taken to be 1 )

Null Deviance: 78.90332 on 58 degrees of freedom

Residual Deviance: 54.06811 on 56 degrees of freedom

Number of Fisher Scoring Iterations: 5
```

2. Test proměnných (funce drop1)

To zda jsou pro model významné můžeme otestovat kromě t-testu obdobným způsobem jako u "stepwise" analýzy, tedy postupně odebírat jednotlivé proměnné a sledovat výslednou statistiku Cp, která bere v potaz množství vysvětlené variability a počet proměnných zahrnutých v modelu.

Máme-li uložený model pod jménem "glex.1" vyvoláme analýzu postupného odebírání proměnných funkcí `drop1`. V okně "Commands" tedy voláme: `drop1 (glex.1)` a dostáváme tabulku.

```
> drop1(glex.1)
Single term deletions
```

Model:

```
surv ~ flow + root
```

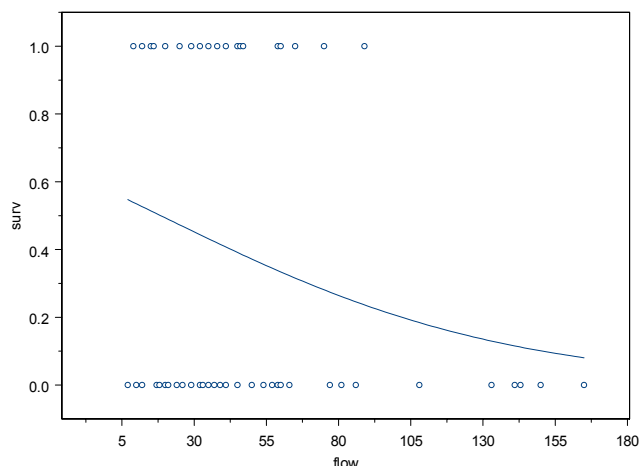
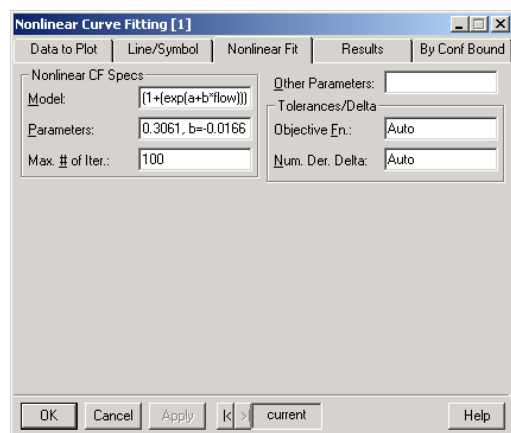
	Df	Sum of Sq	RSS	Cp
<none>			52.46392	58.08506
flow	1	10.20366	62.66759	66.41501
root	1	9.93938	62.40331	66.15073

Testovací statistika Cp v žádném případě neklesla pod hodnotu kompletního modelu a tak tedy můžeme zahrnout obě proměnné do modelu. Získané závislosti tedy ukazují, že čím méně vytvořených květních úborů, tím vyšší pravděpodobnost přežití a zároveň s velikostí kořene také klesá pravděpodobnost smrti.

3. Grafické znázornění

Získané závislosti pro jednotlivé proměnné si můžeme znázornit také v grafech. Pro graf závislosti přežití na počtu vytvořených úborů voláme `Graph>2D plot...>Fit - Nonlinear curvefit (x, y)`. Na osu *x* vynášíme nezávislou, tedy proměnnou `flow`. Na záložce "Nonlinear Fit" vyplníme jakou funkcí data prokládáme. Pole "Model" vyžaduje alespoň jeden parametr a nezávislou proměnnou. V našem případě vyplníme:

$surv = \frac{\exp(a+b \cdot flow)}{1 + \exp(a+b \cdot flow)}$. Do okna "Parameters" zadáme hodnoty regresních koeficientů *a* a *b*. Ty si musíme spočítat předem, pro model s proměnnou `flow` jsou *a* = 0.3061 *b* = -0.0166. V poli parametry oddělujeme čárkou.



4. Vliv interakce (funkce update)

Avšak může nás také zajímat, jaká je závislost mezi přežitím a oběma parametry navzájem, tedy interakce mezi velikostí a investicí do reprodukce. Přidáme tedy do našeho modelu hodnoty interakcí. Interakce přidáme jednoduše buď vytvořením nového modelu či pomocí funkce `update`. Změnu modelu zadáme pomocí "tečkové" konvence v S+. Model můžeme

obecně vyjádřit jako závislost proměnné na levé straně na proměnných, které jsou uvedeny na straně pravé. Tedy v našem případě $\text{surv} \sim \text{flow} + \text{root}$, což je obecně $. \sim .$, kde tečka zastupuje veškeré proměnné na dané straně rovnice. Pokud tedy do modelu `glex.1` chceme přidat interakce pak použijeme funkci `update` takto:

```
update(glex.1, . ~ . +flow:root).
```

Obdobným způsobem můžeme proměnné i odebírat. Pro náš příklad tedy vytvoříme nový model (`glex.full`) na základě `glex.1` s přidáním interakcemi. Voláme tedy:

```
glex.full<-update(glex.1, .~.+flow:root)
```

Po vyvolání modelu `glex.full` již S+ vypíše celý model i s interakcemi

```
> glex.full
```

Call:

```
glm(formula = surv ~ flow + root + flow:root, family =  
     binomial(link = logit), data = GLEX26, na.action =  
     na.exclude, control = list(epsilon = 0.0001, maxit =  
     50, trace = F))
```

Coefficients:

```
(Intercept)          flow          root  flow:root  
-2.960503 -0.07887645 25.15061 -0.2090799
```

Degrees of Freedom: 59 Total; 55 Residual

Residual Deviance: 37.12823

Nyní stejným způsobem jako pro model `glex.1` bez interakcí otestujeme významnost jednotlivých proměnných pomocí funkce `drop1`.

```
> drop1(glex.full)  
Single term deletions
```

Model:

```
surv ~ flow + root + flow:root  
              Df Sum of Sq      RSS      Cp  
<none>                60.63389 69.45336  
flow:root    1     5.58286 66.21675 72.83135
```

A opět vidíme, že všechny proměnné použité v našem modelu mají své opodstatnění. Použité interakce v modelu (regresní koeficient -0.20908) můžeme interpretovat tak, že větší rostliny mají nižší pravděpodobnost přežití pokud vytvoří stejný počet květních úborů jako menší jedinci.

Získaná regresní rovnice tedy vypadá:

$y = -2.960503 - 0.07887645 * \text{flow} + 25.15061 * \text{root} - 0.2090799 * \text{flow:root}$

a odhadnuté pravděpodobnosti přežití pro dané hodnoty vytvořených úborů a velikosti kořene získáme zadáním do $p = \frac{e^y}{1+e^y}$. Hodnoty pro interakce získáme prostým vynásobením hodnot pro kořen a květenství.

Použitá data z příkladu glex26 pro program GLIM:

surv	flow	root	1	32	0.30
0	165	1.57	0	63	0.66
0	41	0.2	1	9	0.2
0	33	0.2	0	7	0.02
0	141	0.91	1	46	0.55
0	150	2.36	1	15	0.25
0	21	0.2	0	24	0.15
0	35	0.2	0	12	0.01
0	57	0.66	0	18	0.15
0	108	1.57	1	35	0.25
1	38	0.66	1	29	0.39
1	41	0.44	0	21	0.25
0	37	0.2	1	12	0.25
0	86	0.66	0	45	0.2
0	32	0.25	0	21	0.25
1	45	0.55	0	59	0.44
0	26	0.2	1	35	0.98
0	143	1.57	1	60	0.66
0	39	0.25	1	35	0.44
1	20	0.2	1	59	0.66
0	50	0.44	1	16	0.15
0	29	0.1	1	65	1.57
1	75	1.18	1	25	0.44
0	60	0.2	1	89	1.97
1	47	0.2			
0	10	0.05			
0	81	0.55			
1	16	0.29			
0	133	0.88			
0	54	0.25			
0	57	0.04			
0	17	0.2			
1	20	0.55			
0	77	0.15			
0	20	0.1			
0	41	0.2			

