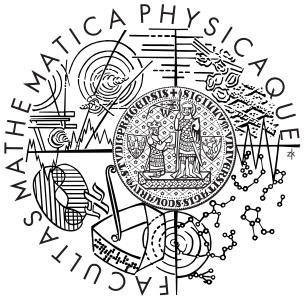
Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

DIPLOMOVÁ PRÁCE



Bc. Přemysl Bejda

Diskrétní a omezené vysvětlované proměnné v ekonometrii

Katedra pravděpodobnosti a matematické statistiky

prof. RNDr. Tomáš Cipra, DrSc., KPMS

ekonometrie

2008/2009

Chtěl bych především poděkovat svému školiteli za laskavý přístup při vedení práce, věcné připomínky a kontrolu gramatiky. Své rodině za podporu. Firmě Penco za poskytnutí dat. Mojí mamince za pomoc při jejich zpracování. Tiboru Vansovi za pomoc při práci se softwarem. Petře Strouhové a mamince za spolupráci při kontrole gramatiky. Otci za vytištění diplomové práce. Knihovně MFF, kde jsem získal většinu materiálů. Všem, kteří se mnou při psaní této práce měli trpělivost.

Prohlašuji, že jsem svou bakalářskou práci napsal(a) samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce a jejím zveřejňováním.

V Praze dne Přemysl Bejda

Obsah

Název práce: Diskrétní a omezené vysvětlované proměnné v ekonometrii
Autor: Přemysl Bejda
Katedra (ústav): Katedra pravděpodobnosti a matematické statistiky
Vedoucí bakalářské práce: prof. RNDr. Tomáš Cipra, DrSc.
e-mail vedoucího: Tomas.Cipra@mff.cuni.cz

Abstrakt: V předložené práci studujeme diskrétní a omezené vysvětlované proměnné. Začneme binárními proměnnými. Ukážeme příklad na praktických datech, ve kterém předvedeme možnosti softwaru EViews a doplníme je o vlastní procedury, které nám pomohou v analýze dat. Pomocí metody „jackknife“, či za pomoci testovací množiny (vybírané prostým náhodným výběrem) zkoumáme, jak je náš model schopen předpovídat. Srovnáme modely logit, probit a gompit. Doplníme graf odhadu podmíněné pravděpodobnosti. Výše zmíněné funkce nejsou v EViews přímo implementovány. Podobně postupujeme v případě ordinálních vysvětlovaných proměnných. Používáme stejná data jako v předchozím příkladu a také doplníme výstupy z EViews o metodu „jackknife“, prostý náhodný výběr a grafy podmíněných pravděpodobností. Zabýváme se statistikou, která by nám mohla pomoci při diskusi o vhodnosti modelu. Pouze z teoretického hlediska zkoumáme neuspořádané vysvětlované proměnné. V druhé kapitole se zaměříme na omezené vysvětlované proměnné. Nejprve probereme cenzorované a pak useknuté vysvětlované proměnné. Jako aplikaci uvažujeme na proměnné vyjadřující dobu trvání. Uvedeme stručně teorii k analýze přežití. Tohoto tématu se týká poslední příklad, který se zabývá tím, do kdy se nějaký výrobek přestane prodávat. Výpočty se provádí v R, neboť v EViews tato problematika není implementována.

Klíčová slova: Diskrétní a omezené vysvětlované proměnné, ekonometrické modelování, EViews, R

Title: Discrete and limited explained variables in econometry
Author: Přemysl Bejda
Department: Department of Probability and Mathematical Statistics
Supervisor: prof. RNDr. Tomáš Cipra, DrSc.
Supervisor's e-mail address: Tomas.Cipra@mff.cuni.cz

Abstract: In the present work we study discrete and limited dependent variables. We begin with binary dependent variables. Then we show an example, where we use the data from psychological area. We work with econometric software EViews and show its possibilities, which are connected with our subject of study. We write procedures for "jackknife" method and simple random sample, compare logit, probit and gompit models and draw a graph of conditional probability of our models. Likewise we work with ordinal dependent variables. We use the same data as in the previous example. It means that we investigate possibilities of EViews and add some procedures for "jackknifing," simple random sampling and for drawing pictures of conditional probability. Just from theoretical point of view we consider unordered dependent variables. In the next chapter we focus on limited dependent variables. We show theory of censored and truncated explained variables. As an application we show theory of survival analysis, which is used in our last example. Statistical computing is performed in R, because no suitable methods are implemented in EViews.

Keywords: Discrete and limited dependent variables, econometric modeling, EViews, R

Úvod

Předložená diplomová práce obsahuje některé dílčí příspěvky k problematice diskrétních a omezených vysvětlovaných proměnných, a to jak z hlediska teoretického, tak z hlediska softwarových aplikací.

Nejprve se zabýváme binárními proměnnými. Uvedeme příklad na tuto problematiku řešený pomocí EViews. Především najdeme vhodný model, který se hodí k našim datům. Zkoumáme, oč je lepší námi zkonstruovaný model oproti modelu pouze s konstantou. Z tohoto důvodu vytvoříme novou proceduru s naprogramovanou metodou „Jackknife“. Kdybychom měli rozsáhlá data, bylo by vhodnější použít nějakou testovací množinu. My naprogramujeme proceduru, která vybírá testovací množinu prostým náhodným výběrem. Dále ukážeme jak lze v EViews srovnat modely logit, probit a gompit. Na závěr tohoto příkladu vytvoříme (jako doplněk stávajícího softwaru) graf odhadu podmíněné pravděpodobnosti.

Další část se týká ordinálních vysvětlovaných proměnných. Nejprve se jimi zabýváme teoreticky. Dále uvedeme příklad se stejnými daty jako u binárních proměnných. Nalezneme vhodný model. Upozorníme na nepříliš vhodnou tabulku, kterou EViews obsahuje. Opět naprogramujeme metodu „Jackknife“ a prostý náhodný výběr. Zavedeme novou statistiku, která nám může pomoci při rozhodování o korektnosti modelu a diskutujeme její výhody a nevýhody. Vytvoříme (pomocí krátké procedury) graf odhadů podmíněných pravděpodobností pro modely probit, logit a gompit. Také necháme vytvořit graf, který znázorňuje odhady podmíněných pravděpodobností pro všechny hodnoty vysvětlované proměnné.

Dále se zabýváme pouze teoreticky neuspořádanými diskrétními vysvětlovanými proměnnými.

Druhá kapitola se věnuje omezeným vysvětlovaným proměnným. Začneme cenzorovanými a useknutými proměnnými, ale hlavní důraz je kladen na veličiny vyjadřující dobu trvání. Pro tyto veličiny je uveden příklad na reálných datech, která poskytla firma Penco. Tentokrát pracujeme v soft-

waru R. Postupně vyzkoušíme všechny modely, které popisujeme v teoretické části. Necháme vykreslit grafy funkce přežití pro dva z těchto modelů.

Na přiloženém CD naleznete, kromě diplomové práce, také procedury, kterými se budeme zabývat, a tabulky s daty.

Podstatnou součástí diplomové práce jsou numerické příklady pro příslušné vysvětlované proměnné, které se týkají reálných dat. Pro tyto příklady je provedena detailní ekonometrická analýza těch aspektů, kterými se diplomová práce zabývá.

Kapitola 1

Diskrétní vysvětlované proměnné

1.1 Binární vysvětlovaná proměnná

Velmi častým typem kategoriální vysvětlované proměnné je *binární proměnná (binary dependent variable)* nabývající jako svých hodnot pouze jedničky či nuly. Tento typ proměnné se vyskytuje především v těchto případech:

- Jedná se o dummy proměnnou, tj. proměnnou, která nabývá kvůli své podstatě pouze dvou hodnot. Může to být logická proměnná, odpověď v anketě ano či ne atd.

- Jedná se o proměnnou, která je vytvořená z jiné jejím zjednodušením, např. cena výrobku vyšší, či nižší, než padesát korun atd.

Lineární model se může konstruovat stejně jako v případě, kdy vysvětlovaná proměnná je spojitá. Jenže v tuto chvíli nemá velký význam prokládat mrakem bodů přímku. Vzniká klíčová otázka interpretace takového modelu.

Podívejme se tedy na daný model blíže. Řekněme, že pro binární vysvětlovanou proměnnou y_t (v čase t, nebo pro $t-$tou pozorovanou jednotku průřezového výběru) platí: 1 znamená, že došlo k výskytu sledovaného jevu, a 0, že k němu nedošlo. Pak lze pravděpodobnostní model zapsat následujícím způsobem

$$\mathsf{P}(y_t = 1|\boldsymbol{x}_{t\cdot}, \boldsymbol{\beta}) = 1 - F(-\boldsymbol{x}_{t\cdot}\boldsymbol{\beta}), \qquad t = 1, \ldots, T, \tag{1.1}$$

či ekvivalentně

$$\mathsf{P}(y_t = 0|\boldsymbol{x}_{t\cdot}, \boldsymbol{\beta}) = F(-\boldsymbol{x}_{t\cdot}\boldsymbol{\beta}), \qquad t = 1, \ldots, T, \tag{1.2}$$

kde $F(\cdot)$ je vhodná spojitá distribuční funkce. Tento způsob zápisu předpokládá, že čím je výraz $\boldsymbol{x}_{t\cdot}\boldsymbol{\beta}$ vyšší, tím bude i pravděpodobnost, že y_t nabyde hodnoty 1. Je-li distribuční funkce symetrická, resp. její hustota funkce sudá, pak lze (1.1) a (1.2) přepsat do tvaru

$$\mathsf{P}(y_t = 1|\boldsymbol{x}_{t\cdot}, \boldsymbol{\beta}) = F(\boldsymbol{x}_{t\cdot}\boldsymbol{\beta}), \qquad \mathsf{P}(y_t = 0|\boldsymbol{x}_{t\cdot}, \boldsymbol{\beta}) = 1 - F(\boldsymbol{x}_{t\cdot}\boldsymbol{\beta}).$$

Poznámka 1.1 *Výše zmíněný problém interpretace se může řešit různými přístupy. My zde uvedeme tři z nich.*

1. V prvním případě budeme používat skrytou, neboli latentní proměnnou y^*, která je provázána s regresory $\boldsymbol{X}$ lineárním modelem

   $$y_t^* = \boldsymbol{x}_{t\cdot}\boldsymbol{\beta} + \varepsilon_t, \tag{1.3}$$

   kde ε_t jsou iid náhodné veličiny s nulovou střední hodnotou. To znamená , že (1.3) je obvyklý lineární model a y^* je spojitá vysvětlovaná proměnná. Jakých hodnot nabývá náhodná veličina y, určíme následujícím způsobem (ptáme se, zda je její hodnota nad, či pod nulovým prahem)

   $$y_t = \begin{cases} 1 & \text{pro } y_t^* > 0 \\ 0 & \text{pro } y_t^* \leq 0. \end{cases}$$

   Odtud dostaneme

   $$\mathsf{P}(y_t = 1|\boldsymbol{x}_{t\cdot}, \boldsymbol{\beta}) = \mathsf{P}(y_t^* > 0|\boldsymbol{x}_{t\cdot}, \boldsymbol{\beta}) = \mathsf{P}(\boldsymbol{x}_{t\cdot}\boldsymbol{\beta} + \varepsilon_t > 0) = 1 - F(-\boldsymbol{x}_{t\cdot}\boldsymbol{\beta}).$$

   Nyní ovšem interpretujeme $F(\cdot)$ jako distribuční funkci reziduální složky ε modelu (1.3). Volba nulové úrovně prahu není podstatná, pokud model (1.3) obsahuje intercept.

2. Další interpretace využívá podmíněné střední hodnoty

   $$\begin{aligned} \mathsf{E}(y_t|\boldsymbol{x}_{t\cdot}, \boldsymbol{\beta}) &= 1 \cdot \mathsf{P}(y_t = 1|\boldsymbol{x}_{t\cdot}, \boldsymbol{\beta}) + 0 \cdot \mathsf{P}(y_t = 0|\boldsymbol{x}_{t\cdot}, \boldsymbol{\beta}) \\ &= \mathsf{P}(y_t = 1|\boldsymbol{x}_{t\cdot}, \boldsymbol{\beta}) = 1 - F(-\boldsymbol{x}_{t\cdot}\boldsymbol{\beta}). \end{aligned}$$

   Jestliže píšeme

   $$y_t = \big(1 - F(-\boldsymbol{x}_{t\cdot}\boldsymbol{\beta})\big) + \varepsilon_t,$$

potom ε představuje odchylku náhodné veličiny y od její podmíněné střední hodnoty a platí pro ni

$$\mathsf{E}(\varepsilon_t|\boldsymbol{x}_{t\cdot},\boldsymbol{\beta}) = 0, \qquad \mathsf{var}(\varepsilon_t|\boldsymbol{x}_{t\cdot},\boldsymbol{\beta}) = F(-\boldsymbol{x}_{t\cdot}\boldsymbol{\beta})\big(1 - F(-\boldsymbol{x}_{t\cdot}\boldsymbol{\beta})\big).$$

Rozptyl stačí spočítat pro y_t, neboť $1-F(-\boldsymbol{x}_{t\cdot}\boldsymbol{\beta})$ je díky podmíněnosti pouze konstanta.

3. Kdybychom použili nejjednodušší možnou konstrukci a model zapsali ve tvaru $y_t = \boldsymbol{x}_{t\cdot}\boldsymbol{\beta} + \varepsilon_t$ a díky nulové střední hodnotě reziduí spočítali $\boldsymbol{x}_{t\cdot}\boldsymbol{\beta} = \mathsf{E}(y_t) = 0 \cdot \mathsf{P}(y_t = 0) + 1 \cdot \mathsf{P}(y_t = 1) = \mathsf{P}(y_t = 1)$, pak by se vyskytly následující problémy. Bylo by nutné přidat omezení $0 \leq \boldsymbol{x}_{t\cdot}\boldsymbol{\beta} \leq 1$ a rezidua by byla heteroskedastická. Tato interpretace se tedy nepoužívá.

Jednotlivým parametrům β_i nemůžeme přiřknout stejný význam, jako je tomu u obvyklého lineární modelu, ale pokusme se o následující analýzu

$$\frac{\partial \mathsf{E}(y_t|\boldsymbol{x}_{t\cdot},\boldsymbol{\beta})}{\partial x_{ti}} = \frac{\partial \mathsf{P}(y_t = 1|\boldsymbol{x}_{t\cdot},\boldsymbol{\beta})}{\partial x_{ti}} = f(-\boldsymbol{x}_{t\cdot}\boldsymbol{\beta}) \cdot \beta_i, \tag{1.4}$$

kde $f(\cdot)$ je hustota odpovídající nějaké distribuční funkci $F(\cdot)$. Odtud

$$\frac{\partial \mathsf{E}(y_t|\boldsymbol{x}_{t\cdot},\boldsymbol{\beta})/\partial x_{ti}}{\partial \mathsf{E}(y_t|\boldsymbol{x}_{t\cdot},\boldsymbol{\beta})/\partial x_{tj}} = \frac{\beta_i}{\beta_j},$$

tedy podíl dvou parametrů odpovídá podílu dvou rychlostí změny při změně dvou odpovídajících regresorů. Používaným nástrojem je *preferenční poměr (odds ratio)*

$$\frac{\mathsf{P}(y_t = 1|\boldsymbol{x}_{t\cdot},\boldsymbol{\beta})}{\mathsf{P}(y_t = 0|\boldsymbol{x}_{t\cdot},\boldsymbol{\beta})} = \frac{1 - F(-\boldsymbol{x}_{t\cdot}\boldsymbol{\beta})}{F(-\boldsymbol{x}_{t\cdot}\boldsymbol{\beta})} = \frac{F(\boldsymbol{x}_{t\cdot}\boldsymbol{\beta})}{1 - F(\boldsymbol{x}_{t\cdot}\boldsymbol{\beta})},$$

který relativně udává pravděpodobnost výskytu jevu vůči tomu, že jev nenastane. Poslední rovnost platí pouze za předpokladu, že $F(\cdot)$ je symetrická.

V praxi se ovšem používají jen některá speciální rozdělení. Uveďme tedy nejčastěji užívané modely.

1. *Probit:*

$$\mathsf{P}(y_t = 1|\boldsymbol{x}_{t\cdot},\boldsymbol{\beta}) = 1 - F(-\boldsymbol{x}_{t\cdot}\boldsymbol{\beta}) = 1 - \Phi(-\boldsymbol{x}_{t\cdot}\boldsymbol{\beta}) = \Phi(\boldsymbol{x}_{t\cdot}\boldsymbol{\beta}).$$

Používá distribuční funkci normálního rozdělení $\Phi(\cdot)$, přesněji distribuční funkci rozdělení $N(0,1)$.

2. *Logit*

$$\mathsf{P}(y_t = 1|\boldsymbol{x}_{t\cdot}, \boldsymbol{\beta}) = 1 - F(-\boldsymbol{x}_{t\cdot}\boldsymbol{\beta}) = 1 - \frac{\mathrm{e}^{-\boldsymbol{x}_{t\cdot}\boldsymbol{\beta}}}{1 + \mathrm{e}^{-\boldsymbol{x}_{t\cdot}\boldsymbol{\beta}}} = \frac{\mathrm{e}^{\boldsymbol{x}_{t\cdot}\boldsymbol{\beta}}}{1 + \mathrm{e}^{\boldsymbol{x}_{t\cdot}\boldsymbol{\beta}}}$$

používá distribuční funkci logistického rozdělení. Výsledky jsou velmi podobné, jako v předchozím případě. Hustota logistického rozdělení je $f(x) = \frac{\mathrm{e}^x}{(1+\mathrm{e}^x)^2}$. Jeho referenční poměr je $\exp(\boldsymbol{x}_{t\cdot}\boldsymbol{\beta})$. Bližší informace o logistickém rozdělení lze najít např v [3, str. 23]. Pak stačí dosadit $a = 0$ a $b = 1$. Jednoduchým výpočtem se pak už dostaneme k předchozím vzorcům.

3. *Gompit*

$$\begin{aligned}\mathsf{P}(y_t = 1|\boldsymbol{x}_{t\cdot}, \boldsymbol{\beta}) &= 1 - F(-\boldsymbol{x}_{t\cdot}\boldsymbol{\beta}) = 1 - \big(1 - \exp\big(-\mathrm{e}^{-\boldsymbol{x}_{t\cdot}\boldsymbol{\beta}}\big)\big) \\ &= \exp\big(-\mathrm{e}^{-\boldsymbol{x}_{t\cdot}\boldsymbol{\beta}}\big).\end{aligned}$$

Distribuční funkce má stejné rozdělení jako náhodná veličina s extremálním rozdělením typu I. Pomocí tohoto rozdělení se modelují extremální hodnoty. Je nesymetrické s nenulovou šikmostí.

Poznámka 1.2 *Samozřejmě vyvstává přirozená otázka, které z těchto tří rozdělení zvolit. Logistické rozdělení má distribuční funkci velmi podobnou normálnímu, jen má těžší „chvosty.“ Připomíná t-rozdělení se sedmi stupni volnosti. Z tohoto vyplývá, že pro hodnoty $\boldsymbol{x}_{t\cdot}\boldsymbol{\beta}$, které jsou blízké nule, řekněme, že se pohybují v intervalu (-1,2;1,2), dostaneme u obou modelů velmi podobné pravděpodobnosti. Logit model dává větší pravděpodobnosti hodnotě $y = 0$, pokud $\boldsymbol{x}_{t\cdot}\boldsymbol{\beta}$ je velmi malé. Naopak pokud $\boldsymbol{x}_{t\cdot}\boldsymbol{\beta}$ je vysoké, pak dostaneme u modelu logit nízký odhad pravděpodobnosti toho, že $y = 0$ ve srovnání s modelem probit.*

Je obtížné dát obecné pravidlo, zda vybrat logit, či probit, neboť by bylo nutné znát dopředu správné parametry $\boldsymbol{\beta}$. Ovšem v následujících dvou případech se mohou výsledky z obou rozdělení lišit podstatně, a to pokud je u vysvětlované proměnné znatelně více pozorování jednoho druhu. Nebo pokud má důležitá vysvětlující proměnná vysokou variabilitu, a to zvláště pokud je pravdivý i první případ.

Pak je většinou nutné rozlišovat případ od případu. Někdy lze preferovat jedno rozdělení před druhým, ale není vyřešeno, jak zobecnit vhodnost použití toho kterého modelu. Hlouběji se touto otázkou zabývá článek [2].

Ovšem ve většině případů se nezdá, že by byl významný rozdíl v použití modelů probit a logit.

Jiná situace nastane, pokud použijeme asymetrické rozdělení, např. model gompit. Potom se výsledky mohou lišit více. I v tomto případě je ovšem těžké rozhodnout, zda použít gompit, nebo předchozí dva.

Odhad parametru $\boldsymbol{\beta}$ se většinou provádí metodou maximální věrohodnosti, čili ML metodou. Tato metoda bývá též používána softwarem. O metodě maximální věrohodnosti viz [3, str. 146]. Věrohodnostní funkce

$$l(\boldsymbol{\beta}) = \prod_{t=1}^{T} \big(1 - F(-\boldsymbol{x}_{t\cdot}\boldsymbol{\beta})\big)^{y_t} \big(F(-\boldsymbol{x}_{t\cdot}\boldsymbol{\beta})\big)^{1-y_t}$$

přejde po zlogaritmování do tvaru

$$L(\boldsymbol{\beta}) = \sum_{t=1}^{T} y_t \ln\big(1 - F(-\boldsymbol{x}_{t\cdot}\boldsymbol{\beta})\big) + \sum_{t=1}^{T} (1 - y_t) \ln\big(F(-\boldsymbol{x}_{t\cdot}\boldsymbol{\beta})\big). \tag{1.5}$$

V případě symetrické distribuční funkce můžeme psát

$$L(\boldsymbol{\beta}) = \sum_{t=1}^{T} \ln\big(F(\boldsymbol{x}_{t\cdot}\boldsymbol{\beta})\big) + \sum_{t=1}^{T} (1 - y_t) \ln\big(1 - F(\boldsymbol{x}_{t\cdot}\boldsymbol{\beta})\big).$$

Tuto funkci budeme maximalizovat přes $\boldsymbol{\beta}$. Tak získáme odhad $\hat{\boldsymbol{\beta}}$.

Lze také konzistentně odhadnout (asymptotickou) rozptylovou matici tohoto odhadu. Např. v modelu logit vypadá

$$\left(\sum_{t=1}^{T} f(\boldsymbol{x}_{t\cdot}\hat{\boldsymbol{\beta}}) \boldsymbol{x}_{t\cdot}^{\top} \boldsymbol{x}_{t\cdot} \right)^{-1},$$

kde $f(\cdot)$ je hustota logistického rozdělení.

Kvalita odhadnutého modelu se posuzuje pomocí tzv. *McFaddenova koeficientu* R^2_{McFadden}. V praxi se používá obdobně jako koeficient determinace. Je založen na věrohodnostním poměru

$$R^2_{\text{McFadden}} = 1 - \frac{L_U}{L_R},$$

kde L_U je maximální hodnota logaritmické věrohodnostní funkce (1.5) a L_R je její maximální hodnota, pokud platí omezení $\beta_1 = \beta_2 = \cdots = \beta_k = 0$.

Předchozí modely můžeme použít pro předpověď. Mějme vektor vysvětlujících proměnných x^* a chtějme odhadnout, jaká by měla být hodnota vysvětlované proměnné. Model předpovídá, že daný jev nastane (tj. $\bar{y} = 1$), pokud

$$\hat{\bar{\mathsf{P}}}^* = 1 - F(-\bar{\boldsymbol{x}}^\top \hat{\boldsymbol{\beta}}) \geq 0,5. \tag{1.6}$$

Další užitečnou pomůckou, která se uvádí na výstupu mnoha softwarů je počet těch t, kde $t = 1, \ldots, T$, pro která by daný model dával správné výsledky. Tj. pro dané $\boldsymbol{x_t}$ by předpověděl skutečnou hodnotu y_t.

Příklad 1.1 *Nyní si ukažme příklad odhadu nějaké binární proměnné pomocí výše zmíněných modelů.*

Nejprve se ovšem musíme seznámit s daty, která budeme používat. Byla převzata z [8]. Tento článek je velmi zajímavý a obsahuje podrobnou analýzu dat. My s nimi budeme pracovat odlišným způsobem, protože je používáme pouze kvůli demonstračním účelům. Ve výše zmíněném článku lze také nalézt jejich podrobnější popis.

Odkud tedy pochází naše data? V Austrálii byl prováděn test, kterého se účastnilo 134 lidí, 88 žen a 46 mužů. Většina z nich byli studenti vysoké školy. Jejich věk se průměrně pohyboval okolo 23 let. Přitom 66 z nich studovalo psychologii a ostatní většinou humanitní vědy jako sociologii, historii aj. Byli vybíráni přímo ve škole nebo v kavárně.

Účastníci byli rozděleni do dvou skupin. Jedné skupině bylo sděleno, že pokud budou postupovat správně, mohou vyhrát 150 ATS (australských šilinků), což je přibližně 200 korun. V druhé skupině pouze řekli, ať si představí, že mohou danou částku vyhrát.

Účastníci byli znovu rozděleni do dvou skupin, ale jiných než v předchozím případě. Oběma skupinám byly předloženy příklady z testu zkoumajícího znalost slov. První skupině bylo ukázáno obtížné zadání a druhé jednodušší. Skupiny také později dostanou různé testy. První skupina lehčí, druhá obtížnější.

Po této proceduře si každý mohl vybrat z následujících možností:

1. *Psát test a v případě, že by výsledek dopadl dobře, získat slíbený finanční obnos nebo si představit jeho získání. Výsledek je dobrý, pokud se nachází v horní půli mezi ostatními výsledky. Tedy je lepší*

než nejméně 50% ostatních výsledků. Této možnosti volby říkejme test. Resp. jedinec si vybral možnost Test.[1]

2. *Hodit šestistěnnou kostkou. S 50% pravděpodobností vyhrát. V tomto případě budeme mluvit o možnosti loterie.*

Účastníkům byly po výběru kladeny otázky jako: „Jste si jist, že jste udělal(a) správné rozhodnutí "; „Jak dobrý budete v testu?";„Kolik bodů si myslíte, že získáte?" aj. Tyto odpovědi byli kvantifikovány. Později se o nich ještě zmíníme, když budeme mluvit o veličinách.

Nezávisle na volbě uchazeče se psal test a házelo kostkou. Tj. všichni psali test a každý také hodil kostkou. Poté byly testy vyhodnoceny a uchazeči odměněni. Znovu jim bylo položeno několik otázek jako: „Jste spokojeni se svými výsledky testu?"

Popišme veličiny, které se v datech objevují. Názvy jsme převedli do češtiny.

Název	*Popis*	*Hodnoty*
obt	*Obtížnost testu (1=těžký)*	*0, 1*
plat	*Zda účastník obdrží skutečně peníze, nebo si má pouze představovat jejich obdržení. (1=dostane peníze)*	*0, 1*
hlas	*Zda si jedinec vybral test, nebo loterii. (1= Test)*	*0, 1*
jist	*„Jste si jist, že jste si vybral správně?" (7=velmi jist)*	*1 , . . . , 7*
zmenroz	*„Jak obtížné by pro Vás bylo změnit rozhodnutí?" (7=velmi obtížné)*	*1 , . . . , 7*
dulez	*„Je pro Vás důležité uspět v testu?" (7= velmi důležité)*	*1 , . . . , 7*
odhobt	*„Myslíte si, že test bude obtížný?" (7= ano, velmi)*	*1 , . . . , 7*
buddobr	*„Jak dobrý budete v testu?" (7= velmi dobrý)*	*1 , . . . , 7*
budbod	*„Kolik bodů z 20 možných nejspíš získáte?" (Čím více bodů, tím lepší výsledek)*	*0 , . . . , 20*
budostbod	*„Jaký bude průměrný výsledek ostatních účastníků?"*	*0 , . . . , 20*

[1]Výsledky budou porovnávány v příslušné skupině obtížnosti

Název	*Popis*	*Hodnoty*
vysl	*Výsledek testu.*	*0 , . . . , 20*
spokoj	*„Jste spokojeni se svým výsledkem?“ (7=velmi spokojen)*	*1 , . . . , 7*
dobrrozh	*„Jste si jisti, že jste udělali správné rozhodnutí ohledně výběru mezi testem a loterií?“ Tato otázka byla kladena po testu. (7=velmi jist)*	*1 , . . . , 7*
menit	*„Jak obtížně by se Vám nyní měnilo rozhodnutí týkající se Vaší volby?“ (7=velmi obtížně)*	*1 , . . . , 7*
testobt	*„Zdál se Vám test obtížný?“ (7=velmi obtížný)*	*1 , . . . , 7*
majostbod	*„Kolik bodů bude průměrný výsledek skupiny?“ Tato otázka byla kladena po testu, ale před vyhodnocením výsledků.*	*0 , . . . , 20*
vek	*Věk v letech*	*17 , . . . , 32*
pohl	*Pohlaví (2=muž)*	*1, 2*
leps	*Poměr mezi odhadnutým svým výsledkem a odhadnutým průměrným výsledkem ostatních. Tento poměr se získával z veličin, které se zkoumali před psaním testu.*	*0,29 ;. . .; 1,67*
lipnezost	*„Byl odhad mého výsledku vyšší, než odhadovaný průměrný výsledek skupiny?“ (1=ano)*	*0, 1*

Tabulku s daty viz dodatek A.2.

My se budeme zabývat odhadem veličiny hlas*. Tedy tím, jestli si jedinec vybral loterii, či test. Nejprve se pokusíme vysvětlit* hlas *pomocí veličin, jež můžeme zjistit ještě před tím, než účastníkovi vysvětlíme principy našeho testu. Tzn. budeme používat veličiny* plat, pohl, vek *a* obt*.*

Podle různých informačních kritérii jako např. Akaikeho, Schwarzova aj. se zdá být nejlepší odhad, kdy hlas *závisí pouze na konstantě a* obt*.* $R^2_{McFadden}$ *sice trochu klesl, na rozdíl od modelu se všemi proměnnými, ale tento pokles je pouze o 0,4%. Nicméně je také pravda, že* $R^2_{McFadden}$ *je velmi nízký z čehož můžeme usuzovat, že náš model není příliš dobrý.*

Všechny výpočty byly prováděny v EViews. Podívejme se nyní na výstup.

Dependent Variable: HLAS
Method: ML - Binary Probit (Quadratic hill climbing)
Date: 01/01/09 Time: 11:37
Sample (adjusted): 1 134
Included observations: 134 after adjustments
Convergence achieved after 3 iterations
covariance matrix computed using second derivatives

Variable	Coefficient	Std. Error	z-Statistic	Prob.
C	0.341942	0.151945	2.250441	0.0244
OBT	-0.441576	0.219341	-2.013193	0.0441

Mean dependent var	0.552239	S.D. dependent var	0.499130
S.E. of regression	0.493363	Akaike info criterion	1.374770
Sum squared resid	32.12967	Schwarz criterion	1.418021
Log likelihood	-90.10960	Hannan-Quinn criter.	1.392346
Restr. log likelihood	-92.14904	Avg. log likelihood	-0.672460
LR statistic (1 df)	4.078885	McFadden R-squared	0.022132
Probability(LR stat)	0.043422		

Obs with Dep=0	60	Total obs	134
Obs with Dep=1	74		

Tabulka 1.1: Vysvětlení volby pomocí obtížnosti

Nebudeme se příliš zabývat tím, co tento výstup znamená. Co nás alespoň orientačně zajímá, je sloupec prob. *s p-hodnotami. K určení této p-hodnoty je používán předchozí sloupec, kde můžeme nalézt hodnoty t-statistiky, ale ty jsou v tomto případě porovnávány s normálním rozdělením, proto se tento sloupec jmenuje* z-Statistic. *Tento postup je ovšem standardní, viz [7].*

Je zřejmé, že náš model nevysvětluje mnoho.

Nyní již budeme moci použít pro vysvětlení volby libovolnou veličinu. Dá se odhadnout ovšem, že některé veličiny by mohly zapříčinit multikolinearitu. Např. spokojen *s* dobrrozh *aj. Vypustíme tedy z našich úvah ty korelované veličiny, které přinášejí méně informace. Za příznak kolinearity budeme považovat to, že korelační koeficient je vyšší než 60%.*

Navíc, jistě se budou velmi lišit výsledky pro jedince, kteří psali obtížný test, od těch, kteří psali jednoduchý. Ukazuje se, že nestačí pouze zahrnout veličinu obt *do modelu, ale je výhodné rozdělit data do dvou skupin.*

$R^2_{McFadden}$ *je po rozdělení o 20% vyšší, než by byl v případě, kdybychom data nedělili. Model s interakcemi nebyl zkoumán.*

Na vybrání jedinců, kteří psali lehký test stačí do příkazového okna EViews napsat: smpl if obt=0

Nejjednodušším způsobem, jak sestrojit binární logit model je v příkazovém okně zadat: binary(d=n) hlas c budostbod ...

Nakonec se ukázal jako nejlepší následující model (viz. tab. 1.2). Nezařadíme do něj veličinu dulez. *Sice kvůli tomu klesne* $R^2_{McFadden}$ *o 2%, jenže klesne např. Schwarzovo kritérium aj.*

Dependent Variable: HLAS
Method: ML - Binary Probit (Quadratic hill climbing)
Date: 01/01/09 Time: 19:47
Sample: 1 134 IF OBT=0
Included observations: 71
Convergence achieved after 6 iterations
covariance matrix computed using second derivatives

Variable	Coefficient	Std. Error	z-Statistic	Prob.
C	7.172953	3.148850	2.277960	0.0227
BUDOSTBOD	-0.220749	0.109288	-2.019889	0.0434
JIST	-0.278195	0.138092	-2.014567	0.0440
MAJOSTBOD	-0.311561	0.134461	-2.317111	0.0205
MENIT	0.351199	0.106497	3.297740	0.0010
ODHOBT	-0.495054	0.193719	-2.555529	0.0106
SPOKOJ	-0.432507	0.156948	-2.755727	0.0059
VYSL	0.364291	0.126192	2.886795	0.0039

Mean dependent var	0.633803	S.D. dependent var	0.485193
S.E. of regression	0.402486	Akaike info criterion	1.113801
Sum squared resid	10.20569	Schwarz criterion	1.368751
Log likelihood	-31.53994	Hannan-Quinn criter.	1.215187
Restr. log likelihood	-46.63995	Avg. log likelihood	-0.444224
LR statistic (7 df)	30.20002	McFadden R-squared	0.323757
Probability(LR stat)	8.73E-05		

Obs with Dep=0	26	Total obs	71
Obs with Dep=1	45		

Tabulka 1.2: Vysvětlení volby pomocí vybraných veličin

Ukažme si ještě jeden nástroj, který nabízí EViews. Je to tabulka předpokládaných hodnot. Pro každé pozorování z našeho výběru dosadí EViews potřebné hodnoty do modelu. Uživatel určí mez useknutí $\in (0;1)$*. Pokud výsledná hodnota pro dané pozorování je vyšší, než mez useknutí, bude odhad závisle proměnné v modelu pro toto pozorování položen jedné. Nechme tabulku nejprve vypsat, pak k ní udělejme diskusi. Za mez useknutí* $\approx C$ *jsme dosadili 0,5.*

	Estimated Equation			Constant Probability		
	Dep=0	Dep=1	Total	Dep=0	Dep=1	Total
P(Dep=1)<=C	18	6	24	0	0	0
P(Dep=1)>C	8	39	47	26	45	71
Total	26	45	71	26	45	71
Correct	18	39	57	0	45	45
% Correct	69.23	86.67	80.28	0.00	100.00	63.38
% Incorrect	30.77	13.33	19.72	100.00	0.00	36.62
Total Gain	69.23	-13.33	16.90			
Percent Ga...	69.23	NA	46.15			

Tabulka 1.3: Tabulka z EViews pomáhající určit, jak je náš model dobrý

Na výstupu se objevují 2 tabulky. Nám ale bude stačit pouze tato. Pro jistotu vysvětlíme některé hodnoty, které se v ní vyskytují. V prvním řádku nalezneme počty pozorování, u nichž byla hodnota závisle proměnné odhadnuta nulou. V prvním sloupci počty pozorování, u kterých hodnota závisle proměnné je skutečně nula.

Dále jsou zajímavé sloupce 5 až 7. Jsou v nich uvedeny hodnoty, jako kdybychom do modelu zahrnuli pouze konstantu. V tomto případě budou samozřejmě hodnoty odhadnuty tou hodnotou, která se ve vzorku vyskytuje častěji.

Zajímavý údaj v této tabulce je v posledním řádku a čtvrtém sloupci. Tato hodnota chce říci, jak moc je náš model zlepšením v porovnání s modelem, kde je jen konstanta. Pro jeho výpočet se vezme množství správně odhadnutých pozorování v našem modelu a odečte se od něho množství správně odhadnutých pozorování v zjednodušeném modelu. Nakonec se toto číslo vydělí počtem špatně odhadnutých pozorování ze zjednodušeného modelu.

Takto by bylo možné udávat míru zlepšení, leč bohužel je nutné k tomuto

výsledku přistupovat s rezervou. Při čtení manuálu EViews 5.1 [6, str. 613 - 615] se nepodařilo nalézt žádnou zmínku o tom, z jakého modelu jsou hodnoty regresandu odhadovány. Nejspíše je použit přímo model, který při výpočtu koeficientů bere všechna pozorování ze vzorku. Jenže pak už nelze predikci považovat za nezávislou na koeficientech.

EViews nicméně umožňuje napsat menší program. My tak učiníme, viz dodatek A.1.1 (tam lze také nalézt stručný popis metody, kterou budeme používat). Použijeme metodu „jackknife.“ Při takovém postupu se nám podařilo odhadnout 51 pozorování korektně. Pokud v modelu použijeme jen konstantu, pak odhadneme pouze 45 pozorování správně. Tj. pokud použijeme výše zmíněnou míru zlepšení, pak náš model, oproti nejjednoduššímu možnému, zlepší situaci o 23 %.

Pokud bychom měli rozsáhlá data, bylo by vhodnější místo časově náročné „jackknife“ metody použít testovací množinu. Jenže tady by mohl být problém s výběrem pozorování určených pro testování, pokud bychom si nebyli jisti, jestli nejsou pozorování seřazena podle nějakého klíče. Kdyby např. byla seřazena podle velikosti nějaké veličiny, potom by bylo nevhodné použít jako testovací množinu první či poslední pozorování. Z tohoto důvodu je v A.1.2 napsán program, který nalezne nějaký prostý náhodný výběr a ten ohodnotí. Ovšem pro naše data nemá velký smysl jej použít.

Proveďme ještě analýzu toho, jak se liší v našem případě modely logit, probit a gompit. Výpočty provedeme v EViews, ale nyní již použijeme tabulku, která se v EViews nepoužívá.

Ve sloupci směr nalezneme derivaci střední hodnoty. Srovnej s (1.4). *Tedy*

$$\frac{\partial \mathsf{E}(y_t|\boldsymbol{x}_{t\cdot},\boldsymbol{\beta})}{\partial x_{ti}} = f(-\boldsymbol{x}_{t\cdot}\boldsymbol{\beta}) \cdot \beta_i.$$

	lineární		probit		logit		gompit	
veličina	**koef**	**směr**	**koef**	**směr**	**koef**	**směr**	**koef**	**směr**
konst.	1,77	-	7,17	-	12,50	-	9,51	-
budostbod	-0,04	-0,04	-0,22	-0,07	-0,40	-0,09	-0,34	-0,11
jist	-0,05	-0,05	-0,28	-0,09	-0,48	-0,11	-0,35	-0,11
majostbod	-0,06	-0,06	-0,31	-0,10	-0,51	-0,12	-0,33	-0,11
menit	0,08	0,08	0,35	0,12	0,62	0,14	0,48	0,15
odhobt	-0,10	-0,10	-0,50	-0,17	-0,87	-0,20	-0,61	-0,20
spokoj	-0,10	-0,10	-0,43	-0,15	-0,75	-0,17	-0,53	-0,17
vysl	0,08	0,08	0,36	0,12	0,62	0,14	0,43	0,14
$f(-\bar{\boldsymbol{x}}^\top\hat{\boldsymbol{\beta}})$	1		0,336		0,230		0,319	

Tabulka 1.4: Odhady koeficientů a směry růstu střední hodnoty

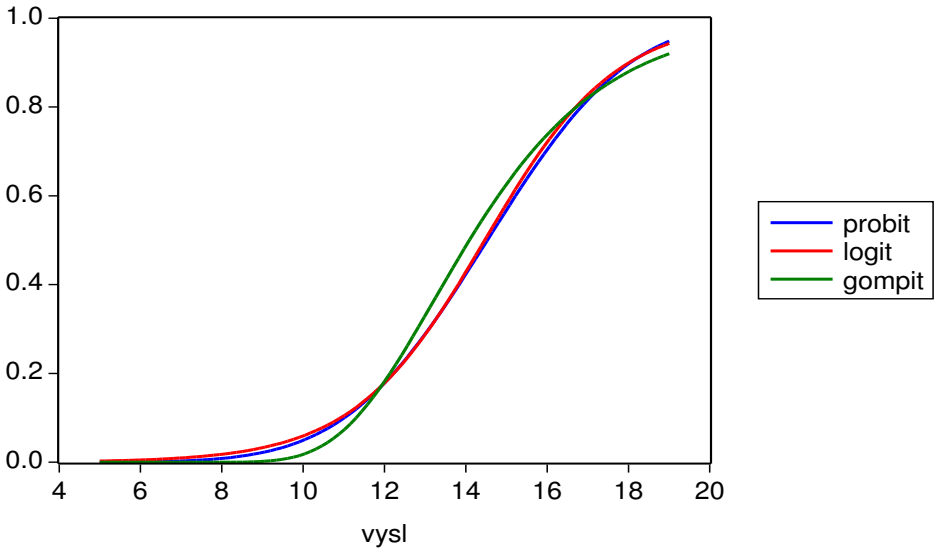
Při výpočtu derivace střední hodnoty za $\boldsymbol{x}$ bereme průměr, jak je naznačeno v posledním řádku. Je opravdu vidět, že odhad gradientu střední hodnoty u modelů logit a probit se příliš neliší. Zdá se, že při daném zaokrouhlení, jako by se odhady v absolutní hodnotě lišily o 0,02. Resp. odhady logit jsou v absolutní hodnotě o 0,02 vyšší. Toto zjištění odpovídá též poznámce 1.2, neboť logit je díky tomu „citlivější“ na změnu nezávisle proměnné při pohybu od průměru. Takže odhadnutá podmíněná střední hodnota by mohla mít větší rozptyl. To odpovídá skutečnosti, že logistické rozdělení má těžší chvosty.

V našem případě se ani model gompit od ostatních příliš neodlišuje.

Jak již bylo výše uvedeno, standardní prostředky EViews podobnou tabulku nevypíší, proto bylo opět třeba napsat drobný program. Na něj se můžeme podívat v A.1.3

Dalším zajímavým zjištěním může být, že pokud použijeme výše zmíněnou „jackknife“ metodu, modely logit a probit odhadnou stejné množství pozorování korektně, tedy 51. Nikoli však model gompit, který odhadne správně 53 pozorování. Přičemž v jednoduchém modelu je stále pouze 45 správně odhadnutých pozorování (nezávisle na volbě modelu, prostě jen volíme tu možnost, která se vyskytne v datech častěji, takže to nemůže záviset na distribuční funkci). Takže v tomto případě dochází ke zlepšení o 31 %.

Podívejme se ještě na to, jak vypadá odhad podmíněné pravděpodobnosti $\mathsf{P}(y_t = 1|\boldsymbol{x}_{t\cdot}, \boldsymbol{\beta}) = 1 - F(-\boldsymbol{x}_{t\cdot}\boldsymbol{\beta})$ za podmínek, že u všech veličin v modelu vezmeme jejich průměr. Pouze veličinu vysl *necháme probíhat od 5 do 19, což je její minimum a maximum.*



Obrázek 1.1: Odhad podmíněné pravděpodobnosti pro modely probit, logit a gompit

Opět můžeme poznamenat, že gompit se od ostatních dvou modelů liší více. Jak vytvořit matici, u které v nabídce View-Graph-XY line-One X against all Y's *lze získat graf jako v 1.1, je uvedeno v části A.1.4. Tento graf je pak možné upravit interaktivně.* △

1.2 Ordinální vysvětlované proměnné

Pokud chceme zobecnit binární vysvětlovanou proměnnou, dostaneme multinomickou vysvětlovanou proměnnou. Ta může nabývat dvou a více hodnot, ale vždy jen konečně mnoha. My se budeme v tomto odstavci zabývat především ordinálními, tj. uspořádanými multinomickými proměnnými. Hodnoty těchto proměnných jsou uspořádány, tzn. lze určit jejich pořadí. Taková veličina může např. určovat v jakých letech nastala nějaká událost, nebo velikost v litrech aj. My budeme předpokládat, že máme multinomickou veličinu s prvky $1, \ldots, R$, jejíž prvky jsou setříděné.

Sestavíme model, u kterého použijeme obdobnou interpretaci jako byla použita v bodě 1 za poznámkou 1.1. Tzn. zavedeme latentní vysvětlovanou

proměnnou y^*, kterou provážeme s regresory $\boldsymbol{X}$ v modelu

$$y_t^* = \boldsymbol{x}_{t\cdot}\boldsymbol{\beta} + \varepsilon_t, \tag{1.7}$$

kde ε_t jsou iid náhodné veličiny s nulovou střední hodnotou. Vztah mezi latentní a skutečnou vysvětlovanou proměnnou má tvar

$$y_t = \begin{cases} 0 & \text{pro } y_t^* \leq m_1, \\ 1 & \text{pro } m_1 < y_t^* \leq m_2, \\ 2 & \text{pro } m_2 < y_t^* \leq m_3, \\ \vdots & \\ R & \text{pro } m_R < y_t^*. \end{cases}$$

Prahy $m_1, \ldots, m_R$ jsou kromě $\beta_1, \ldots, \beta_k$ a reziduálního rozptylu dalšími neznámými parametry modelu. Dále pokračujme následovně

$$\mathsf{P}_r = \begin{cases} \mathsf{P}(y_t = 0|\boldsymbol{x}_{t\cdot}, \boldsymbol{\beta}, \boldsymbol{m}) = F(m_1 - \boldsymbol{x}_{t\cdot}\boldsymbol{\beta}), \\ \mathsf{P}(y_t = 1|\boldsymbol{x}_{t\cdot}, \boldsymbol{\beta}, \boldsymbol{m}) = F(m_2 - \boldsymbol{x}_{t\cdot}\boldsymbol{\beta}) - F(m_1 - \boldsymbol{x}_{t\cdot}\boldsymbol{\beta}), \\ \mathsf{P}(y_t = 2|\boldsymbol{x}_{t\cdot}, \boldsymbol{\beta}, \boldsymbol{m}) = F(m_3 - \boldsymbol{x}_{t\cdot}\boldsymbol{\beta}) - F(m_2 - \boldsymbol{x}_{t\cdot}\boldsymbol{\beta}), \\ \vdots \\ \mathsf{P}(y_t = R|\boldsymbol{x}_{t\cdot}, \boldsymbol{\beta}, \boldsymbol{m}) = 1 - F(m_R - \boldsymbol{x}_{t\cdot}\boldsymbol{\beta}), \end{cases} \tag{1.8}$$

kde $\mathsf{P}_r = \mathsf{P}(y_t = r|\boldsymbol{x}_{t\cdot}, \boldsymbol{\beta}, \boldsymbol{m})$ pro $r = 0, \ldots, R$ a $F(\cdot)$ je distribuční funkce reziduí v (1.7). Obdobně jako v předchozím odstavci můžeme rozlišovat modely logit, probit a gompit. To, že používáme kategorie $0, \ldots, R$, není podstatné. Označení může být libovolné. Důležité je pouze dodržet uspořádání. Tj. $y_s < y_t$ právě tehdy, když $y_s^* < y_t^*$.

Nyní použijeme vyjádření (1.8) pro následující výpočet

$$\frac{\partial \mathsf{P}_r}{\partial x_{ti}} = \frac{\partial F(m_{r+1} - \boldsymbol{x}_{t\cdot}\boldsymbol{\beta})}{\partial x_{ti}} - \frac{\partial F(m_r - \boldsymbol{x}_{t\cdot}\boldsymbol{\beta})}{\partial x_{ti}}, \qquad r = 1, \ldots, R-1.$$

Je vidět, že nemůžeme uvést žádný závěr o vlivu změny regresoru x_{ti} na pravděpodobnost P_t, např. na základě znaménka β_i. Toto můžeme učinit pouze v koncových bodech, jak se snadným výpočtem, za použití (1.8), ověří.

Odhad parametrů v námi zkoumaném modelu provedeme jako v předchozí kapitole metodou maximální věrohodnosti. Odhadujeme tedy parametry $\boldsymbol{\beta}$ a $\boldsymbol{m}$. Reziduální rozptyl musíme opět určit předem. Logaritmická věrohodnostní funkce má tvar

$$L(\boldsymbol{\beta}, \boldsymbol{m}) = \sum_{t=1}^{T} \sum_{r=1}^{R-1} I_r(y_t) \cdot \ln\big(\mathsf{P}(y_t = r|\boldsymbol{x}_{t\cdot}, \boldsymbol{\beta}, \boldsymbol{m})\big), \tag{1.9}$$

kde $I_r(y_t)$ je indikátor toho, zda $y_t = r$. Jako příklad uveďme logaritmickou věrohodnostní funkci modelu logit.

$$\begin{aligned} L(\boldsymbol{\beta}, \boldsymbol{m}) &= \sum_{t=1}^{T} I_0(y_t) \cdot \ln\left(\frac{\mathrm{e}^{m_0 - \boldsymbol{x}_{t\cdot}\boldsymbol{\beta}}}{1 + \mathrm{e}^{m_0 - \boldsymbol{x}_{t\cdot}\boldsymbol{\beta}}}\right) \\ &+ \sum_{t=1}^{T} \sum_{r=0}^{R} I_r(y_t) \cdot \ln\left(\frac{\mathrm{e}^{m_{r+1} - \boldsymbol{x}_{t\cdot}\boldsymbol{\beta}}}{1 + \mathrm{e}^{m_{r+1} - \boldsymbol{x}_{t\cdot}\boldsymbol{\beta}}} - \frac{\mathrm{e}^{m_r - \boldsymbol{x}_{t\cdot}\boldsymbol{\beta}}}{1 + \mathrm{e}^{m_r - \boldsymbol{x}_{t\cdot}\boldsymbol{\beta}}}\right) \\ &+ \sum_{t=1}^{T} I_R(y_t) \cdot \ln\left(1 - \frac{\mathrm{e}^{m_R - \boldsymbol{x}_{t\cdot}\boldsymbol{\beta}}}{1 + \mathrm{e}^{m_R - \boldsymbol{x}_{t\cdot}\boldsymbol{\beta}}}\right). \end{aligned}$$

Stejně jako v předchozí kapitole, můžeme náš model použít pro předpovědi a poté zkoumat jeho úspěšnost.

Příklad 1.2 *Pokračujme ve zkoumání dat z příkladu 1.1. Nyní nám samozřejmě půjde o zkoumání veličiny, která nabývá více než dvou hodnot.*

Půjde nám o veličinu spokoj. *Ze stejných důvodů jako ve výše zmíněném příkladu rozdělíme data do dvou skupin, podle toho, jaký test účastník psal. Budeme se zabývat těmi jedinci, kteří psali jednodušší variantu.*

V EViews necháme model, který má jako regresant uspořádanou multinomickou proměnnou spočítat příkazem ordered(d=n) spokoj hlas plat ... *Jedná se o model probit.*

Asi nikoho příliš nepřekvapí, že v našem modelu jsou velmi důležitými vysvětlujícími proměnnými vysl *a* plat.

Dependent Variable: SPOKOJ
Method: ML - Ordered Probit (Quadratic hill climbing)
Date: 01/01/09 Time: 19:47
Sample: 1 134 IF OBT=0
Included observations: 71
Number of ordered indicator values: 7
Convergence achieved after 6 iterations
covariance matrix computed using second derivatives

	Coefficient	Std. Error	z-Statistic	Prob.
BUDOSTBOD	-0.119766	0.065514	-1.828101	0.0675
BUDDOBR	0.327837	0.148520	2.207356	0.0273
HLAS	-0.862358	0.312098	-2.763102	0.0057
PLAT	0.748324	0.289464	2.585205	0.0097
VYSL	0.313404	0.074171	4.225453	0.0000
		Limit Points		
LIMIT_2:C(6)	1.072513	1.696068	0.632353	0.5272
LIMIT_3:C(7)	2.805511	1.651191	1.699084	0.0893
LIMIT_4:C(8)	3.203369	1.651495	1.939679	0.0524
LIMIT_5:C(9)	3.791485	1.659285	2.285011	0.0223
LIMIT_6:C(10)	4.217714	1.669924	2.525692	0.0115
LIMIT_7:C(11)	5.182405	1.687874	3.070375	0.0021

Akaike info criterion	2.930985	Schwarz criterion	3.281541
Log likelihood	-93.04997	Hannan-Quinn criter.	3.070390
Restr. log likelihood	-113.8691	Avg. log likelihood	-1.310563
LR statistic (5 df)	41.63819	LR index (Pseudo-R2)	0.182834
Probability(LR stat)	6.97E-08		

Tabulka 1.5: Vysvětlení spokojenosti s výsledkem pomocí vybraných veličin

Hodnoty LIMIT_2:C(6) ... *udávají hodnoty* $m_1, \ldots, m_r$ *viz např.* (1.8)

U veličin budostbod *a* buddobr *se projevila zajímavá vlastnost. Pokud jednu z nich z modelu vyjmeme, druhá se následně projeví jako nevýznamná. Tento efekt vyplývá z toho, co obě veličiny představují. To, že budu „dobrý", mě zajímá, pouze pokud se mohu srovnat s ostatními. Za zmínku také stojí, že tyto dvě veličiny nebylo vhodné nahradit poměrem* leps*, který vyjadřuje, zda si účastník myslí, že bude lepší než druzí.*

Kdybychom chtěli v EViews nalézt podobnou tabulku jako v příkladu pro binární vysvětlovanou proměnnou, která by se týkala správných odhadů, nalezli bychom tab. 1.6.

Potíž spočívá v tom, že tato tabulka vůbec nevypovídá o tom, jestli náš model dobře předpovídá. Pouze tvrdí, kolik pozorování bylo odhadnuto hodnotou k a kolik jich ve skutečnosti hodnotu k má. Kvůli tomu je v podstatě čtvrtý sloupec matoucí, protože pozorování může být odhadnuto nějakou hodnotou, ale nemusí ji mít. Tento sloupec je pouze rozdíl dvou předchozích. Není v něm zkoumáno, zda dochází ke skutečné shodě. Např. kdyby třetí po-

Value	Count	Count of obs with Max Prob	Error	Sum of all Probabilities	Error
1	1	1	0	1.056	-0.056
2	5	6	-1	4.981	0.019
3	4	0	4	3.388	0.612
4	8	1	7	7.652	0.348
5	7	0	7	7.516	-0.516
6	19	29	-10	20.011	-1.011
7	27	34	-7	26.397	0.603

Tabulka 1.6: Tabulka z EViews, která by měla vyjadřovat množství správně odhadnutých hodnot

zorování mělo hodnotu 1 a žádné jiné pozorování této hodnoty nenabývalo. Náš model dejme tomu odhadne, že pozorování 5 má mít hodnotu 1 a žádné jiné pozorování již takto neodhadne. Pak je error v prvním řádku 0, ačkoli pozorování 3 je nutně odhanuto špatně. Upozorněme na to, že tento příklad není vybrán z našeho modelu. Tam je pozorování s hodnotu 1 odhadnuto správně. Ale výše popsaná chyba se zde také vyskytuje. Bylo by ovšem komplikovanější na ní ukázat princip.

Podívejme se tedy raději opět na metodu „jackknife" (program viz v A.1.1). Pokud použijeme všech 71 pozorování (tedy ta ze vzorku, pro která je obt=0) nebude možné použít model gompit. Když vyloučíme sto třicáté pozorování, abychom je mohli odhadnout, dostaneme tuto chybovou hlášku: Non positive likelihood function for observation *72. Jde zřejmě o to, že věrohodnostní funkce pro dané koeficienty je při odhadu příliš blízká nule (podrobnější vysvětlení v nápovědě ani v manuálu nelze dohledat). Toto by mohlo znamenat, že pozorování 72 je odlehlé. Tímto způsobem je také možné zkoumat odlehlá pozorování obecně.*

Takže metodu „jackknife" pro tento vzorek a model gompit nelze použít. Pro logit je správných odhadů 28, přičemž nejčastější hodnota veličiny spokoj *se ve vzorku vyskytne 27 krát. Tedy od strategie, kde bychom vybírali pouze nejčastější hodnotu, nejde o veliké zlepšení. Model logit dopadne ještě hůře. V tomto případě odhadneme správně jen 27 pozorování.*

Pokud ovšem z našeho vzorku vyřadíme 72 pozorování, dostaneme jiné výsledky. Pro modely probit a gompit je správně určeno 30 pozorování. Jedná

se o zlepšení proti nejjednodušší strategii o 7 %. Ale pro logit ke zlepšení nedošlo a správně je pouze 27 pozorování.

Pokusme se nalézt další ukazatel, který by nám mohl pomoci při určování vhodnosti modelu. V prvé řadě je dobré si připomenout, že použitá metoda nedokáže odlišit rozdíl mezi tím, zda se dvě sousední hodnoty liší o 1, nebo o 100. Tedy nezáleží na tom, pokud má ordinální veličina 3 hodnoty, jestli to jsou hodnoty 1, 2 a 3, nebo -5, 0 a 100. Toto je způsobeno tvarem funkce, kterou minimalizujeme, tedy logaritmickou věrohodnostní fcí viz (1.9)*. Žádná hodnota vysvětlované proměnné se zde nevyskytuje.*

Abychom vystihli toto chování modelu, budeme počítat následující statistiku

$$\sum_{i=1}^{T} |i_t - \hat{i_t}|, \tag{1.10}$$

kde i_t vyjadřuje kolikátou hodnotu pozorování t má. Tedy pokud závisle proměnná nabývá hodnot -5, 0 a 11 a jestliže $y_t = 0$, pak $i_t = 2$. Přitom $\hat{i_t}$ je odhad i_t za pomoci použitého modelu.

Potíž této statistiky spočívá v tom, že pokud bude model odhadovat více pozorování prostřední hodnotou, nejspíš vyjde nižší, tedy lepší. Proto také používáme absolutní hodnotu místo druhé mocniny, která by tuto okolnost ještě více podtrhla.

Samozřejmě by v jistých případech mohlo být také výhodné použít $\sum_{i=1}^{T} |y_t - \hat{y}_t|$. Přitom vzdálenosti mezi hodnotami vysvětlované proměnné by byly voleny tak, aby vystihovaly skutečný odstup mezi hodnotami. Např. by byla kategorie lehké zranění, což by mělo hodnotu 1, střední zranění s hodnotou 2 a zranění s následkem smrti s hodnotou 20. Pokud by někdo chtěl použít výše zmíněnou statistiku jako vedlejší kritérium pro výběr modelu, zřejmě by taková statistika preferovala modely, které nedělají chybu v zařazování do třetí skupiny, ale už by tolik nezáleželo na tom, jestli tento model zařadí člověka do skupiny lehké nebo střední zranění.

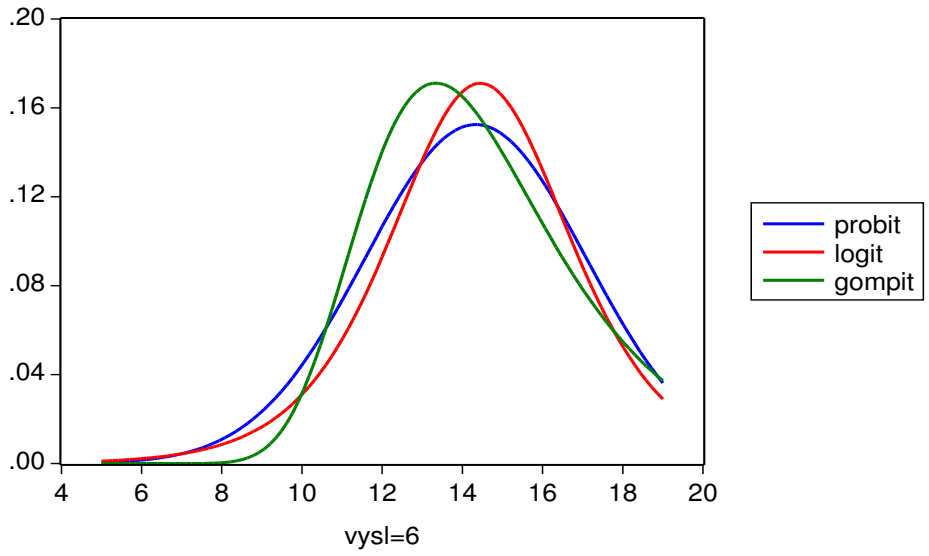
V našem případě ovšem není podstatné, zda použijeme tuto statistiku nebo (1.10)*, poněvadž hodnoty vysvětlované proměnné mají od sebe vzdálenost 1.*

Kdybychom měli hodnotu, která je v celých datech jen jedna (a my takovou máme), pak ji, po jejím vyloučení, model není schopen odhadnout.
V tomto případě ale stejně vybereme hodnotu, která má nejvyšší pravděpodobnost, i když je nutně jiná, než ta, kterou odhadujeme.

V případě nejjednodušší strategie má tato statistika hodnotu 102. Pro model probit 71, logit 68 a gompit 71. Je vidět, že ač model logit se nejméně

často trefí do správného pozorování, jeho odhady jsou vzdáleny nejméně od skutečných hodnot (ve smyslu dříve zmíněné statistiky). Toto může být způsobeno tím, že vysoké hodnoty 6 a 7 se ve vzorku vyskytují nejčastěji, takže modely, které často tyto hodnoty odhadují, se „trefují" častěji. Nicméně se častěji mýlí u pozorování s nízkými hodnotami, než model jehož hodnoty jsou více ve středu (v našem případě logit).

Nyní se podívejme na to, jak vypadá graf odhadu podmíněných pravděpodobností pro modely probit, logit a gompit. Kromě veličiny vysl *bereme jako hodnotu všech ostatních veličin průměr. U veličiny* vysl *dosazujeme 101 hodnot od minima po maximum a vzdálenosti bereme jako ekvidistantní. Samozřejmě, pokud bychom nebrali průměry, ale jinou hodnotu u ostatních veličin, dostali bychom grafy odlišné.*



Obrázek 1.2: Odhad podmíněné pravděpodobnosti pro modely probit, logit a gompit

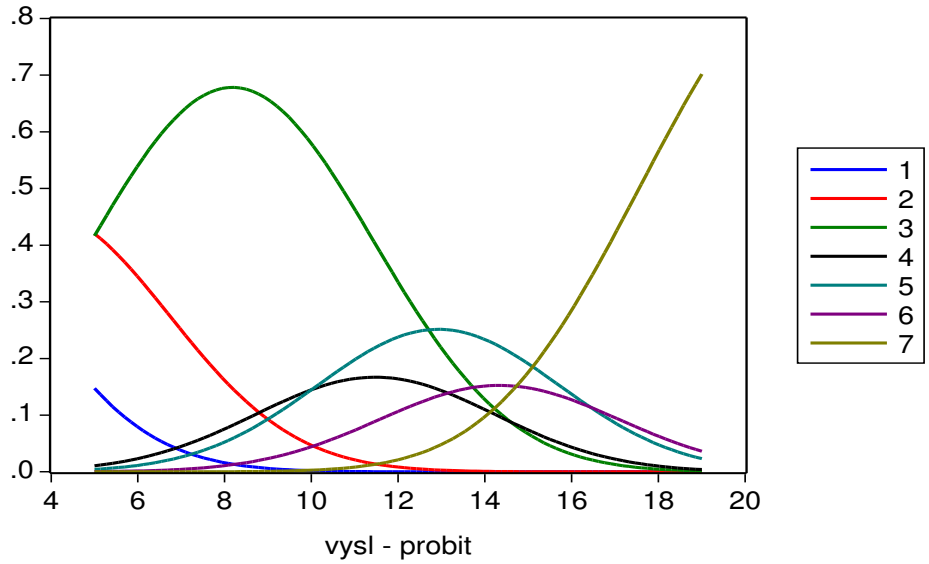
Vidíme, že v tomto případě odhadované pravděpodobnosti nenabývají ani 0.2, což je dáno právě také tím, že od většiny veličin bereme pouze průměr.

Také si z obrázku můžeme všimnout, že se model gompit opět trochu více odlišuje od ostatních dvou. Nicméně v tomto případě kupodivu dávají modely probit a gompit srovnatelnější výsledky než logit. Ovšem, jestliže se podíváme

na odhady, které udělají jednotlivé modely (v našem modelu odhad_v*), zjistíme, že ne nutně se odhady v modelech gompit a probit liší více od odhadů v modelu logitu.*

Grafy pro ostatní hodnoty vypadají obdobně.

Ještě si ukažme graf podmíněných pravděpodobností pro všechny hodnoty. Budeme uvažovat model probit.



Obrázek 1.3: Odhad podmíněných pravděpodobností pro všechny hodnoty veličiny vysl u modelu probit

Z obrázku se zdá, jako kdyby náš model odhadoval pouze hodnoty 3, 5 a 7. To ovšem není pravda. My opět necháváme všechny ostatní veličiny, kromě vysl *nabývat pouze své průměrné hodnoty. Kdyby nabývaly hodnoty jiné, jistě by i podmíněné pravděpodobnosti vypadaly jinak.*

Také je dobré poznamenat, že medián je 17. Takže, i když v grafu by nejvíce hodnot veličiny vysl *bylo odhadnuto trojkou, nejvíce pozorování bude odhadnuto sedmičkou, jelikož tam se nachází většina pozorování.*

△

1.3 Neuspořádané diskrétní vysvětlované proměnné

V případě, kdy zkoumáme multinomickou proměnnou, která nabývá hodnot $r = 1, \ldots, R$, ale nemá žádné explicitní uspořádání, nemůžeme model zmíněný v předchozím odstavci použít. Takovouto veličinou může např. být barva automobilu, typ spoření aj.

Zaměřme se nyní podrobněji na logit model pro takovýto typ proměnné. Parametry $\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_R$ v tomto případě odpovídají tomu, že každá hodnota proměnné má svůj vektor parametrů. Předpokládejme, že $\mathsf{P}_1, \ldots, \mathsf{P}_R$ jsou pravděpodobnosti týkající se kategorií $1, \ldots, R$. Tj. jsou to nezáporné funkce proměnné $\boldsymbol{x}$, a $\boldsymbol{\beta}_j$ jejichž hodnoty leží mezi nulou a jedničkou. Součet všech dá jedničku.

Myšlenka spočívá v tom, že se pokusíme úlohu transformovat na binární případ. Budeme podmiňovat pravděpodobnost toho, že nastane j podmínkou, že nastane j či R

$$F(\boldsymbol{x}_{t\cdot}\boldsymbol{\beta}_j) = \frac{\mathsf{P}_j}{\mathsf{P}_j + \mathsf{P}_R} \quad \text{pro} \quad j = 1, \ldots, R-1,$$

kde $F(\cdot)$ je nějaká vhodná distribuční funkce stejně jako $G(\cdot)$. Takže dále položme

$$G(\boldsymbol{x}_{t\cdot}\boldsymbol{\beta}_j) = \frac{\mathsf{P}_j}{\mathsf{P}_R} = \frac{F(\boldsymbol{x}_{t\cdot}\boldsymbol{\beta}_j)}{1 - F(\boldsymbol{x}_{t\cdot}\boldsymbol{\beta}_j)} \quad j = 1, \ldots, R-1. \tag{1.11}$$

Protože

$$\sum_{j=1}^{R-1} \frac{\mathsf{P}_j}{\mathsf{P}_R} = \frac{1 - \mathsf{P}_R}{\mathsf{P}_R} = \frac{1}{\mathsf{P}_R} - 1,$$

dostaneme

$$\mathsf{P}_R = \left(1 + \sum_{j=1}^{R-1} G(\boldsymbol{x}_{t\cdot}\boldsymbol{\beta}_j)\right)^{-1}, \tag{1.12}$$

z čehož a podle (1.11)

$$\mathsf{P}_j = \frac{G(\boldsymbol{x}_{t\cdot}\boldsymbol{\beta}_j)}{\left(1 + \sum_{i=1}^{R-1} G(\boldsymbol{x}_{t\cdot}\boldsymbol{\beta}_i)\right)^{-1}}. \tag{1.13}$$

V podstatě je možné za distribuční funkci $F(\cdot)$ vzít řadu distribučních funkcí. Avšak z výpočetních důvodů bude nejjednodušší použít rozdělení logistické. Tedy za F dosadíme distribuční funkci logistického rozdělení a pak už jen jednoduchým výpočtem z (1.11) dostaneme, že $G(\boldsymbol{x}_{t\cdot}\boldsymbol{\beta}_j) = \exp(\boldsymbol{x}_{t\cdot}\boldsymbol{\beta}_j)$. Dosaďme do (1.12) a (1.13)

$$\mathsf{P}_j = \frac{\mathrm{e}^{\boldsymbol{x}_{t\cdot}\boldsymbol{\beta}_j}}{D} \qquad j = 1, \ldots, R-1,$$

$$\mathsf{P}_R = \frac{1}{D}, \qquad \text{kde} \qquad D = 1 + \sum_{j=1}^{R-1} \mathrm{e}^{\boldsymbol{x}_{t\cdot}\boldsymbol{\beta}_j}\,. \tag{1.14}$$

Tento model se nazývá *multinomický model logit (multinomial logit model).*

Pro praktické odhady parametrů $\boldsymbol{\beta}_j$ použijeme opět metodu maximální věrohodnosti. Připomeňme, že máme pozorování $t = 1, \ldots, T$. P_{tj} znamená, že pozorování t padne do kategorie j, tzn. $\mathsf{P}(y_t = j)\mathsf{P}_{tj}$.

Také budeme používat indikátor I_{tj}, který nabývá hodnoty 1, pokud pozorování t nabyde hodnoty j. V ostatních případech je tato funkce nulová.

Potom věrohodnostní funkci můžeme psát ve tvaru

$$l = \prod_{t=1}^{T} \mathsf{P}_{t1}^{I_{t1}} \cdots \mathsf{P}_{tR}^{I_{tR}}.$$

Logaritmickou věrohodnostní funkci

$$L = \ln(l) = \sum_{t=1}^{T} \sum_{j=1}^{R} I_{tj} \ln \mathsf{P}_{tj} \tag{1.15}$$

rozepíšeme za pomoci (1.14)

$$\mathsf{P}_{tj} = \frac{\mathrm{e}^{\boldsymbol{x}_{t\cdot}\boldsymbol{\beta}_j}}{1 + \sum_{i=1}^{R-1} \mathrm{e}^{\boldsymbol{x}_{t\cdot}\boldsymbol{\beta}_i}} \quad \text{pro} \quad j = 1, \ldots, R-1$$

a

$$\mathsf{P}_{tR} = \frac{1}{1 + \sum_{i=1}^{R-1} \mathrm{e}^{\boldsymbol{x}_{t\cdot}\boldsymbol{\beta}_i}}.$$

Takže P_{tj} je již funkcí pouze všech $\boldsymbol{\beta}_i$, kde $i = 1, \ldots, R-1$. Abychom mohli spočítat maximum, musíme spočíst první derivace

$$\frac{\partial \mathsf{P}_{tj}}{\partial \boldsymbol{\beta}_j} = \mathsf{P}_{tj}(1 - \mathsf{P}_{tj})\boldsymbol{x}_t \quad j = 1, \ldots, R-1,$$

$$\frac{\partial \mathsf{P}_{tj}}{\partial \boldsymbol{\beta}_k} = -\mathsf{P}_{tj}\mathsf{P}_{tk}\boldsymbol{x}_t \quad j,k = 1,\ldots,R-j \quad k \neq j,$$

$$\frac{\partial \mathsf{P}_{tj}}{\partial \boldsymbol{\beta}_R} = 0 \quad j = 1,\ldots,R-1,$$

$$\frac{\partial \mathsf{P}_{tR}}{\partial \boldsymbol{\beta}_j} = -\mathsf{P}_{tj}\mathsf{P}tR\boldsymbol{x}_t \quad j = 1,\ldots,R.$$

Nyní dosadíme do logaritmické věrohodnostní funkce a pro $k = 1,\ldots,R-1$ dostaneme

$$\begin{aligned}
\frac{\partial L}{\partial \boldsymbol{\beta}_k} &= \sum_{t=1}^{T}\left[\frac{I_{tk}}{\mathsf{P}_{tk}}\mathsf{P}_{tk}(1-\mathsf{P}_{tk}) + \sum_{\substack{j=1\\ j\neq k}}^{R}\frac{I_{tj}}{\mathsf{P}_{tj}}(-\mathsf{P}_{tj}\mathsf{P}_{tk})\right]\boldsymbol{x}_t \\
&= \sum_{t=1}^{T}\left[I_{tk} - \mathsf{P}_{tk}\left(\sum_{j=1}^{R} I_{tj}\right)\right]\boldsymbol{x}_t \\
&= \sum_{t=1}^{T}(I_{tk} - \mathsf{P}_{tk})\boldsymbol{x}_t, \qquad (1.16)
\end{aligned}$$

neboť $\sum_{j=1}^{R} I_{tj} = 1$. Budeme tedy řešit následující rovnici

$$\sum_{t=1}^{T}(I_{tk} - \mathsf{P}_{tk})\boldsymbol{x}_t = 0 \quad \text{pro} \quad k = 1,\ldots,R-1 \qquad (1.17)$$

a P_R dopočteme ze vztahu $\sum_{j=1}^{R}\mathsf{P}_j = 1$. Pro účely příslušné optimalizační procedury proveďme následující analýzu:

$$\frac{\partial^2 L}{\partial^2 \boldsymbol{\beta}_k} = -\sum_{t=1}^{T}\mathsf{P}_{tk}(1-\mathsf{P}_{tk})\boldsymbol{x}_t\boldsymbol{x}_t^{\top} \quad k = 1,\ldots,R-1 \qquad (1.18)$$

Rovnice (1.18) je bohužel nelineární, neboť P_{tk} je nelineární funkcí pro všechna $\boldsymbol{\beta}_j$, kde $k = 1,\ldots,R$ a $j = 1,\ldots,R-1$.

Protože matice (1.19) je viditelně negativně semidefinitní, můžeme zaručit existenci jediného maxima

$$\frac{\partial^2 L}{\partial \boldsymbol{\beta}_k \partial \boldsymbol{\beta}_l^\top} = \sum_{t=1}^{T} \mathsf{P}_{tk}\mathsf{P}_{tl}\boldsymbol{x}_t\boldsymbol{x}_t^\top \quad k,l = 1,\ldots,R-1 \quad k \neq l. \tag{1.19}$$

Můžeme tedy řešit naši úlohu Newton-Raphsonovou metodou, či nějakou metodou pro hledání maxima v případě nelineárního programování.

Kapitola 2

Omezené vysvětlované proměnné

V této kapitole budeme pracovat s modelem

$$y_t^* = \boldsymbol{x}_{t.}\boldsymbol{\beta} + \sigma \cdot \varepsilon_t, \tag{2.1}$$

kde y^* bude latentní vysvětlovaná proměnná, kterou budeme mít možnost pozorovat skrz nějakou jinou proměnnou y.

Zaměříme se na spojité vysvětlované proměnné, z jejichž dat nemusíme vyčíst úplnou informaci. Budeme přitom uvažovat jejich dva speciální případy, a to cenzorované a useknuté vysvětlované proměnné.

Pokud by se čtenář zajímal o teoretický rámec problematiky, můžeme doporučit knihu [10] (ta je vhodná i pro diskrétní vysvětlované proměnné).

2.1 Cenzorované veličiny

U cenzorované veličiny můžeme pozorovat hodnoty pouze z nějakého intervalu. Pokud by měla nabýt hodnotu mimo tento interval, tak bychom dostali okrajové hodnoty intervalu.

Zapišme toto formálně.

Definice 2.1 (Cenzorovaná veličina) *Mějme spojitou latentní veličinu y^*. My ovšem pozorujeme pouze veličinu y. Dále mějme meze $d_t < h_t$ pro každé pozorování t. Pak cenzorovanou veličinou nazveme veličinu, která se chová následovně*

$$y_t = \begin{cases} d_t & pro\ y_t^* \leq d_t, \\ y_t & pro\ d_t < y_t^* \leq h_t \\ h_t & pro\ h_t \geq y_t^*. \end{cases}$$

Velmi často platí $d_t = d$ a $h_t = h$ pro všechna $t \in T$. Pak označíme meze jen d a h.

Pokud $d = -\infty$ říkáme, že neprovádíme cenzorování zleva. Pokud $h = \infty$ neprovádíme cenzorování zprava.

Jde tedy o to, že pozorování, která leží mimo jistou mez, jsou v datech reprezentována touto mezí.

Lze namítnout, jestli by nebylo lepší takováto pozorování úplně ze vzorku vyloučit, jenže tím bychom ztratili velkou část informace.

S takovýmto druhem veličin se můžeme setkat v případech, kdy jsou z nějakého důvodu příliš velké hodnoty nepublikovatelné, např. vnitřní předpisy nějaké organizace mohou zakazovat zveřejnění platů nad určitou mezí.

Nebo pokud náš přístroj, kterým měříme nějakou fyzikální veličinu, má vymezený rozsah a pozorování za hranicí tohoto rozsahu ohodnotí hraniční hodnotou.

Můžeme si položit otázku, kdy cenzorování nastává a kdy ne. Uvažujme např. veličinu, která měří vzdálenost. Je sice pravda, že bychom si mohli říci, že $d = 0$ a $h = \infty$, jenže takovýto postup by nebyl příliš přirozený. Rozumnější je předpokládat, že takováto veličina má hustotu, jež je nulová pro záporná čísla.

Na druhou stranu v knize [4] je uveden příklad. Nějaký fond nabídne klientům nový investiční produkt. Zajímavá je samozřejmě výše investice, kterou jednotliví klienti uskuteční. Samozřejmě většina klientů vůbec neinvestuje. V tuto chvíli by ovšem záporná investice mohla mít rozumnou interpretaci, proto je možné v tomto případě cenzorovanou proměnnou použít.

Důležitý případ nastane, když $d = 0, h = \infty$, pak je vysvětlovaná proměnná nezáporná. Pokud je v tomto případě navíc reziduální složka normálně rozdělena, mluvíme o *modelu tobit.* Viz [13].

Podívejme se nyní, jak je to s marginálními efekty u modelu tobit. Provedeme několik pomocných výpočtů.

$$\mathsf{E}(y_t|y_t > 0) = \mathsf{E}(y_t^*|y_t^* > 0) = \boldsymbol{x}_{t.}\boldsymbol{\beta} + \sigma\mathsf{E}(\varepsilon_t|\varepsilon_t > -\boldsymbol{x}_{t.}\boldsymbol{\beta}/\sigma) = \boldsymbol{x}_{t.}\boldsymbol{\beta} + \sigma\frac{\varphi(\boldsymbol{x}_{t.}\boldsymbol{\beta}/\sigma)}{\Phi(\boldsymbol{x}_{t.}\boldsymbol{\beta}/\sigma)}$$

Toto plyne z toho, že $\mathsf{E}(u|u > -c) = \frac{\varphi(c)}{\Phi(c)}$ pro u $N(0,1)$, což dostaneme jako výsledek integrování podmíněné hustoty.

Označme $\varphi_t = \varphi(\boldsymbol{x}_{t\cdot}\boldsymbol{\beta}/\sigma)$ je hustota $N(0,1)$ v daném bodě a $\Phi_t = \Phi(\boldsymbol{x}_{t\cdot}\boldsymbol{\beta}/\sigma)$ je distribuční funkce $N(0,1)$.

Nyní odvodíme vzorec pro střední hodnotu

$$\begin{aligned}
\mathsf{E}(y_t) &= \mathsf{P}(y_t > 0) \cdot \mathsf{E}(y_t|y_t > 0) + \mathsf{P}(y_t = 0) \cdot \mathsf{E}(y_t|y_t = 0) \\
&= \Phi_t \cdot \left(\boldsymbol{x}_{t\cdot}\boldsymbol{\beta} + \sigma \frac{\varphi_t}{\Phi_t} \right) + (1 - \Phi_t) \cdot 0 \\
&= \Phi_t \cdot \boldsymbol{x}_{t\cdot}\boldsymbol{\beta} + \varphi_t \sigma. \qquad (2.2)
\end{aligned}$$

Z vět o derivaci a integrálu lze odvodit

$$\frac{\partial \Phi(\alpha/\lambda)}{\partial \alpha} = \frac{1}{\lambda} \varphi(\alpha/\lambda)$$

a také

$$\frac{\partial \varphi(\alpha/\lambda)}{\partial \alpha} = -\frac{\alpha}{\lambda^2} \varphi(\alpha/\lambda).$$

Odsud už snadno dospějeme k

$$\frac{\partial \mathsf{E}(y_t)}{\partial x_{ti}} = \beta_i \Phi_t.$$

Odhad parametrů σ a $\boldsymbol{\beta}$ se provede metodou maximální věrohodnosti. Provádí se tedy maximalizací logaritmické věrohodnostní funkce tvaru

$$\begin{aligned}
l(\boldsymbol{\beta}, \sigma) = \sum_{i=1}^{T} \Bigg\{ & I_{(-\infty, d_t\rangle}(y_t) \cdot \ln F \left(\frac{d_t - \boldsymbol{x}_{t\cdot}\boldsymbol{\beta}}{\sigma} \right) + \\
& I_{(d_t, h_t)}(y_t) \cdot \ln f \left(\frac{y_t - \boldsymbol{x}_{t\cdot}\boldsymbol{\beta}}{\sigma} \right) - I_{(d_t, h_t)}(y_t) \cdot \ln(\sigma) \\
& I_{\langle h_t, \infty)}(y_t) \cdot \ln \left(1 - F \left(\frac{h_t - \boldsymbol{x}_{t\cdot}\boldsymbol{\beta}}{\sigma} \right) \right) \Bigg\},
\end{aligned}$$

kde f a F jsou jako obvykle hustota a distribuční funkce daného rozdělení.

Také uvedeme logaritmickou věrohodnostní funkci pro model tobit

$$l(\boldsymbol{\beta},\sigma) = \sum_{i=1}^{T}\left\{ I_{(-\infty,0\rangle}(y_t)\cdot\ln\left(\Phi\left(\frac{\boldsymbol{x}_{t\cdot}\boldsymbol{\beta}}{\sigma}\right)\right) + I_{[0,\infty)}(y_t)\cdot\ln\left(-\frac{1}{2}\ln(2\pi) - 2\ln(\sigma) - \frac{1}{2\sigma^2}(y_t-\boldsymbol{x}_{t\cdot}\boldsymbol{\beta})^2\right)\right\}.$$

Podle (2.2) můžeme napsat vzorec pro odhad cenzorované veličiny v tomto modelu

$$\hat{y}_t = \mathsf{E}(y_t|\boldsymbol{x}_{t\cdot},\hat{\boldsymbol{\beta}},\hat{\sigma}) = \Phi(\boldsymbol{x}_{t\cdot}\hat{\boldsymbol{\beta}}/\hat{\sigma})\cdot\boldsymbol{x}_{t\cdot}\hat{\boldsymbol{\beta}} + \hat{\sigma}\cdot\varphi(\boldsymbol{x}_{t\cdot}\hat{\boldsymbol{\beta}}/\hat{\sigma}),$$

kde $\hat{\boldsymbol{\beta}}$ a $\hat{\sigma}$ jsou odhady $\boldsymbol{\beta}$ a σ.

2.2 Useknuté veličiny

Ještě se stručně zabývejme useknutými veličinami.

Rozdíl od cenzorovaných spočívá v tom, že pozorování, která překročí danou mez, nemáme. Tedy pozorování vně nějakého intervalu se ve vzorku nevyskytnou.

Definice 2.2 (Useknutá veličina) *Pokud platí* (2.1)*, a* $d_t < h_t$*, potom useknutou vysvětlovanou veličinou nazveme proměnnou, kterou pozorujeme pouze v případě, kdy* $d_t < y_t^* < h_t$ *a pokud je toto splněno, potom* $y_t = y_t^*$*.*

S tímto typem veličin se můžeme například setkat, pokud je zakázáno publikovat data nad určitou hodnotou. Pokud je tato mez překročena, pak jsou data vyloučena.

Jestliže je měřená veličina vně intervalu, který je schopen změřit nějaký přístroj a navíc pokud toto nastane a my nejsme schopni určit příčinu, zda jde o překročení zleva, zprava, nebo chybu měření, musíme takto naměřená pozorování chápat jako useknutá.

Někdy by také mohlo mít použití dat s useknutou vysvětlovanou proměnnou praktické důvody. Kdybychom například chtěli vytvořit model jen pro určitou část klientů, u nichž se vysvětlovaná proměnná nachází v nějakém intervalu. Odůvodněním by mohlo být to, že klienti vně tohoto intervalu jsou pro daný model odlehlá pozorování a bylo by vhodnější pro jednotlivé třídy klientů vytvořit samostatný model.

Odhad parametrů se opět provede metodou maximální věrohodnosti. Uveďme tedy logaritmickou věrohodnostní funkci

$$l(\boldsymbol{\beta},\sigma) = \sum_{\substack{t=1 \\ d_t<y_t^*<h_t}}^{T} \left\{ \ln f\left(\frac{y_t - \boldsymbol{x}_{t\cdot}\boldsymbol{\beta}}{\sigma}\right) - \ln(\sigma) - \ln\left(F\left(\frac{h_t - \boldsymbol{x}_{t\cdot}\boldsymbol{\beta}}{\sigma}\right) - F\left(\frac{d_t - \boldsymbol{x}_{t\cdot}\boldsymbol{\beta}}{\sigma}\right)\right)\right\}.$$

2.3 Proměnné vyjadřující dobu trvání

Stručný náhled do této problematiky najdeme například v práci [11] a nebo [12]. Zajímavá je skutečnost, že většina knih popisujících tuto problematiku má medicínskou tématiku nebo se zabývá teorií spolehlivosti. Užitečným zdrojem informací může být také článek [5]. Stručnou zmínku také můžeme nalézt v [4, str. 182-184], či [7, 791-801].

Jde tedy především o to, že vysvětlovaná proměnná vyjadřuje čas. Zkoumáme, kdy dojde k nějaké události. Odtud také pochází jméno, které se používá pro tento druh analýz, tím je *analýza přežití (survival analysis.)* V tomto případě se zkoumá, za jak dlouho daný jedinec zemře. Samozřejmě lze také zkoumat dobu, kdy nějaký předmět přestane sloužit (rozbije se stroj, praskne žárovka).

Jistě lze takto zkoumat problémy, u kterých pouze předpokládáme, že musí jednou skončit. Typickým příkladem může být čas do chvíle, kdy klient přestane splácet úvěr. Nicméně může to také být doba, po kterou je nějaký jedinec nezaměstnán nebo čas do prodeje nějakého výrobku.

Z výše uvedených příkladů vyplývá, že se může často stát, že dostaneme data, která jsou shora cenzorována. Tedy dostaneme pozorování, u kterých daný jev ještě nenastal. Samozřejmě můžeme taková data ze vzorku vyloučit, ale to by třeba právě u klientů banky, u kterých zkoumáme, kdy přestanou splácet úvěr, byla příliš velká ztráta informace (lze předpokládat, že většina dlužníků svůj dluh splatí).

Nyní se zaměřme na teoretický rámec. Především zkoumáme *funkci přežití (survival function)*, která říká, jaká je pravděpodobnost toho, že dané pozorování překročí nějaký čas. Mějme pozorování času $y_1, \ldots, y_T$. Tato pozorování nechť jsou iid s hustotou f

a distribuční funkcí F. Pak funkci přežití definujeme jako

$$S(\tau) = \mathsf{P}(y_t > \tau) = 1 - F(\tau).$$

Další používanou charakteristikou je *intenzita úmrtnosti (hazard rate, mortality rate).* Tato veličina vyjadřuje změnu podmíněné pravděpodobnosti toho, že jev nastane při malé změně času. Zapíšeme ji tedy takto

$$\lambda(\tau) = \lim_{\delta \to 0^+} \frac{\mathsf{P}(\tau < y_t \leq \tau + \delta | y_t > \tau)}{\delta}.$$

Touto funkcí je již jednoznačně určeno rozdělení, neboť existuje jednoznačný vztah mezi ní a hustotou. Platí totiž vztahy

$$\lambda(\tau) = \lim_{\delta \to 0^+} \frac{F(\tau + \delta) - F(\tau)}{\delta} \frac{1}{S(\tau)} = \frac{f(\tau)}{S(\tau)} = -\frac{\partial \log S(\tau)}{\partial \tau}. \tag{2.3}$$

Další vztah, který lze v tomto kontextu použít jako argument, je

$$S(\tau) = \exp\left[-\int_0^\tau \lambda(s)\, ds\right]. \tag{2.4}$$

Nyní se podíváme, jak použít teorii k cenzorovaným proměnným v našem případě. Budeme odhadovat parametry metodou maximální věrohodnosti a budeme vycházet z (2.3). Nicméně, aby zápis byl úspornější, budeme pracovat místo s logaritmickou věrohodnostní funkcí s věrohodnostní funkcí samotnou. K dalším výpočtům použijeme vztahy (2.3) a (2.4). Případ $c_t = 0$ znamená, že je pozorování t cenzorováno. Pokud $c_t = 1$, pak není. V případě cenzorování je horní mezí $h_t = y_t$. Budeme tedy ve vzorku používat latentní vysvětlovanou proměnnou y_t^*, která není cenzorována a tedy představuje vždy skutečnou dobu do dané události.

Pro ujasnění situace: v tuto chvíli máme veličinu, která je cenzorována zprava a nabývá pouze kladných hodnot (zleva cenzorována není). Věrohodnostní funkce má tvar

$$\begin{aligned} L &= \prod_{t=1}^{T} f(y_t^*)^{c_t} \mathsf{P}(y_t^* > h_t)^{1-c_t} = \prod_{t=1}^{T} f(y_t^*)^{c_t} (1 - F(h_t))^{1-c_t} \\ &= \prod_{t=1}^{T} \lambda(y_t)^{c_t} (1 - F(h_t)) = \prod_{t=1}^{T} \lambda(y_t)^{c_t} \exp\left[-\int_0^{h_t} \lambda(u)\, du\right]. \end{aligned}$$

Ještě jsme ovšem nepoužili exogenní veličiny. Obecně budeme předpokládat vztah

$$\ln(y_t) = \boldsymbol{x}_{t\cdot}\boldsymbol{\beta} + \sigma\varepsilon_t. \tag{2.5}$$

Jakobian při transformaci z ε_t na $\ln(y_t)$ je $\frac{\partial\varepsilon_t}{\partial\ln(y_t)} = \frac{1}{\sigma}$. Takže hustotu budeme psát ve tvaru

$$f(\ln(y_t)|\boldsymbol{x}_{t\cdot},\boldsymbol{\beta},\sigma) = \frac{1}{\sigma}f\left(\frac{\ln(y_t)-\boldsymbol{x}_{t\cdot}\boldsymbol{\beta}}{\sigma}\right)$$

a funkci přežití ve tvaru

$$S(\ln(y_t)|\boldsymbol{x}_{t\cdot},\boldsymbol{\beta},\sigma) = S\left(\frac{\ln(y_t)-\boldsymbol{x}_{t\cdot}\boldsymbol{\beta}}{\sigma}\right).$$

Tyto funkce pak budeme dosazovat do modelů, kde používáme různé hustoty. Příklady takových modelů nyní uvedeme.

1. *Exponenciální model* doby trvání má intenzitu úmrtnosti

   $$\lambda(\tau) = \gamma,$$

   která odpovídá funkci přežití ve tvaru $f(\tau) = \gamma\exp(-\gamma\tau)$.

2. Velmi častý je *Weibullův model* s intenzitou

   $$\lambda(\tau) = \alpha\gamma\tau^{\alpha-1}.$$

   Jeho hustotu zapíšeme ve tvaru $f(\tau) = \alpha\gamma\tau^{\alpha-1}\exp(-\gamma\tau^{\alpha})$. Pokud položíme $\alpha := 1$, pak dostaneme exponenciální model.

3. *Logaritmicko-normální model*

   $$\lambda(\tau) = \phi\left(\frac{\ln\tau}{\sigma}\right)\Big/\sigma\tau\left(1-\Phi\left(\frac{\ln\tau}{\sigma}\right)\right).$$

   Veličina $\ln(y_t)$ má normální rozdělení s parametry μ a σ^2.

4. *Model s proporcionální intenzitou úmrtnosti,* resp. *Coxův model*

   $$\lambda_t(\tau) = \lambda_0(\tau)\exp(\boldsymbol{x}_{t\cdot}\boldsymbol{\beta}),$$

kde $\lambda_0(\tau)$ je *bazická intenzita úmrtnosti (baseline hazard function).* Tato intenzita nezávisí na čase t a často se normuje tak, abychom nemuseli používat intercept v $\boldsymbol{x}_{t\cdot}\boldsymbol{\beta}$.

Je také zajímavé, že pokud vezmeme poměr intenzit úmrtnosti pro dvě pozorování, pak je tento poměr nezávislý na bazické intenzitě úmrtnosti

$$\frac{\lambda_{t_1}(\tau)}{\lambda_{t_2}(\tau)} = \frac{\exp(\boldsymbol{x}_{t_1\cdot}\boldsymbol{\beta})}{\exp(\boldsymbol{x}_{t_2\cdot}\boldsymbol{\beta})}.$$

Funkce přežití Coxova modelu má tvar

$$S_t(\tau) = S_0(\tau)^{\exp(\boldsymbol{x}_{t\cdot}\boldsymbol{\beta})}.$$

Pokud předchozí model rozšíříme tak, že i regresory budou závislé na čase dostaneme obecnější Coxův model

$$\lambda_t(\tau) = \lambda_0(\tau)\exp(\boldsymbol{x}_{t\cdot}(\tau)\boldsymbol{\beta}).$$

Bazická intenzita úmrtnosti a parametry $\boldsymbol{\beta}$ se odhadnou metodou maximální věrohodnosti. Při odhadu se využívá právě vlastnosti, že podíl dvou intenzit úmrtnosti je nezávislý na bazické intenzitě úmrtnosti. Tato metoda je, narozdíl od předchozích parametrických metod, semiparametrická. Bazická intenzita úmrtnosti se totiž odhaduje neparametricky.

Srovnejme nyní Weibullův a Coxův model. Vyjmenujeme rozdíly v použití:

1. Coxův model lze použít ve větším množství případů.
2. Jestliže můžeme aplikovat Weibullův model, pak lze aplikovat i Coxův.
3. Pokud lze aplikovat oba modely, potom Coxův je méně vhodný, neboť mu odpovídá menší síla testů.

Empirické potvrzení těchto informací lze nalézt v [14].

Kromě výše zmíněných metod se používají i metody neparametrické. K těmto metodám patří především Kaplan-Meierův odhad [9], nebo jeho zobecnění Nelson-Aalenův odhad [1].

Příklad 2.1 *V EViews nejsou metody pro odhad doby trvání vůbec implementovány. Lze zde ale pracovat s cenzorovanými veličinami. Bohužel pouze pokud předpokládáme normální, logistické, či extremální rozdělení typu I.*

Takže bude rozumnější zvolit jiný software. Pro tuto analýzu si vybereme volně dostupný program R. Předem uvedeme, že nápověda ke knihovně obsahující procedury k analýze přežití je v R poněkud matoucí. Není zde naprosto přesně popsáno, co se vlastně odhaduje. K tomu, že se odhady, které budou ukázány, shodují s tím, co je uvedeno v této kapitole, nás mohou vést pouze následující okolnosti: diskuse na internetu psané lidmi, kteří balíček užívají a simulace.

Nejprve popíšeme data. Na tomto místě ještě jednou poděkujme firmě Penco, která data ochotně zapůjčila.

Data se týkají potravinových výrobků, které slouží především pro sportovce. Jde o doplňky, které mají různé charakteristiky, jako zvýšení výkonu, hubnutí, či přísun vitamínů a minerálů. Charakteristiky výrobků budou stručně popsány v části, kde popisujeme veličiny.

My pro naši analýzu budeme předpokládat, že každý výrobek v určité chvíli přestane být prodejný. V tomto okamžiku bude stažen z výroby. Budeme při daných charakteristikách zkoumat, za jakou dobu k tomuto okamžiku dojde.

Zmiňme také, že data byla z původní podoby (poskytnuté firmou Penco) upravena v programu Gawk, neboť nebyla ve vhodné podobě pro pozdější analýzu. V následující analýze budeme pracovat již pouze s upravenými daty. Lze je najít v A.2.

Celkem máme 52 pozorování. Pozorování je cenzorováno, pokud se výrobek nepřestal prodávat. Cenzorovaných pozorování je 35.

Název	*Popis*	*Hodnoty*
hmot	*Hmotnost výrobku v gramech.*	$\mathbb{Z}$
prasek	*Má-li výrobek formu prášku (1=prášek).*	*0, 1*
tobol	*Je-li výrobek balen v tobolkách (1=tobolka).*	*0, 1*
tyc	*Jde-li o tyčinku (1=ano).*	*0, 1*
gel	*Má-li výrobek formu gelu (1=gel).*	*0, 1*
tekut	*Má-li výrobek tekutou formu (1=ano).*	*0, 1*
nakl	*Náklady na výrobu v korunách.*	$\mathbb{R}^+$
cena	*Prodejní cena v korunách.*	$\mathbb{R}^+$
cukr	*Je-li ve výrobku přítomna glukóza (1=ano).*	*0, 1*
umel	*Jsou-li ve výrobku umělá sladidla (1=ano).*	*0, 1*
vitam	*Výrobek slouží jako zdroj vitamínů (1=ano).*	*0, 1*

Název	*Popis*	*Hodnoty*
miner	*Výrobek slouží jako zdroj minerálů (1=ano).*	*0, 1*
energ	*Zdroj energie (1=ano).*	*0, 1*
vykon	*Zvýšení výkonu (1=ano).*	*0, 1*
snizhm	*Výrobek pomáhá při snižování hmotnosti (1=ano).*	*0, 1*
kloub	*Kloubní přípravek (1=ano).*	*0, 1*
fci	*Kolik funkcí má přípravek.*	*1, 2, 3*
karton	*Zda je výrobek prodáván v kartonové válcové krabici (1=ano).*	*0, 1*
plast	*Je-li výrobek balen v plastu (1=ano).*	*0, 1*
folie	*Výrobek má obal z folie (1=ano).*	*0, 1*
let	*Kolik let se výrobek prodává, či prodával.*	$\mathbb{Z}$
cens	*Zda se výrobek stále prodává (1=ano).*	*0, 1*

Zde jsme uvedli všechny veličiny, ale my nebudeme používat veličiny tekut, fci *a* folie. *Tyto veličiny je možné získat z ostatních lineárními kombinacemi.*

Ještě jednou upřesněme cíl naší analýzy. Chceme odhadnout počet let, která se bude výrobek prodávat. Používáme všechna pozorování, tedy i ta, kdy se výrobek stále prodává. Ovšem pokud se stále prodává, považujeme hodnotu počtu let *v prodeji za cenzorovanou.*

Abychom mohli provést analýzu v R, museli jsme nainstalovat některé knihovny: `survival`, `Hmisc` *a* `Design`. *Je také nutné mít nainstalovanou knihovnu* `splines`.

Knihovny zavoláme příkazem např. `librari(survival)`. *Samozřejmě načteme všechny výše zmíněné knihovny.*

Data zavoláme příkazem `data=read.table("penco_prepis.txt", header = TRUE)`.

Abychom mohli všechny veličiny obsažené v datech volat přímo, použijeme příkaz `attach(data)`.

Ještě je třeba změnit veličinu cens, *neboť R přiřazuje 0, pokud je cenzorováno* `cens=abs(cens-1)`.

Nyní se podíváme na výsledky modelů, které popisujeme v teoretické části a budeme je zkoumat ve stejném pořadí. Do modelů zahrneme pouze veličiny, kdy je p-hodnota menší než 5 %.

Začněme tedy exponenciálním modelem. Pro výstup, který bude následovat použijeme příkaz `summary(survreg( Surv(let, cens) ~ tobol +`

```
gel + nakl + cena + cukr + vitam + energ + kloub, dist=
"exponential")).
```

Použijeme výstup, který se shoduje s výstupem v R.

```
              Value Std. Error      z        p
(Intercept)  2.32456    0.52459  4.431 9.37e-06
tobol        0.94926    1.20091  0.790 4.29e-01
gel         10.32056    0.00000    Inf 0.00e+00
nakl         0.01303    0.01184  1.101 2.71e-01
cena        -0.00345    0.00413 -0.836 4.03e-01
cukr         1.67497    0.87393  1.917 5.53e-02
vitam       -1.55552    0.86748 -1.793 7.29e-02
energ        0.56776    0.80816  0.703 4.82e-01
kloub        7.35734    0.00000    Inf 0.00e+00

Scale fixed at 1

Exponential distribution
Loglik(model)= -66.4   Loglik(intercept only)= -73.1
        Chisq= 13.45 on 8 degrees of freedom, p= 0.097
Number of Newton-Raphson Iterations: 11
n= 52
```

Je vidět, že model jako celek zamítáme na hladině 5 %. Nicméně, při použití exponenciálního rozdělení, lze do modelu zahrnout méně veličin než činíme, a potom získáme p-hodnotu menší.

Nyní se podívejme na Weibullův model, který dává dobré výsledky. Podezřelé ovšem je, že téměř všechny veličiny je vhodné použít.

Voláme jej příkazem `summary(survreg( Surv(let, cens)~ prasek + tobol + tyc + gel + nakl + cena + cukr + umel + vitam + miner + energ + vykon + snizhm + kloub , dist="weibull")`

```
              Value Std. Error     z        p
(Intercept)  1.62493    0.56435  2.88 3.99e-03
prasek       0.99427    0.24727  4.02 5.80e-05
tobol        2.32739    0.41496  5.61 2.04e-08
tyc          0.60118    0.37171  1.62 1.06e-01
gel          4.54943    0.00000   Inf 0.00e+00
nakl         0.00619    0.00394  1.57 1.16e-01
```

```
cena         -0.00184    0.00137 -1.34 1.80e-01
cukr          1.38173    0.19608  7.05 1.83e-12
umel          1.49531    0.27860  5.37 8.00e-08
vitam        -1.43975    0.43334 -3.32 8.92e-04
miner        -1.25068    0.38202 -3.27 1.06e-03
energ         1.38081    0.38873  3.55 3.82e-04
vykon        -1.38375    0.58899 -2.35 1.88e-02
snizhm       -1.40835    0.63390 -2.22 2.63e-02
kloub         1.42213    0.00000   Inf 0.00e+00
Log(scale)   -1.60144    0.21870 -7.32 2.44e-13

Scale= 0.202

Weibull distribution
Loglik(model)= -48.1   Loglik(intercept only)= -67.2
        Chisq= 38.2 on 14 degrees of freedom, p= 0.00048
Number of Newton-Raphson Iterations: 16
n= 52
```

Označíme-li s hodnotu `Scale` *z tabulky výše, potom* $\alpha = \frac{1}{s}$*, kde* α *je parametr Weibullova rozdělení, viz výše.*

Z modelu je patrné, že by bylo nevhodné vypustit parametr Scale, tedy α*. Z toho plyne, že užití exponenciálního modelu by bylo v tomto případě nevhodné.*

Připomeňme vztah (2.5)*. Koeficienty* $\boldsymbol{\beta}$ *zde jsou právě v tomto smyslu. Tedy* $\boldsymbol{x}_t.\boldsymbol{\beta}$ *odhaduje* $\ln(y_t)$*.*

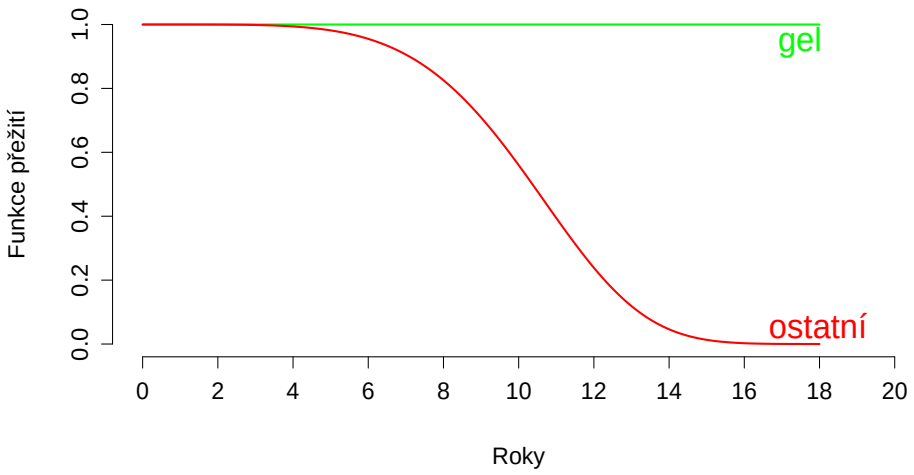
Pokud vezmeme veličinu gel *nebo* kloub*, pak ty pokud nabudou hodnoty 1, je dané pozorování vždy cenzorované. Toto způsobilo, že na výstupu je p-hodnota rovna nule.*

Pro veličinu gel *necháme vykreslit funkci přežití. Resp. pokud veličina* gel *nabývá hodnoty 1, znázorníme funkci přežití zelenou barvou. Pokud nula použijeme červenou. Ostatní veličiny budou nabývat svých průměrů.*

Tento obrázek získáme posloupností následujících příkazů.

```
gl=factor(gel,levels=c(1,0),labels=c("gel","ostatní"))
dd <- datadist(let, gl)
options(datadist='dd')
fit=psm(Surv(let, cens)$\sim$prasek + tobol + tyc + gl + nakl +
cena + cukr + umel + vitam + miner + energ + vykon + snizhm +
```

```
kloub, dist="weibull")
postscript('weibull_prez_gel.eps',width=10,height=6,onefile=
TRUE, paper='special',encoding="CP1250",horizontal=FALSE)
par(cex=1.4)
survplot(fit, gl=NA, prasek=mean(prasek), obol=mean(tobol),
tyc=mean(tyc), nakl=mean(nakl), cena=mean(cena),
cukr=mean(cukr), umel=mean(umel), vitam=mean(vitam),
miner=mean(miner), energ=mean(energ), vykon=mean(vykon),
snizhm=mean(snizhm), kloub=mean(kloub), xlab="Roky",
lab="Funkce přežití", col=c("green","red"),
lty=c(1,1),lwd=c(2,2))
dev.off()
```



Obrázek 2.1: Funkce přežití pro gel=1 a pro gel=0 (Weibullův model).

Pokud bychom chtěli nechat vykreslit intenzitu úmrtnosti, museli bychom do `survplot(...)` *přidat na konec* `what="hazard"`.

U logaritmicko-normálního modelu nebudeme uvádět výstup, protože není příliš zajímavý. Postupovali jsme jako v předchozích případech. Ve srovnání s Weibullovým modelem byl tento horší. Pro zajímavost uveďme veličiny,

které se v tomto modelu vyskytly: prasek, tobol, tyc, gel, nakl, cena, cukr, umel, vitam, miner, energ, vykon, snizhm *a* kloub.

K tomu, abychom použili logaritmicko-normální model, stačí v proceduře `survreg(...)` *napsat* `,dist="lognormal"`. *Samozřejmě je třeba změnit veličiny.*

Poslední model, který budeme zkoumat, je model Coxův. Zavoláme jej příkazem `summary(coxph(Surv(let, cens)~prasek + tobol))`.

Problém spočívá v tom, že musíme vyřadit veličiny, které nenabývají jedničky v necenzorované podobě. Tzn. veličiny gel *a* kloub. *Možná i proto je tento model daleko chudší než třeba Weibullův.*

Uveďme část tohoto výstupu.

```
        coef exp(coef) se(coef)     z      p
prasek -2.08    0.1247    0.782 -2.66 0.0078
tobol  -3.18    0.0414    1.222 -2.61 0.0092


Likelihood ratio test= 10.5  on 2 df,   p=0.00525
Wald test            = 9.17  on 2 df,   p=0.0102
Score (logrank) test = 11.5  on 2 df,   p=0.00320
```

Veličiny v tomto modelu jsou podmnožinou těch z Weibullova. Oba modely ale byly konstruovány ze všech veličin.

Uveďme neparametrický odhad pro bazickou intenzitu úmrtnosti. Pokud tento odhad voláme, pak místo `summary` *v předchozím příkazu zadáme* `basehaz`. *Nakonec musíme přidat ještě příkaz* `centered=FALSE`.

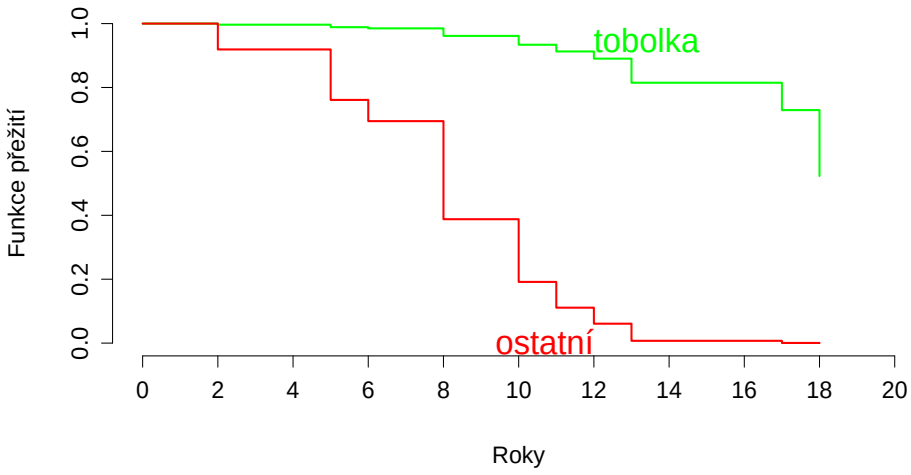
Uveďme výstup:

```
1   0.0844254    2
2   0.2730002    5
3   0.3643189    6
4   0.9473886    8
5   1.6527603   10
6   2.1996351   11
7   2.8011714   12
8   4.9354745   13
9   7.6081187   17
10 15.6260515   18
```

Nakonec si ukažme ještě funkci přežití v Coxově modelu, pokud veličina prasek *nabývá svého průměru a veličina* tobol *hodnot jedna a nula.*

Ještě uvedeme posloupnost příkazů, kterými tento obrázek v R vytvoříme.

```
tbl=factor(tobol,levels=c(1,0),labels=c("tobolka","ostatní"))
dd <- datadist(let, tbl)
options(datadist='dd')
fit=cph(Surv(let, cens)$\sim$prasek + tbl,surv=TRUE)
postscript('cox_prez_tob.eps',width=10,height=6,onefile=TRUE,
paper='special',encoding="CP1250",horizontal=FALSE)
par(cex=1.4)
survplot(fit,tbl=NA,prasek=0,xlab="Roky",ylab="Funkce přežití",
col=c("green","red"),lty=c(1,1),lwd=c(2,2),adj.subtitle=FALSE)
dev.off()
```



Obrázek 2.2: Funkce přežití pro tobol=1 a pro tobol=0 (Coxův model).

△

Dodatek

A.1 Programy v EViews

V této části předvedeme několik procedur napsaných v EViews. Budeme popisovat kód a zabývat se metodami.

A.1.1 „Jackknife“

V následujícím odstavci se zaměříme na metodou „jackknife.“ Při popisu kódu bude stručně nastíněno, o jakou techniku jde.

Nyní bude následovat text zabývající se metodou „jackknife“ pro binární proměnnou.

Určíme mez, podle které budeme jednotlivým pozorováním přiřazovat hodnotu 1, či 0. Tj. model nám odhadne pravděpodobnost, že pozorování nabyde hodnoty 1, my pak musíme určit, při jaké hodnotě stanovíme, že tento odhad znamená kategorii 1 nebo 0 (je přirozené volit tuto mez rovnu 0,5).

Dále sdělíme programu, jaké chceme do modelu zahrnout závisle a nezávisle proměnné (2-4 řádek). Upozorněme na to, že mezi třetím a čtvrtým řádkem nemusíme v proceduře odřádkovat, takto byl text zarovnán
v EViews.

Dále jsou definovány pomocné proměnné (řádky 5-14).

Pouze na 8. řádce přetvoříme závisle proměnnou tak, aby byla nejen binární, ale opravdu nula-jedničková.

Nejpodstatnější část programu tvoří řádky 15-26, kde bereme množinu pozorování a jedno z ní vyloučíme. Na základě takto získané množiny postavíme model (řádek 18). Pro vyloučené pozorování uděláme za pomoci tohoto modelu predikci (19) a zkontrolujeme, jestli tato predikce odpovídá skutečnosti (21-24). Pozorování vrátíme do vzorku a postup opakujeme s následujícím.

Uděláme také nejjednodušší možnou předpověď, a to způsobem, že všechna pozorování odhadneme tou hodnotou, která se v celkovém vzorku vyskytuje častěji (29-33). To většinou odpovídá tomu, kdybychom použili metodu „jackknife“ a nezávisle proměnná by byla pouze konstanta. Postup není závislý na tom, jestli bereme v potaz model probit, logit či gompit. Stačí si uvědomit, že pokud bychom měli např. k_1 a k_2 pozorování v první (jedničkové), resp. druhé skupině a bez újmy na obecnosti $k_1 > k_2 + 1$ (tj. množiny se liší nejméně o 2 prvky), pak při vyloučení jednoho prvku stále zbude více prvků v první skupině. Tedy do ní stále zařazujeme všechna pozorování (konstanta nám umožňuje vybrat jen jednu skupinu). Potíž nastává, pokud $k_1 = k_2 + 1$, pak k_2-krát zařadíme pozorování do první skupiny (tj. ovšem špatně) a k_1-krát nemůžeme o zařazení rozumně rozhodnout. Případ $k_1 = k_2$ by vyšel metodou „jackknife“ tak, že bychom neodhadli ani jeden prvek správně. S takovými odhady by ale nebylo příliš rozumné srovnávat, takže prostě zvolíme jednu ze skupin (resp. početnější skupinu) pro všechny prvky.

Spočítáme, o kolik procent vylepší námi používaný model tento jednoduchý (34-38).

Nakonec smažeme nepotřebné veličiny (39-40).

Pokud chceme změnit model na logit, resp. gompit, přepíšeme na řádce 18 d=m za d=l, resp. d=x.

```
1   scalar mez=0.5
2   series zavisla=hlas
3   group nezavisla  C BUDOSTBOD JIST _
4   MAJOSTBOD MENIT ODHOBT SPOKOJ VYSL
5   equation jeden_ven
6   scalar spravnych=0
7   scalar pocet= @obssmpl
8   zavisla=(zavisla-@min(zavisla))/@max(zavisla)
9   vector zavisla_vyj
10  stom(zavisla,zavisla_vyj)
11  vector zavisla_ne
12  stom(zavisla,zavisla_ne)
13  series odhad
14  vector odhad_v
15  for !i=1 to pocet
16     zavisla_vyj(!i)=NA
17     mtos(zavisla_vyj,zavisla)
```

```
18      jeden_ven.binary(d=n) zavisla nezavisla
19      jeden_ven.forecast odhad
20      stom(odhad,odhad_v)
21      if (odhad_v(!i)<=mez and zavisla_ne(!i)=0) or
22         (odhad_v(!i)>mez and zavisla_ne(!i)=1) then
23         spravnych=spravnych+1
24      endif
25      zavisla_vyj(!i)=zavisla_ne(!i)
26   next
27   scalar jednicek=@sum(zavisla_ne)
28   scalar nul=pocet-jednicek
29   if (jednicek>nul) then
30      scalar spravnych_jed=jednicek
31   else
32      scalar spravnych_jed=nul
33   endif
34   if(pocet-spravnych_jed>0) then
35      scalar zlepseni=
(spravnych-spravnych_jed)/(pocet-spravnych_jed)
36   else
37      scalar zlepseni=0
38   endif
39   delete   nezavisla odhad_v mez zavisla_vyj zavisla_ne
40   delete   zavisla odhad jeden_ven jednicek nul pocet
```

Podívejme se nyní na metodu „jackknife“ pro ordinální vysvětlovanou proměnnou.

Opět nejprve definujeme závisle a nezávisle proměnnou (1,2).

Další část není naprosto nezbytná a dokonce trvá poměrně dlouho (jsou to řádky 6 až 51).

Nejprve zde počítáme, kolik hodnot má závisle proměnná a dosazujeme je do vektoru (7-25).

Na řádcích 26-38 seřadíme tyto hodnoty podle velikosti. Je pravda, že námi použitý postup má kvadratickou složitost. Lze použít postup se složitostí $O(T \ln(T)))$. Nicméně tento zdánlivě rychlejší postup by prodloužil kód a dokonce by výpočet trval déle, neboť v EViews jde také o to, kolik řádek musí program vyhodnotit. Další argument spočívá v tom, že pokud T je počet pozorování a $R + 1$ počet tříd, pak skutečná složitost našeho programu je $O(T \cdot (R + 1))$. Přitom R by nemělo být příliš velké.

Dále spočteme, kolik pozorování má danou hodnotu a změníme vektor závisle proměnné tak, aby nabýval pouze hodnot $1, \ldots, R + 1$, a to podle toho, jaké pořadí to které pozorování mělo (39-51).

Některé z těchto kroků by bylo možné vynechat a hodnoty dosadit zvenku. Nicméně takto napsaná procedura se snaží minimalizovat množství dosazování zvenku a tím snížit možnost chyby při eventuálním použití. Bohužel, ačkoli EViews obsahuje tabulky o počtu hodnot ordinální proměnné a dokonce pro každou tuto hodnotu počet pozorování, která ji nabývají, nelze najít jakým příkazem by se tato čísla dala volat. Toto je ospravedlnitelné tím, že EViews není příliš určen k tomu, aby se v něm tyto procedury psaly, ale spíše k tomu, aby ukazoval výstupy, na které se lze dostat pouhým „kliknutím“.

Ústřední část celé této procedury tvoří řádky 52-83, kde na začátku definujeme pomocné proměnné.

Ovšem pozor, pro ordinální veličiny nelze odhadnout závisle proměnnou stejně jako v případě binární veličiny. Postupujme krok za krokem.

Řádky 61 až 63 vytvoří model, který sice není potřeba, ale díky němu a odhadu pravděpodobností vznikne správný počet posloupností pro odhad pravděpodobností. Resp. kdyby náhodou pozorování číslo jedna nabývalo hodnoty, která se v celém vzorku vyskytne pouze jednou, matici pravděpodobností by chyběl jeden sloupec. Tento způsob je sice poněkud kostrbatý, ale často bývá výhodné snažit se, aby program v EViews obsahoval co možná nejméně příkazů. Níže si popíšeme, jak se model pro ordinální vysvětlovanou proměnnou odhaduje.

65 Vyloučíme pozorování, která budeme odhadovat z modelu sestaveného z ostatních.

66 S řadami nelze provádět některé operace, takže je převedeme na vektor.

67 Odhadneme koeficienty modelu s vyloučeným pozorováním.

68 Na základě rovnice, která definuje náš model, vytvoříme novou strukturu v EViews, která se zde jmenuje také Model (pro odlišení s teoretickým modelem, budeme Model v EViews psát s velkým počátečním písmenem).

69 Spustíme proceduru Model, která nám odhadne pravděpodobnosti toho, do které skupiny pozorování patří.

70 Takto vezmeme všechny objekty, kde se místo otazníku vyskytuje téměř libovolný znak.

72-74 Pokud jsme způsobili vyjmutím testovací množiny vyloučení jedné z hodnot, pak tuto hodnotu nemůžeme odhadnout. Proto pravděpodobnosti toho, že máme odhadnout vyloučenou hodnotu, položíme rovné nule.

75-82 Zjistíme, podle dříve spočítaných pravděpodobností, které číslo má nejvyšší pravděpodobnost výskytu, a tímto číslem odhadneme.

83-87 Dále zjistíme, jestli tento odhad odpovídá skutečnosti (eventuálně spočteme absolutní odchylku od skutečné hodnoty).

89 Vrátíme vyjmuté pozorování na předešlé místo.

Další výpočty (90-97) se týkají jednoduché strategie, kdy vybereme nejčastější hodnotu. Diskusi o tom, že to není vždy to samé jako použít model jen s konstantou, viz před předchozí procedurou.

Absolutní odchylku pro jednoduchou strategii (98-101) spočteme tak, že počet pozorování v dané skupině vynásobíme absolutní hodnotou rozdílu upravené funkční hodnoty dané skupiny a nejčastější hodnotou.

Ještě spočteme, o kolik náš model vylepší jednoduchou strategii (102-106).

Smažeme proměnné, které nebudou na výstupu potřeba.

```
1   series zavisla=spokoj
2   group nezavisla BUDOSTBOD BUDDOBR HLAS PLAT VYSL
3   scalar pocet= @obssmpl
4   vector zavisla_ne
5   stom(zavisla,zavisla_ne)
6   scalar hodnot=1
7   vector(pocet) hodnoty
8   hodnoty(1)=zavisla_ne(1)
9   scalar i
10  scalar j
11  scalar stejne
12  for i=2 to pocet
13     stejne=0
14     j=1
```

```
15     while((stejne=0) and (j<=hodnot))
16        if(zavisla_ne(i)=hodnoty(j)) then
17           stejne=1
18        endif
19        j=j+1
20     wend
21     if(stejne=0) then
22        hodnot=hodnot+1
23        hodnoty(hodnot)=zavisla_ne(i)
24     endif
25  next
26  scalar nahrazovany
27  scalar nejmensi_index
28  for i=1 to hodnot-1
29     nejmensi_index=i
30     for j=i+1 to hodnot
31        if (hodnoty(j)<hodnoty(nejmensi_index)) then
32           nejmensi_index=j
33        endif
34     next
35     nahrazovany=hodnoty(i)
36     hodnoty(i)=hodnoty(nejmensi_index)
37     hodnoty(nejmensi_index)=nahrazovany
38  next
39  vector(hodnot) pocty_hodnot=0
40  for i=1 to pocet
41     !stop=0
42     j=1
43     while(!stop=0)
44        if(zavisla_ne(i)=hodnoty(j)) then
45           !stop=1
46           pocty_hodnot(j)=pocty_hodnot(j)+1
47           zavisla_ne(i)=j
48        endif
49        j=j+1
50     wend
51  next
52  mtos(zavisla_ne,zavisla)
```

```
53 equation jeden_ven
54 vector zavisla_vyj
55 stom(zavisla,zavisla_vyj)
56 vector(pocet) odhad_v
57 scalar spravnych=0
58 scalar odchylka=0
59 scalar zatimnejprst
60 matrix prsti_m=0
61 jeden_ven.ordered(d=x) zavisla nezavisla
62 jeden_ven.makemodel(predpov_model)
63 predpov_model.solve
64 for i=1 to pocet
65    zavisla_vyj(i)=NA
66    mtos(zavisla_vyj,zavisla)
67    jeden_ven.ordered(d=x) zavisla nezavisla
68    jeden_ven.makemodel(predpov_model)
69    predpov_model.solve
70    group prsti zavisla_?_0
71    stom(prsti,prsti_m)
72    if(pocty_hodnot(zavisla_ne(i))=1) then
73       prsti_m(i,zavisla_ne(i))=0
74    endif
75    odhad_v(i)=1
76    zatimnejprst=prsti_m(i,1)
77    for j=2 to hodnot
78       if (zatimnejprst<prsti_m(i,j)) then
79          zatimnejprst=prsti_m(i,j)
80          odhad_v(i)=j
81       endif
82    next
83    if (odhad_v(i)=zavisla_ne(i)) then
84       spravnych=spravnych+1
85    else
86       odchylka=odchylka + @abs(odhad_v(i)-zavisla_ne(i))
87    endif
88    zavisla_vyj(i)=zavisla_ne(i)
89 next
90 scalar spravnych_jed=pocty_hodnot(1)
```

```
91  scalar nej_skup=1
92  for i=2 to hodnot
93     if(spravnych_jed<pocty_hodnot(i)) then
94        spravnych_jed=pocty_hodnot(i)
95        nej_skup=i
96     endif
97  next
98  scalar odchylka_jed=0
99  for i=1 to hodnot
100    odchylka_jed=
odchylka_jed+ @abs(i-nej_skup)* pocty_hodnot(i)
101 next
102 if(pocet-spravnych_jed>0) then
103    scalar zlepseni=
(spravnych-spravnych_jed)/(pocet-spravnych_jed)
104 else
105    scalar zlepseni=0
106 endif
107 delete *zavisla* zatimnejprst stejne predpov_model _
108 pocet odhad_v  nejmensi_index nahrazovany prsti* _
109 jeden_ven i j hodnoty hodnot nej_skup pocty_hodnot
```

A.1.2 Prostý náhodný výběr

Nejprve definujeme, kolik procent ze vzorku se má použít jako testovací množina (1). Počet členů této množiny označíme jako k. Počet všech pozorování T. V testovací množině by nemělo být více jak 30 % pozorování, protože potom by bylo velmi těžké získat výběr metodou, kterou používáme (prostý náhodný výběr).

Definujeme mez, závisle a nezávisle proměnnou (2-5).

Na řádcích 6 až 30 probíhá výběr vzorku, který použijeme jako testovací množinu. Náhodně vygenerujeme k čísel od 1 do T (12-14). Pak zkontrolujeme, jestli v takto vybraném vzorku jsou všechna čísla různá (15-29). Pokud ano, končíme a jako testovací množinu použijeme ta pozorování, která jsou v pořadí na místě, které se vyskytuje v náhodně vygenerovaných datech. Pokud se některá čísla ve vybraném vzorku shodují, nepoužijeme jej a vygenerujeme nový. Takto budeme postupovat do té doby, dokud nebudou všechna čísla různá.

Na řádcích 31-35 vynecháme u závisle proměnné ta pozorování, která jsou v testovací množině. Takto upravená data pak použijeme pro stavbu modelu. EViews je díky tomu do modelu nezahrne.

V další části 36-39 odhadneme Model s upravenými daty. Také v Modelu předpovíme hodnoty na testovací množině.

Spočítáme, jak se naše předpovědi shodují se skutečností (41-53).

Na řádcích 54-63 podobně jako v předchozím programu určíme nejjednodušší možný odhad. Tj. ze všech pozorování (kromě testovací množiny) určíme, které číslo převládá. Tímto číslem odhadneme závisle proměnnou v testovací množině.

Řádky 66-70 počítají zlepšení našeho Modelu ve srovnání s Modelem pouze s konstantou. Pokud náš Model neznamená zlepšení, tento odhad funguje špatně.

Zbývající řádky slouží ke smazání proměnných, které už nejsou podstatné.

```
1   scalar vzorek_procent=10
2   scalar mez=0.5
3   series zavisla=hlas
4   group nezavisla  C BUDOSTBOD JIST  MAJOSTBOD _
5   MENIT ODHOBT SPOKOJ VYSL
6   scalar pocet=@obssmpl
7   zavisla=(zavisla-@min(zavisla))/@max(zavisla)
8   scalar kolik=@round(pocet*(vzorek_procent/100))
9   vector(kolik) vyjmi
10  vector(kolik) jednicky=1
11  !ruzne=0
12  while !ruzne=0
13     rndint(vyjmi,pocet-1)
14     vyjmi=vyjmi+jednicky
15     !stejne=0
16     !i=1
17     while !stejne=0 and !i<=kolik
18        !j=!i+1
19        while !stejne=0 and !j<=kolik
20           if vyjmi(!i)=vyjmi(!j) then
21              !stejne=1
22           endif
23           !j=!j+1
```

```
24          wend
25          !i=!i+1
26       wend
27       if !stejne=0 then
28          !ruzne=1
29       endif
30    wend
31    vector(pocet) zavisla_vec
32    stom(zavisla,zavisla_vec)
33    for !i=1 to kolik
34       zavisla_vec(vyjmi(!i))=NA
35    next
36    series zavislam
37    mtos(zavisla_vec,zavislam)
38    equation jeden_ven.binary(d=n) zavislam nezavisla
39    jeden_ven.forecast odhad
40    scalar shoda=0
41    vector(pocet) odhad_vec
42    stom(odhad,odhad_vec)
43    stom(zavisla,zavisla_vec)
44    for !i=1to kolik
45       if odhad_vec(vyjmi(!i))>mez then
46          !odh_zavisla=1
47       else
48          !odh_zavisla=0
49       endif
50       if zavisla_vec(vyjmi(!i))= !odh_zavisla then
51          shoda=shoda+1
52       endif
53    next
54    scalar jednicek=@sum(zavisla_vec)
55    scalar nul=pocet-jednicek
56    stom(zavisla,zavisla_vec)
57    scalar jednicek_vzorek=0
58    for !i=1 to kolik
59       if (zavisla_vec(vyjmi(!i))=1) then
60          jednicek_vzorek=jednicek_vzorek+1
61       endif
```

```
62  next
63  if (jednicek>nul) then
64      scalar shoda_jed=jednicek_vzorek
65  else
66     scalar shoda_jed=kolik-jednicek_vzorek
67  endif
68  if(shoda-shoda_jed>0) then
69     scalar zlepseni=(shoda-shoda_jed)/(kolik-shoda_jed)
70  else
71     scalar zlepseni=0
72  endif
73  delete vzorek_procent zavisla nezavisla jeden_ven  pocet
74  delete mez  kolik vyjmi jednicky zavisla_vec zavislam
75  delete  odhad_vec jednicek nul jednicek_vzorek odhad
```

Pokud bychom chtěli prostým náhodným výběrem vybrat testovací množinu pro uspořádanou závisle proměnnou, stačilo by zkombinovat předchozí programy.

Budeme používat následující značení I.5 znamená: z druhého programu v odstavci A.1.1 vlož do nyní popisovaného programu řádek 5. II.5 znamená: vlož z předchozího programu 5. řádek do nyní popisovaného programu.

Nejprve provedeme definice II.1, II.3-6. Samozřejmě je třeba změnit veličiny u závisle a nezávisle proměnné.

Dál uděláme prostý náhodný výběr a uložíme jej do vektoru II.8-30.

Upravíme veličinu do vhodného tvaru viz A.1.1, tj. I.4-51.

Zbytek programu raději zase přímo vypíšeme, abychom čtenáři ušetřili práci.

V této nové části nejprve vyjmeme ze souboru ta pozorování, která jsou v testovací množině (1-5).

Vytvoříme Model se všemi pozorováními v souboru, který slouží k tomu, abychom měli veličinu zavisla_?_0 pro každou hodnotu a díky tomu mohli správně upravit matici pravděpodobností.

Na řádcích 9 až 12 odhadneme Model s daty, která již neobsahují testovací množinu. A tento odhad nám také určí odhady pravděpodobností, že pozorování patří do té které skupiny.

Určíme kolik pozorování se vyskytuje v té které skupině pro testovací množinu (16-27).

To také určíme pro množinu, se kterou stavíme Model (28).

Vytvoříme matici pravděpodobností, že daná pozorování budou patřit do některé skupiny (14, 15, 30-34). Pokud by se stalo, že některé pozorování bude pouze v testovací množině, potom odhad toho, že pozorování patří do této skupiny, je vždy nula.

V další části provedeme odhady pozorování v testovací množině. Počítáme, kolik pozorování jsme v testovací množině odhadli správně, a také spočteme statistiku (1.10) (35-53).

Následující postup je obdobou postupu z programu pod A.1.1. Vybereme nejpočetnější skupinu v souboru pro model a tou odhadneme všechna pozorování z testovací množiny. S tímto odhadem pak porovnáváme model se všemi nezávisle proměnnými obdobně jako v předchozích programech (63-72).

Vše nepotřebné smažeme.

```
1   vector(pocet) zavisla_vyj
2   zavisla_vyj=zavisla_ne
3   for !i=1 to kolik
4      zavisla_vyj(vyjmi(!i))=NA
5   next
6   equation jeden_ven.ordered(d=n) zavisla nezavisla
7   jeden_ven.makemodel(predpov_model)
8   predpov_model.solve
9   mtos(zavisla_vyj,zavisla)
10  equation jeden_ven.ordered(d=n) zavisla nezavisla
11  jeden_ven.makemodel(predpov_model)
12  predpov_model.solve
13  group prsti zavisla_?_0
14  matrix prsti_m
15  stom(prsti,prsti_m)
16  vector(hodnot) pocty_hodnot_vz=0
17  for i=1 to kolik
18     !stop=0
19     j=1
20     while(!stop=0)
21        if(zavisla_ne(vyjmi(i))=hodnoty(j)) then
22           !stop=1
23           pocty_hodnot_vz(j)=pocty_hodnot_vz(j)+1
24        endif
25        j=j+1
```

```
26     wend
27  next
28  vector(hodnot) pocty_hodnot_bezvz=
pocty_hodnot-pocty_hodnot_vz
29  vector(pocet) nuly=0
30  for i=1 to hodnot
31     if pocty_hodnot_bezvz(i)=0 then
32        colplace(prsti_m,nuly,i)
33     endif
34  next
35  vector(kolik) odhad_v
36  scalar spravnych=0
37  scalar odchylka=0
38  scalar zatimnejprst
39  for i=1to kolik
40     odhad_v(i)=1
41     zatimnejprst=prsti_m(vyjmi(i),1)
42     for j=2 to hodnot
43        if (zatimnejprst<prsti_m(vyjmi(i),j)) then
44           zatimnejprst=prsti_m(vyjmi(i),j)
45           odhad_v(i)=j
46        endif
47     next
48     if (odhad_v(i)=zavisla_ne(vyjmi(i))) then
49        spravnych=spravnych+1
50     else
51        odchylka=
odchylka+@abs(odhad_v(i)-zavisla_ne(vyjmi(i)))
52     endif
53  next
54  scalar spravnych_jed=pocty_hodnot_vz(1)
55  scalar nej_skup=1
56  scalar skup_nejpocet=pocty_hodnot_bezvz(1)
57  for i=2 to hodnot
58     if(skup_nejpocet<pocty_hodnot_bezvz(i)) then
59        skup_nejpocet=pocty_hodnot_bezvz(i)
60        nej_skup=i
61     endif
```

```
62  next
63  spravnych_jed=pocty_hodnot_vz(nej_skup)
64  scalar odchylka_jed=0
65  for i=1 to hodnot
66     odchylka_jed=
odchylka_jed+ @abs(i-nej_skup) * pocty_hodnot_vz(i)
67  next
68  if(kolik-spravnych_jed>0) then
69     scalar zlepseni=
(spravnych-spravnych_jed)/(kolik-spravnych_jed)
70  else
71     scalar zlepseni=0
72  endif
73  delete zatimnejprst stejne predpov_model nuly *zavisla* _
74  pocet nejmensi_index nahrazovany jednicky vyjmi odhad_v _
75  jeden_ven i j hodnoty hodnot nej_skup vzorek_procent _
76  skup_nejpocet pocty_hodnot* prsti* kolik
```

A.1.3 Srovnání modelů probit, logit a gompit

V tomto programu nám půjde o vytvoření matice obsahující koeficienty $\boldsymbol{\beta}$, které spočítáme pro modely probit, logit a gompit. Program také zpracovává derivaci podmíněné střední hodnoty. Bližší informace viz příklad 1.1.

Opět nejprve definujeme, jaké veličiny chceme použít jako závisle a nezávisle proměnné (1-3).

Pro tyto veličiny sestrojíme po řadě modely probit, logit, gompit a lineární model (4-7).

Definujeme pomocné proměnné. Pro jednoduchost převedeme nezávisle proměnné na matici a určíme počet jejích sloupců (8-12).

Na řádcích 13-16 počítáme průměry jednotlivých proměnných a ukládáme je.

Také u všech modelů uložíme regresní koeficienty (17-20).

Na řádcích 21-29 počítáme pro modely probit, logit a gompit výraz $f(-\bar{\boldsymbol{x}}^\top\hat{\boldsymbol{\beta}})\hat{\beta}_i$.

Následně všechny potřebné výsledky uložíme do matice (30-38).

Nakonec smažeme pro další analýzy nepodstatné proměnné.

```
1   series zavisla=hlas
```

```
group nezavisla c BUDOSTBOD JIST  MAJOSTBOD MENILBYCH _
ODHOBT SPOKOJEN VYSLEDEK
equation pro.BINARY(D=N) zavisla nezavisla
equation logi.BINARY(D=L) zavisla nezavisla
equation gom.BINARY(D=X) zavisla nezavisla
equation lin.ls zavisla nezavisla
scalar poz=@obssmpl
matrix nezavisla_m
stom(nezavisla,nezavisla_m)
scalar sloupcu=@columns(nezavisla_m)
vector(sloupcu) prumer
for !i=1 to sloupcu
   vector v= @columnextract(nezavisla_m,!i)
   prumer(!i)=@mean(v)
next
vector lin_co=lin.@coefs
vector probit_co=pro.@coefs
vector logit_co=logi.@coefs
vector gompit_co=gom.@coefs
vector xb_probit=@transpose(prumer)*probit_co
vector xb_logit=@transpose(prumer)*logit_co
vector xb_gompit=@transpose(prumer)*gompit_co
scalar fmprob=@dnorm(-xb_probit(1))
scalar fmlog=@dlogistic(-xb_probit(1))
scalar fmgomp=@dextreme(-xb_probit(1))
vector smer_probit=fmprob*probit_co
vector smer_logit=fmlog*logit_co
vector smer_gompit=fmgomp*gompit_co
matrix(sloupcu,8) smer
colplace(smer, lin_co, 1)
colplace(smer, lin_co, 2)
colplace(smer, probit_co, 3)
colplace(smer, smer_probit, 4)
colplace(smer, logit_co, 5)
colplace(smer, smer_logit, 6)
colplace(smer, gompit_co, 7)
colplace(smer, smer_gompit, 8)
delete zavisla nezavisla pro logi gom lin xb_probit
```

```
40  delete nezavisla_m sloupcu prumer v lin_co probit_co
41  delete smer_probit smer_logit smer_gompit poz
42  delete xb_logit xb_gompit logit_co gompit_co
```

A.1.4 Graf podmíněné pravděpodobnosti

Stejně jako v předchozích případech si nejprve připravíme veličiny, které použijeme: závisle proměnnou (1), nezávisle proměnné, ale bez veličiny, která bude na ose X (2), závisle proměnnou, která v grafu bude na ose X (3) a počet bodů ve kterých bude hodnota podmíněné pravděpodobnosti určena přesně (4).

Následuje vytvoření modelů probit, logit a gompit (5-7).

Řádky 8-11 určují body na ose X, kde se bude vyhodnocovat odhad podmíněné pravděpodobnosti našich modelů.

Na řádcích 12-20 se vytvoří matice, která obsahuje nové vektory $\boldsymbol{x}$, kde kromě jedné vysvětlované proměnné, která bude na ose X, jsou ostatní veličiny určeny svými průměry. Počet vektorů v matici je dán počtem bodů, ve kterých jsme chtěli přesně určit hodnoty podmíněné pravděpodobnosti.

Dále určíme hodnoty osy Y, tj. hodnoty podmíněných distribučních funkcí (21-35). Ovšem na řádku 32 vkládáme do matice, jež obsahuje všechny veličiny nutné pro vytvoření grafu, body na ose X, ve kterých právě tyto hodnoty podmíněných distribučních funkcí počítáme.

Řádek 36 otevře okno s grafem, který si budeme moci interaktivně upravit.

Na řádcích 37-39 smažeme proměnné, které nebudou dále potřebné.

```
1   series zavisla=hlas
2   group nezavislab C BUDOSTBOD JIST  MAJOSTBOD MENIT ODHOBT
SPOKOJ
3   series x=vysl
4   scalar bodu=101
5   equation pro.BINARY(D=N) zavisla nezavislab x
6   equation logi.BINARY(D=L) zavisla nezavislab x
7   equation gom.BINARY(D=X) zavisla nezavislab x
8   vector(bodu) bodyx
9   for !i=1 to bodu
10     bodyx(!i)=@min(x)+(!i-1)*((@max(x)-@min(x))/(bodu-1))
11  next
12  matrix nezavislab_m
```

```
13  stom(nezavislab,nezavislab_m)
14  scalar sloupcu=@columns(nezavislab_m)
15  matrix(bodu,sloupcu+1) bodyvse
16  for !i=1 to sloupcu
17     vector(bodu) v= @mean(@columnextract(nezavislab_m,!i))
18     colplace(bodyvse, v, !i)
19  next
20  colplace(bodyvse, bodyx, sloupcu+1)
21  vector probit_co=pro.@coefs
22  vector logit_co=logi.@coefs
23  vector gompit_co=gom.@coefs
24  vector xb_probit=bodyvse*probit_co
25  vector xb_logit=bodyvse*logit_co
26  vector xb_gompit=bodyvse*gompit_co
27  vector(bodu) jedna=1
28  vector pprob=jedna- @cnorm(-xb_probit)
29  vector plog=jedna- @clogistic(-xb_logit)
30  vector pgomp=jedna- @cextreme(-xb_gompit)
31  matrix(bodu,4) Distribuce
32  colplace(Distribuce, bodyx, 1)
33  colplace(Distribuce, pprob, 2)
34  colplace(Distribuce, plog, 3)
35  colplace(Distribuce, pgomp, 4)
36  Distribuce.xyline
37  delete zavisla nezavislab x bodu pro logi gom bodyx
38  delete bodyvse v probit_co logit_co gompit_co xb_probit
39  delete xb_gompit jedna pprob plog pgomp
40  delete nezavislab_m sloupcu xb_logit
```

Nyní přidejme program pro vykreslení podmíněných pravděpodobností, pokud závisle proměnná je ordinální. Hlavička tohoto programu vypadá takto.

```
1   series zavisla=spokoj
2   group nezavislab BUDOSTBOD BUDDOBR HLAS PLAT
3   series x=vysl
4   scalar hodnota_kreslena=6
5   scalar bodu=101
```

Definujeme zde závisle proměnnou, část nezávisle proměnných a veličinu nezávisle proměnnou, která bude v grafu na ose X.

Další část je z programu pro uspořádané náhodné veličiny A.1.1. Z něho využijeme řádky 3 až 51. Přičemž lze vynechat 39. a 46. řádek.

Další část raději vypíšeme.

1-3 odhadneme modely.

4-8 vytvoříme systém bodů pro osu X. Vzdálenost mezi minimem a maximem veličiny, kterou pokládáme na osu X, rozdělíme na úseky stejné délky, kde vždy okraj bude souřadnice, u níž se bude určovat podmíněná pravděpodobnost.

Ostatní veličiny, s výjimkou veličiny z osy X, odhadneme průměry (9-16).

Na 17. řádku přidáme také souřadnice proměnné, kterou jsme vybrali pro osu X.

Vypočteme $\boldsymbol{X}^\top \hat{\boldsymbol{\beta}}$ (18-26).

Na řádcích 27-54 necháme spočítat hodnoty pravděpodobností, viz (1.8).

Jejich hodnoty vložíme do matice a necháme si vykreslit požadovaný obrázek.

```
1   equation pro.ordered(d=n) zavisla nezavislab x
2   equation logi.ordered(d=l) zavisla nezavislab x
3   equation gom.ordered(d=x) zavisla nezavislab x
4   vector(bodu) bodyx
5   for !i=1 to bodu
6      bodyx(!i)=@min(x)+(!i-1)*((@max(x)-@min(x))/(bodu-1))
7   next
8   matrix nezavislab_m
9   stom(nezavislab,nezavislab_m)
10  scalar sloupcu=@columns(nezavislab_m)
11  matrix(bodu,sloupcu+1) bodyvse
12  vector(bodu) v
13  for !i=1 to sloupcu
14     v= @mean( @columnextract(nezavislab_m,!i) )
15     colplace(bodyvse, v, !i)
16  next
17  colplace(bodyvse, bodyx, sloupcu+1)
18  vector probit_co=pro.@coefs
19  vector logit_co=logi.@coefs
20  vector gompit_co=gom.@coefs
```

```
21  vector(sloupcu+1) probit_co
22  vector(sloupcu+1) logit_co
23  vector(sloupcu+1) gompit_co
24  vector xb_probit=bodyvse*probit_co
25  vector xb_logit=bodyvse*logit_co
26  vector xb_gompit=bodyvse*gompit_co
27  vector(bodu) jedna=1
28  if (hodnota_kreslena=hodnot) then
29     vector(bodu) m_pro_prav=
pro.@coefs(hodnota_kreslena+sloupcu)
30     vector(bodu) m_log_prav=
logi.@coefs(hodnota_kreslena+sloupcu)
31     vector(bodu) m_gomp_prav=
gom.@coefs(hodnota_kreslena+sloupcu)
32     vector pprob=jedna- @cnorm(m_pro_prav -xb_probit)
33     vector plog=jedna- @clogistic(m_log_prav -xb_logit)
34     vector pgomp=jedna- @cextreme(m_gomp_prav -xb_gompit)
35  endif
36  if (hodnota_kreslena=1) then
37     vector(bodu) m_pro_lev=
pro.@coefs(hodnota_kreslena+sloupcu)
38     vector(bodu) m_log_lev=
logi.@coefs(hodnota_kreslena+sloupcu)
39     vector(bodu) m_gomp_lev=
gom.@coefs(hodnota_kreslena+sloupcu)
40     vector pprob= @cnorm(m_pro_lev-xb_probit)
41     vector plog=@clogistic(m_log_lev-xb_logit)
42     vector pgomp=@cextreme(m_gomp_lev-xb_gompit)
43  endif
44  if ((hodnota_kreslena<hodnot)and(hodnota_kreslena>1)) then
45     vector(bodu) m_pro_prav=
pro.@coefs(hodnota_kreslena+sloupcu-1)
46     vector(bodu) m_log_prav=
logi.@coefs(hodnota_kreslena+sloupcu-1)
47     vector(bodu) m_gomp_prav=
gom.@coefs(hodnota_kreslena+sloupcu-1)
48     vector(bodu) m_pro_lev=
pro.@coefs(hodnota_kreslena+sloupcu)
```

```
49     vector(bodu) m_log_lev=
logi.@coefs(hodnota_kreslena+sloupcu)
50     vector(bodu) m_gomp_lev=
gom.@coefs(hodnota_kreslena+sloupcu)
51     vector pprob=@cnorm(m_pro_lev -xb_probit)-
@cnorm(m_pro_prav -xb_probit)
52     vector plog=@clogistic(m_log_lev -xb_logit)-
@clogistic(m_log_prav -xb_logit)
53     vector pgomp=@cextreme(m_gomp_lev -xb_gompit)-
@cextreme(m_gomp_prav -xb_gompit)
54  endif
55  matrix(bodu,4) Distribuce
56  colplace(Distribuce, bodyx, 1)
57  colplace(Distribuce, pprob, 2)
58  colplace(Distribuce, plog, 3)
59  colplace(Distribuce, pgomp, 4)
60  Distribuce.xyline
61  delete zavisla* x bodu pro logi gom bodyx nezavislab_m _
62  bodyvse v probit_co logit_co gompit_co xb_probit xb_logit _
63  xb_gompit jedna pprob plog pgomp m_*_lev m_*_prav hodnot* _
64  nahrazovany nejmensi_index pocet stejne sloupcu i j
```

A.1.5 Graf odhadů podmíněných pravděpodobností pro všechny hodnoty ordinální veličiny

Budeme chtít sestrojit graf, který je uveden v odstavci 1.2. Je to graf podmíněných pravděpodobností, pro každou hodnotu vysvětlované uspořádané proměnné.

Začátek je obdobný jako u předchozího programu. Kromě řádku 4 z hlavičky postupujeme stejně i u kopírování programu z A.1.1. Z něho také vezmeme řádky 3 až 51. Přitom vynecháme 39. a 46. řádek.

Poté definujeme model tímto způsobem: equation model.ordered(d=n) zavisla nezavislab x.

Následně z předchozího programu použijeme řádky 4-17.

Zbývající řádky pro jednoduchost opět raději vypíšeme.

Spočteme nejprve $\boldsymbol{X}^\top \hat{\boldsymbol{\beta}}$ (1-3).

Ve zbytku už odhadujeme podmíněné pravděpodobnosti. Abychom proceduru mohli použít pro modely logit a gompit, bylo by nutné přepsat a

změnit distribuční funkce. To se děje na řádcích 8, 11 a 16. Samozřejmě také musíme změnit definici modelu, viz výše.

```
1   vector model_co=model.@coefs
2   vector(sloupcu+1) model_co
3   vector xb_=bodyvse*model_co
4   matrix(bodu,hodnot+1) Distribuce
5   colplace(Distribuce, bodyx, 1)
6   vector(bodu) jedna=1
7   vector(bodu) m_mod_lev=model.@coefs(1+sloupcu)
8   vector prob= @cnorm(m_mod_lev-xb_)
9   colplace(Distribuce, prob, 2)
10  vector(bodu) m_mod_prav=model.@coefs(sloupcu+hodnot)
11  vector prob=jedna- @cnorm(m_mod_prav -xb_)
12  colplace(Distribuce, prob, hodnot+1)
13  for i=2 to (hodnot-1)
14     vector(bodu) m_mod_prav=model.@coefs(i+sloupcu-1)
15     vector(bodu) m_mod_lev=model.@coefs(i+sloupcu)
16     vector prob=@cnorm(m_mod_lev-xb_)-@cnorm(m_mod_prav-xb_)
17     colplace(Distribuce, prob, i+1)
18  next
19  Distribuce.xyline
20  delete zavisla* x bodu bodyx nezavislab_m i j _
21  bodyvse v model_co xb_ sloupcu _
22  jedna prob m_*_lev m_*_prav hodnot* _
23  nahrazovany nejmensi_index pocet stejne model nezavislab
```

A.2 Použitá data

Tato data jsou převzata z článku: [8]. Uveďme prvních a posledních deset pozorování z dat, která používáme v 1.1 a 1.2. Podrobný popis proměnných lze nalézt v příkladu 1.1.

obt	plat	hlas	jist	zmenroz	dulez	odhobt	budbobr	budbod	budostbod	vysl
0	0	0	7	7	1	4	4	15	13	16
0	0	1	7	7	2	2	6	20	20	14
0	0	0	2	4	4	6	2	10	17	5
0	0	1	4	1	4	3	5	15	14	16
0	0	0	4	2	3	3	3	14	15	15
0	0	1	6	5	4	3	6	18	17	15
0	0	0	6	6	2	4	5	15	13	18
0	0	1	3	2	4	2	6	18	16	17
0	0	0	6	2	1	4	4	15	15	15
0	0	1	6	6	5	6	5	11	9	18
						...				
0	1	1	7	4	5	5	6	17	12	19
0	1	1	5	3	1	5	4	15	15	18
0	1	0	1	1	7	5	4	15	13	14
0	1	1	5	2	4	2	5	15	15	19
0	1	1	4	4	1	4	2	5	12	18
0	1	1	3	5	4	5	4	10	15	13
0	1	0	7	4	1	4	4	15	15	15
0	1	1	4	1	1	5	6	16	14	16
0	1	0	7	4	2	3	6	16	14	18
0	1	1	4	1	4	2	6	18	17	19

spokoj	dobrrozh	menit	testobt	majostbod	vek	pohl	leps	lipnezost
7	1	1	2	13	28	2	1,15385	1
3	7	7	3	18	20	1	1	0
1	7	7	6	15	25	1	0,58824	0
4	7	4	4	16	23	1	1,07143	1
3	2	1	3	16	24	1	0,93333	0
4	6	7	4	14	19	2	1,05882	1
7	1	1	3	18	21	1	1,15385	1
7	7	2	5	15	20	1	1,12500	1
4	6	1	4	15	22	1	1	0
7	7	6	4	16	27	1	1,22222	1
				...				
7	6	4	2	17	30	1	1,41667	1
7	7	7	2	15	25	1	1	0
7	7	7	5	18	27	2	1,15385	1
7	7	6	3	11	29	1	1	0
7	3	3	2	20	28	2	0,41667	0
7	4	5	5	14	21	2	0,66667	0
7	4	4	3	16	27	2	1	0
7	7	6	2	15	22	1	1,14286	1
7	7	4	2	17	20	1	1,14286	1
7	5	7	2	18	21	1	1,05882	1

Následují data poskytnutá firmou Penco. Popis viz odstavec 2.1.

hmot	prasek	tobol	tyc	gel	tekut	nakl	cena	cukr	umel	vitam	miner
900	1	0	0	0	0	108	334,9	1	0	1	1
400	1	0	0	0	0	91,2	320,2	0	1	0	0
500	1	0	0	0	0	71,8	207,6	0	1	0	0
900	1	0	0	0	0	86,8	334,9	1	0	0	1
700	1	0	0	0	0	74,5	263,8	1	0	0	0
1000	1	0	0	0	0	89,4	329,4	1	0	1	0
4500	1	0	0	0	0	540	1403,7	1	0	1	1
2000	1	0	0	0	0	456	1375,2	0	1	0	0
3000	1	0	0	0	0	294	788,1	0	1	0	0
3000	1	0	0	0	0	210,2	577,1	1	0	1	1
700	1	0	0	0	0	456,8	549,5	0	0	0	0
500	1	0	0	0	0	162,1	366,1	0	0	0	0
500	1	0	0	0	0	112	292,7	0	1	0	0
108	0	1	0	0	0	116	246,8	0	0	0	1
59	0	1	0	0	0	107,2	207,6	0	0	0	1
72	0	1	0	0	0	66,3	246,8	0	0	0	0
84	0	1	0	0	0	274,1	549,5	0	0	0	0
70	0	1	0	0	0	447,5	1100	0	0	0	0
51	0	1	0	0	0	262,9	549,5	0	0	0	0
72	0	1	0	0	0	180,2	338,5	0	0	0	0
30	0	1	0	0	0	123	265,1	0	0	0	0
50	0	0	1	0	0	12,8	27,5	1	0	0	0
40	0	0	1	0	0	5,22	17,3	1	0	1	1
35	0	0	1	0	0	5,2	13,3	1	0	1	1
35	0	0	0	1	0	5,6	14,7	0	0	1	1
35	0	0	0	1	0	5,6	14,7	0	0	1	1
15	1	0	0	0	0	4,2	9,5	1	1	1	1
300	1	0	0	0	0	45,9	139	1	1	1	1
20	1	0	0	0	0	5	10	1	0	0	1
20	1	0	0	0	0	5	10	1	0	1	0
1350	0	0	0	0	1	168	356,9	1	1	1	1
1350	0	0	0	0	1	168	356,9	1	1	1	1
1350	0	0	0	0	1	168	356,9	1	1	1	1
405	0	0	0	0	1	80	164,2	1	1	1	1
405	0	0	0	0	1	80	164,2	1	1	1	1
105	0	0	0	1	0	23	45	0	0	1	1
105	0	0	0	1	0	23	45	0	0	1	1
105	0	0	0	1	0	23	45	0	0	1	1
300	0	0	0	0	1	76,8	200,9	0	1	0	0
300	0	0	0	0	1	76,8	200,9	0	1	0	0
300	0	0	0	0	1	76,8	200,9	0	1	0	0
1000	0	0	0	0	1	154	347,7	0	1	1	1
1000	0	0	0	0	1	154	347,7	0	1	1	1
500	0	0	0	0	1	140,8	457,8	0	1	0	0
500	0	0	0	0	1	140,8	457,8	0	1	0	0
500	0	0	0	0	1	114,1	320,2	0	1	0	0
1500	1	0	0	0	0	153,6	357,2	1	0	0	0
1000	1	0	0	0	0	273	567,9	0	1	0	0
500	1	0	0	0	0	103,2	218,1	1	0	0	0
300	1	0	0	0	0	54,1	139	1	0	1	1
1000	1	0	0	0	0	72,4	246,8	1	0	1	0
5000	1	0	0	0	0	362,1	1100	1	0	1	0

energ	vykon	snizhm	kloub	fci	karton	plast	folie	let	cens
1	0	0	0	3	1	0	0	14	1
0	1	0	0	1	1	0	0	17	1
0	0	1	0	1	1	0	0	17	0
0	0	0	0	1	1	0	0	14	1
0	1	0	0	1	1	0	0	12	0
0	0	0	0	1	1	0	0	13	0
1	0	0	0	3	0	1	0	14	1
0	1	0	0	1	0	1	0	18	0
0	1	0	0	1	0	1	0	15	1
1	0	0	0	3	0	1	0	14	1
0	1	0	0	1	0	1	0	9	1
0	1	0	0	1	0	1	0	4	1
0	1	0	0	1	0	1	0	4	1
0	0	0	0	1	0	1	0	7	1
0	0	0	0	1	0	1	0	10	0
0	0	0	1	1	0	1	0	11	1
0	0	0	1	1	0	1	0	6	1
0	1	0	0	1	0	1	0	16	1
0	1	0	0	1	0	1	0	16	1
0	1	0	0	1	0	1	0	15	1
0	0	1	0	1	0	1	0	15	1
0	1	0	0	1	0	0	1	4	1
1	0	0	0	3	0	0	1	3	1
1	0	0	0	3	0	0	1	10	0
1	0	0	0	3	0	0	1	7	1
1	0	0	0	3	0	0	1	7	1
0	0	0	0	2	0	0	1	8	0
0	0	0	0	2	1	0	0	8	0
0	0	0	0	1	0	0	1	10	1
0	0	0	0	1	0	0	1	10	1
0	0	0	0	2	0	1	0	7	1
0	0	0	0	2	0	1	0	7	1
0	0	0	0	2	0	1	0	7	1
0	0	0	0	2	0	1	0	7	1
0	0	0	0	2	0	1	0	7	1
1	0	0	0	3	0	1	0	5	1
1	0	0	0	3	0	1	0	5	1
1	0	0	0	3	0	1	0	5	1
0	1	0	0	1	0	1	0	5	0
0	1	0	0	1	0	1	0	5	0
0	1	0	0	1	0	1	0	5	0
0	0	0	0	2	0	1	0	2	0
0	0	0	0	2	0	1	0	2	0
0	1	0	0	1	0	1	0	3	1
0	1	0	0	1	0	1	0	3	1
0	0	1	0	1	0	1	0	3	1
0	1	0	0	1	0	1	0	6	1
0	1	0	0	1	0	1	0	6	1
0	0	1	0	1	1	0	0	6	0
1	0	0	0	3	1	0	0	11	0
0	0	0	0	1	1	0	0	13	0
0	0	0	0	1	0	1	0	13	0

Literatura

[1] Aelen O. (1978): Nonparametric estimation of partial transition probabilities in multiple decrement models. *Annals of Statistics* **6(3)**, 457–481.

[2] Amemiya T. (1981): Qualitative response models: A survey. *Journal of Economic Literature* **19(4)**, 481–536.

[3] Anděl J. (2005): Základy matematické statistiky. Matfyzpress, Praha .

[4] Cipra T. (2008): Finanční ekonometrie. Ekopress.

[5] Cox D.R. (1972): Regression models and life-tables. *Journal of the Royal Statistical Society* **34(2)**, 187–220.

[6] EViews 5 user's guide. (2004): Quantitative Micro Software, LLC.

[7] Greene W.H. (2003): Econometric analysis. Prentice Hall, New York.

[8] Hoelzl E., Rustichini A. (2005): Overconfident: Do you put your money on it. *The Economic Journal* **115(April)**, 305–318.

[9] Kaplan E.L., Meier P. (1958): Non parametric estimation from incomplete observations. *Journal of the American statistical association* **53(June)**, 457–481.

[10] Maddala G.S. (1983): Limited dependent and qualitative variables in econometrics. Cambridge University Press.

[11] Pazdera J., Rychnovský M., Zahradník P. (2008): Survival analysis in credit scoring. Seminář: Modelování v ekonometrii, MFF UK, Praha.

[12] Reisnerová S. (2004): Analýza přežití a Coxův model pro diskretní čas. *In: Robust* **13**, 339–346.

[13] Tobin J. (1958): Estimation of relationships for limited dependent variables. *Econometrica* **26(1)**, 24–36.

[14] Zhou M. (2008): Use software R to do survival analysis and simulation. A tutorial. Kentucky, Free download, http://www.stat.nus.edu.sg/ stachenz/Rsurv.pdf.