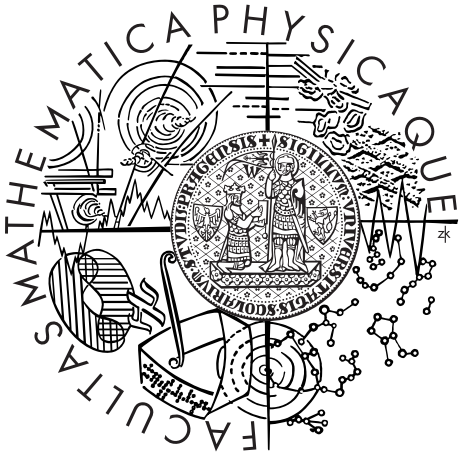
Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

BAKALÁŘSKÁ PRÁCE



Veronika Janíková

Spojování dat

Katedra pravděpodobnosti a matematické statistiky

Vedoucí bakalářské práce:	Mgr. Tomáš Hovorka
Studijní program:	Matematika
Studijní obor:	Obecná matematika

Praha 2014

Ráda bych poděkovala svému vedoucímu práce Mgr. Tomáši Hovorkovi nejen za odborné vedení, pomoc a rady při zpracování této práce, ale také za předání cenných zkušeností z praxe. Mé poděkování patří Prof. RNDr. Jaromíru Antochovi, CSc. za jeho ochotu, odborný dohled, věcné připomínky a zapůjčení literatury. Společnosti Median děkuji za poskytnutí dat nezbytných pro praktickou část práce. Děkuji také své rodině a přátelům za podporu a jazykovou korekturu.

Prohlašuji, že jsem tuto bakalářskou práci vypracovala samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle §60 odst. 1 autorského zákona.

V ..... dne ..... Podpis autora

Název práce: Spojování dat

Autor: Veronika Janíková

Katedra: Katedra pravděpodobnosti a matematické statistiky

Vedoucí bakalářské práce: Mgr. Tomáš Hovorka, Median s.r.o.

Abstrakt: Práce se zabývá spojováním databází, jakožto jednou z možností řešící velmi častý problém dostupnosti dat v praxi. Úvodem je zmíněno praktické využití fúze dat, zejména v oblasti marketingu, a základní algoritmy a problémy spojování dat. Hlavní část práce se pak zabývá takzvanou „statistickou fúzí bez omezení“. Nejprve je podrobně teoreticky popsána jedna z možností průběhu tohoto typu fúze, přičemž dochází k větvení na čtyři různé typy. Následně je za pomoci statistických ukazatelů navržena metoda teoretického vyhodnocení úspěšnosti obecné fúze. V praktické části práce je pomocí statistického programu R naprogramován průběh všech čtyř typů statistické fúze bez omezení i jejich následné vyhodnocení. Fúze je poté aplikována na námi vygenerovanou databázi a dále na skutečná data sesbíraná v praxi, která nám byla poskytnuta společností Median. V poslední části práce jsou pak interpretovány, vyhodnocovány a diskutovány výsledky fúzí na těchto dvou konkrétních databázích.

Klíčová slova: Spojování a integrace dat, modelování dat

Title: Data fusion

Author: Veronika Janíková

Department: Department of Probability and Mathematical Statistics

Supervisor: Mgr. Tomáš Hovorka, Median s.r.o.

Abstract: This bachelor's thesis deals with a data fusion, which is a one of the possible solutions to the common problem of data availability in praxis. In the first part, practical use of data fusion, especially in marketing, fundamental algorithms and data fusion problems are discussed. The main part of this thesis deals with the so-called "unconstrained statistical fusion". Firstly, one of the possible processes of this type of fusion is described theoretically in detail. This process involves branching into four different types of data fusion. Next, a method of theoretical evaluating the quality of the general fusion model is designed using statistical indicators. The practical part of the thesis contents processes of four types of unconstrained statistical fusion and their evaluation which both are programmed in statistical program R. Furthermore, the fusion is applied to our artificially generated database and also to a real data collected in praxis by Czech public opinion research company Median. In the very last part of the thesis, the results of fusions applied to these two databases are interpreted, evaluated and discussed.

Keywords: Data fusion, data integration, data modelling

Obsah

Seznam použitého značení

Seznam odpovídajícího značení v programu

Seznam použitých zkratek

Kapitola 1

Úvod

1.1 Motivace

Statistické zpracování dat patří mezi neustále se rozvíjející směry zkoumání, které v dnešní době nabývají stále většího významu. V řadě oborů, ať už přírodovědných, lékařských či technických, ale také v mikroekonomii, médiích a reklamě, hraje analýza dat velmi důležitou roli.

Při každodenní praxi se setkáváme s jedním ze zásadních problémů zpracování dat, kterým je jejich samotná dostupnost. I když je jich shromažďováno stále více, často jsou rozptýleny mezi velké množství zdrojů. Ve většině případů je pak získávání nových dat finančně náročně, složité či dokonce i nemožné. (van der Putten, 2000)

Chtěli bychom proto z již získaných databází vytěžit ještě více informací. Představme si, že máme dvě odlišné databáze obsahující napozorované hodnoty k sledovaným veličinám. Nabízí se otázka, zda by z nich nešlo vytvořit novou databázi, která by v každém pozorování obsahovala hodnoty ke všem veličinám z obou předchozích databází. Tímto problémem, který bývá často označen jako spojování či fúzování dat, se budeme v této práci podrobně zabývat.

1.2 Základní myšlenka fúze dat

Nejprve definujme, jak vypadají samotná data. Jejich vhodnou reprezentací je tabulka, ve které každý sloupeček představuje získané hodnoty sledované veličiny, kterou budeme nazývat proměnná. Počet proměnných představuje šířku tabulky, rozsah pozorování pak její délku.

Mějme nyní dvě takové tabulky – databáze. Nechť první obsahuje ke každé proměnné n_p napozorovaných hodnot a druhá n_d napozorovaných hodnot. Dolní indexy $_p$ a $_d$ objasníme níže. Představme si například, že každý řádek databází obsahuje odpovědi respondenta na různé otázky - proměnné, sesbírané při nějakém dotazování. Chtěli bychom nyní vytvořit novou databázi, která by byla rozšířením první databáze o proměnné druhé databáze a i-tý řádek nové databáze by obsahoval hodnoty všech těchto proměnných. Na proměnné druhé databáze ale i-tý respondent první databáze neodpověděl. Tyto hodnoty bychom nyní chtěli získat na základě odpovědí respondentů v druhé databázi.

Princip fúze dat předpokládá existenci proměnných, které se vyskytují v obou databázích. Například v marketingových datech a datech z průzkumu veřejného

mínění obsahujících informace o zákaznících se velmi často vyskytují sociálně-demografické proměnné jako je pohlaví, věk, zaměstnání, bydliště atd. Takovým proměnným budeme říkat spojovací.

Bez újmy na obecnosti předpokládejme, že se tyto proměnné nachází v posledních sloupcích první databáze a naopak v prvních sloupcích databáze druhé. Pro další použití bude vhodné rozdělit si hodnoty proměnných první databáze na spojovací, které označíme jako tabulku S_p, a na ostatní proměnné, které označíme jako tabulku A_p. Analogicky označme všechny spojovací proměnné druhé databáze jako tabulku S_d, a ty, které nejsou spojovací, jako tabulku B_d. Dolní indexy $_p$ a $_d$ opět objasníme níže. Tabulky S_p a S_d tedy obsahují ty samé proměnné, přičemž předpokládáme, že jsou v tabulkách ve stejném pořadí. Všechny hodnoty spojovacích proměnných dále sloučíme do tabulky S a to tak, že na prvních n_p řádcích se nachází hodnoty z tabulky S_p a pod nimi na řádcích $n_p+1,\ldots,n_p+n_d$ hodnoty z tabulky S_d. Jedná se tedy o sloupcové spojení tabulek S_p a S_d. Označme obecně takto definované sloupcové spojení dvou tabulek symbolem $\oplus_s$. Zápisem $S_p \oplus_s S_d$ tedy rozumíme tabulku S.

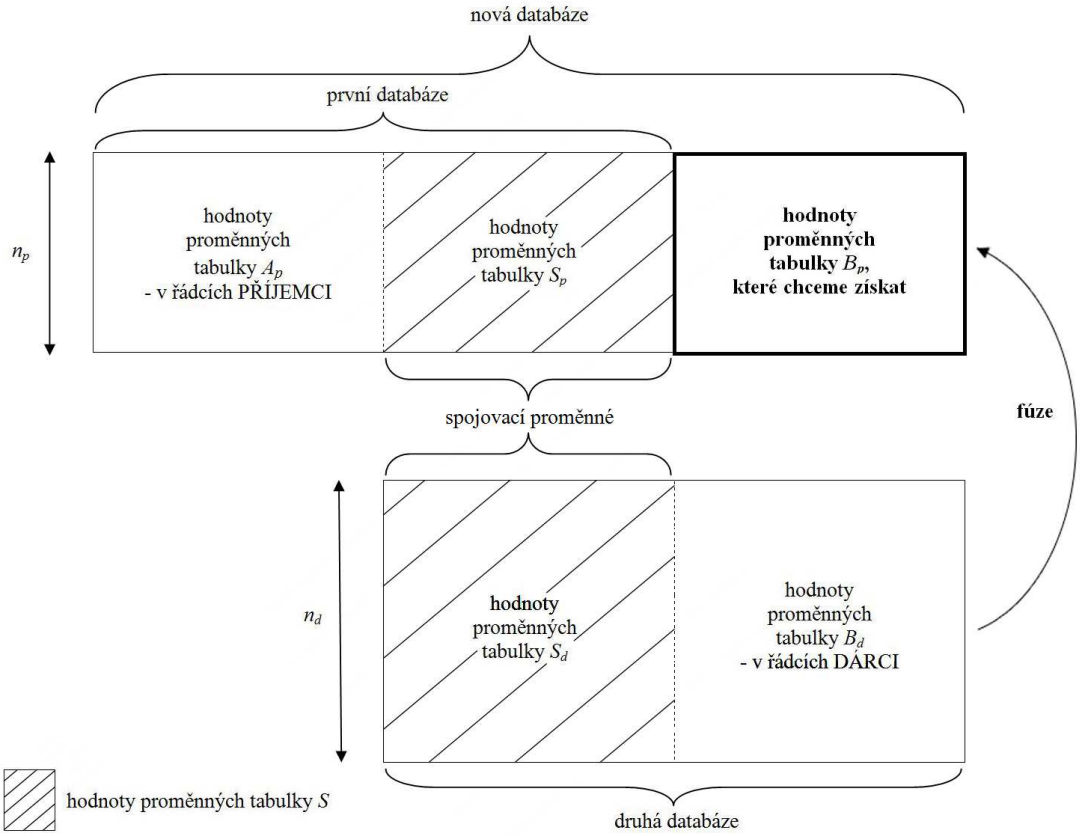
Analogicky definujme řádkové spojení dvou tabulek a označme ho symbolem $\oplus_r$. Při fúzi přidáme ke spojeným tabulkám $A_p \oplus_r S_p$ prázdné sloupce, které označíme jako B_p. Tyto sloupce budou reprezentovat proměnné druhé databáze, které nejsou spojovací, tedy proměnné tabulky B_d. V každém řádku budeme chtít doplnit prázdné hodnoty na základě informací z tabulky B_d. Ke každému řádku tabulky $A_p \oplus_r S_p$, budeme mu říkat příjemce - index $_p$, tedy chceme najít vhodný řádek tabulky $S_d \oplus_r B_d$, budeme ho nazývat dárce - index $_d$, tak aby po přiřazení hodnot dárce příjemci, nové hodnoty v každém řádku co nejlépe odpovídaly těm skutečným. Ty bychom získali, kdybychom při dotazování v první databázi zaznamenávali ke každému respondentovi navíc i jeho odpovědi na přidané proměnné. Pro lepší představu je na Obrázku 1.1 tato situace načrtnuta.

Obecně je tedy cílem fúze dat nalézt ke každému příjemci vhodného dárce na základě podobnosti hodnot jejich spojovacích proměnných. V průběhu fúze tedy pracujeme s tabulkami B_d a S. Na základě spárování dárců a příjemců jsou pak všechny nespojovací hodnoty příslušného dárce nacházející se v řádku tabulky B_d přeneseny do řádku tabulky B_p příslušícímu příjemci. V ideálním případě je nejlepším dárcem ten, který se s příjemcem shoduje ve všech hodnotách spojovacích proměnných. Tato situace je ale výjimečná, proto nejčastěji přiřazujeme toho dárce, jehož hodnoty jsou hodnotám příjemce nejvíce podobné.

Existuje mnoho faktorů, které obecně ovlivňují kvalitu fúze. Mezi ty nejdůležitější patří například podobnost populací, z kterých probíhá výběr příjemců i dárců. Například, budou-li jedna data představovat obecné chování obyvatelů České republiky, a na druhé straně budou data představující chování uživatelů internetu v České republice, nejspíš si propojení dárci a příjemci nebudou ve skutečnosti tak podobní, jak bychom pro další použití vzniklé databáze chtěli. Důležitou roli v tomto ohledu hraje také samotný rozsah databází a míra provázání spojovacích a fúzovaných proměnných.

Je zřejmé, že fúze bude tak úspěšná, nakolik se budou přiřazená data shodovat se skutečnými, neboli nakolik dobře spárujeme každého dárce s příjemcem. Algoritmů přiřazování je více. V následující podkapitole uvedeme ty nejznámější a přesvědčíme se, že vybrat nejlepší z nich není snadné.

Poznámka 1. Některé algoritmy fúze dat vyžadují počáteční splnění podmínky



Obrázek 1.1: Spojování dat

$n_d \geq n_p$, tedy aby dárců bylo alespoň tolik co příjemců. Jedná se zpravidla o algoritmy, při kterých se smí každý dárce přiřadit nejvýše jednomu příjemci. Naopak u algoritmů povolujících přiřazení stejného dárce více příjemcům, není splnění podmínky $n_d \geq n_p$ nutné.

1.3 Fúze dat v praxi

1.3.1 Poptávka po fúzi

Jak již bylo zmíněno, dostupnost dat je v řadě oborů zásadní problém. Nejvíce je fúze dat žádána v marketingových oborech, často ve spojení médií s reklamou.

Konkrétním příkladem, který se objevuje v mnoha článcích, například v Soong a de Montigny (2003b), Soong a de Montigny (2004) - project A, či Soong a de Montigny (2001), a na kterém je fúze dat nejčastěji vysvětlována, je takzvaná „TAM-TGI fusion“. TAM (Television Audience Measurement) je specializovaný obor výzkumu médií shromažďující podrobné informace o televizním publiku. Televize se téměř v každé zemi po celém světě stala dominantním médiem pro reklamu. To vedlo ke zvýšení poptávky po vysílání reklam, a reklamní agentury chtěly proto mít přesné informace o televizním divákovi. Naproti tomu TGI (Target Group Index) je celosvětový zdroj výzkumu trhu, shromažďující informace o marketingovém, mediálním a spotřebitelském chování obyvatelstva. Do výzkumu je zapojeno 60 zemí z celého světa, díky čemuž jsou každoročně získány

informace o více než 1,5 miliardě respondentů. (Nielsen, KantarMedia)

Na jedné straně máme tedy data o televizním divákovi, například jak často sleduje televizi, na jaké programy se dívá atd., na druhé straně máme data o jeho chování, např. co kupuje v supermarketu. Spojovací proměnné, tedy informace, které jsou uvedeny v obou databázích, jsou nejčastěji sociodemografické údaje. Provedeme-li pak na těchto databázích fúzi, dostaneme v našem příkladě informace o tom, jaké typy televizních diváků kupují v supermarketu určité výrobky. Toto je velmi cenná informace pro reklamní agentury, které pak například ví, před který pořad mají dát reklamu na určitý výrobek.

1.3.2 Jiné metody řešící problém spojování databází

Spojování dat lze provést i pomocí jiných metod než je fúze dat zavedená v sekci 1.2. Zmiňme například situaci, kdy bychom chtěli získat z tabulky B_d hodnoty pouze jedné proměnné. Pak daleko efektivnější metodou by v tomto případě bylo použití lineární regrese, jedná-li se o kardinální proměnnou, případně logistické regrese, jedná-li se o proměnnou binární (více o typech proměnných v sekci 5). Dopodrobna jsou tyto metody rozebrány v knize Zvára (2008). Kdybychom však postupnou regresí jednotlivých proměnných chtěli získat hodnoty více proměnných tabulky B_p, mohli bychom porušit vzájemné vazby mezi těmito proměnnými. Naproti tomu fúze dat díky přenesení celého řádku tabulky B_d příslušnému příjemci, tyto vzájemné vazby zachová.

1.3.3 Základní algoritmy spojování dat

Existuje mnoho způsobů fúzování dat. Každý algoritmus má velké množství různých variant, z nichž některé se často liší jen v malých detailech. Nyní stručně zmíníme základní typy, které jsou v praxi nejčastěji používány, diskutovány a publikovány.

Náhodná fúze

Při této metodě se příjemci a dárci přiřazují naprosto náhodně. Její úspěšnost závisí na předpokladu vzájemné statistické nezávislosti fúzovaných databází, která se však v praxi velmi často nedá předpokládat. Přesto se občas náhodná fúze provede i bez splnění tohoto předpokladu. (Soong a de Montigny, 2004)

Je zřejmé, že vzhledem k náhodnému spárování dárců a příjemců nejsou výsledky této metody v praxi použitelné. Náhodnou fúzi používáme především pro účely ověření stability fúze a pro porovnávání při aplikaci více typů fúzí na konkrétních datech.

Simulační metoda

Tato fúze je jakýmsi zobecněním náhodné fúze v případě, že není splněn předpoklad statistické nezávislosti proměnných a nechceme ho opomenout. Na základě vhodné kombinace proměnných dochází k rozdělení datové tabulky do jednotlivých částí, ve kterých je již splněn předpoklad nezávislosti. Na jednotlivé části lze tedy již aplikovat náhodnou fúzi. (Soong a de Montigny, 2004)

Statistická fúze

Jedná se o metodu, při které je každé spojovací proměnné přiřazena jakási váha, která představuje její význam vzhledem k přesnosti fúze. Na základě vah jsou určeny vzdálenosti mezi všemi potencionálními dárci a příjemci, pomocí nichž pak dochází k jejich spárování. Rozlišujeme dva základní typy této fúze.

První typ je známý pod anglickým názvem „Unconstrained Statistical Matching“, tedy v překladu statistická fúze bez omezení. Ta předpokládá vznik tabulky B_p tak, jak bylo definováno v sekci 1.2, tedy každý příjemce je v tabulce B_p zahrnut právě jednou. Na to, zda budou při přiřazování dárců příjemcům použiti všichni dárci nebo jen část, není kladeno žádné omezení. Tímto typem statistické fúze se budeme podrobně zabývat ve zbývajících kapitolách práce. Teoreticky popíšeme metody průběhu této fúze (viz kapitola 2), aplikujeme je na konkrétních datech (viz kapitola 3) a uvedeme a prodiskutujeme výsledky (viz kapitola 4). (Ingram a kol., 2000)

Naopak při „Constrained Statistical Matching“, neboli statistické fúzi s omezením je požadováno, aby tabulka B_p obsahovala hodnoty všech dárců, tedy aby byli při fúzi všichni dárci použiti alespoň jednou. Proto jsou při tvorbě tabulky B_p v případě většího počtu dárců než příjemců řádky některých příjemců duplikovány. (Ingram a kol., 2000)

Těmito dvěma typy statistické fúze se podrobně zabývá například článek Soong a de Montigny (2001), který je aplikuje na spojení TAM a TGI databází (viz 1.3.1).

Prediktivní izotonická fúze

Tato metoda zachovává všechny vlastnosti statistické fúze s omezením, přičemž její algoritmus je rychlejší a je přizpůsoben pro optimální přesnost pro určité kategorie dat. Fúze vychází z počátečního prediktivního modelu, typicky z modelu logistické regrese (viz Zvára (2008), kapitola 12). Omezená statistická fúze se pak provede pomocí odhadu prediktivního modelu. (Soong a de Montigny, 2004)

Jako u většiny algoritmů bychom i při fúzování dat chtěli, aby metody, které jsou přizpůsobeny pro řešení konkrétních problémů, nacházely dobrá řešení a aby jejich algoritmy byly co nejrychlejší. Takovýmto rychlým fúzím, přizpůsobeným pro určitá data, říkáme „fusion on the fly“, tedy jakési „fúze za pochodu“. (Soong a de Montigny, 2003a)

1.3.4 „Split-sample foldover test“

Nápadů, jak provést fúzi, je v praxi dostatek, ale musíme být schopni ji vyhodnotit. Hlavní metodou pro ověření výsledků je takzvaný „Split-sample foldover test“, při kterém je jedna konkrétní databáze rozdělena na dárce a příjemce a je určen blok, který bude následně fúzován příslušným algoritmem. Získané údaje jsou pak porovnávány s původními například na základě shody marginálních rozdělení, zachování závislostí a shody individuálních proměnných (viz sekce 2.2). Tím získáme představu o úspěšnosti fúze na daných datech. (Soong a de Montigny, 2003b)

1.3.5 Problém zobecnění

Jedním ze zásadních problémů spojování dat je vybrat ze všech metod tu „nejlepší“ ve smyslu největší shody nafúzovaných dat s daty skutečnými. Provedeme-li různé typy fúzí na konkrétních datech, porovnáním výsledků jsme schopni pokusit se vybrat ten nejlepší. Problém nastane v okamžiku, kdy ty samé metody použijeme na jiných datech. Ve výsledku často zjistíme, že pak je nejlepší metodou jiný typ fúze, než tomu bylo na prvních datech.

Obecně je problém nemožnosti vybrání nejlepšího obecného algoritmu známý pod názvem problém zobecnění. V kontextu metod řešících fúzi dat se tento problém také často nazývá „No free lunch theorem“ a poukazuje na nemožnost najít takovou metodu spojování dat, která by byla nejlepší pro obecná data. Z literatury zabývající se tímto problémem odkažme například na článek Soong a de Montigny (2004).

Východiskem z tohoto problému by mohlo být roztřídění dat do určitých specifických kategorií tak, aby pro každou kategorii již existovala právě jedna nejlepší metoda fúze. Pokud ale chceme rozhodnout o úspěšnosti fúze, potřebujeme znát skutečné hodnoty dat, které chceme získat.

Kapitola 2

Teorie

2.1 Metody

V této kapitole si popíšeme možný průběh jedné z nejčastějších metod spojování dat, a to statistickou fúzi bez omezení. Tato metoda je jedna z nejpopulárnějších hlavně díky tomu, že je intuitivní, relativně jednoduše realizovatelná, cenově efektivní, a nemá velké systémové požadavky.

2.1.1 Příprava dat

Budeme se držet značení zavedeného v sekci 1.2. K dispozici máme tabulky A_p a B_d, které obsahují n_p a n_d naměřených hodnot, a k nim příslušné tabulky spojovacích proměnných S_p a S_d, které jsme sloučili do tabulky S o délce $n_p + n_d$. Dolní indexy $_p$ a $_d$ nám symbolizují příjemce a dárce. Předpokládejme dále, že n_d je větší nebo rovno n_p. Na tomto základě bychom nyní chtěli vytvořit novou tabulku B_p tak, jak bylo ilustrováno na Obrázku 1.1, a vyplnit ji nafúzovanými daty.

Poznámka 2. Existence spojovacích proměnných je pro fúzi dat klíčová. Pokud bychom žádné takové proměnné neměli, mohli bychom provést pouze náhodnou fúzi. Získaná data by pak ve většině případů nebyla pro praktické účely použitelná.

Poznámka 3. Jelikož si uvedeme čtyři možné typy statistické fúze bez omezení, není ve smyslu poznámky 1 předpoklad $n_d \geq n_p$ nutný vždy.

Zaveďme ještě značení pro počet proměnných příslušných tabulek, neboli pro jejich šířky. Nechť p_A představuje počet proměnných tabulky A_p, analogicky nechť p_S představuje počet proměnných tabulky S (tedy i tabulek S_p a S_d) a p_B nechť představuje počet proměnných tabulky B_d, respektive B_p.

Poznámka 4. V následujících metodách fúze budeme pracovat pouze s tabulkami S_p, S_d a B_d. S daty uloženými v tabulce A_p při fúzi nepracujeme, jsou však důležitá pro reportování výsledků fúze, například při zjišťování zachování závislostí v databázích (viz sekce 2.2).

Poznámka 5. Pro úplnost lze ještě dodefinovat tabulku A_d rozsahu $n_d \times p_A$, jako tabulku, obsahující hodnoty proměnných tabulky A_p při pozorováních z tabulky B_d. Tabulka A_d by se tedy v Obrázku 1.1 nacházela pod tabulkou A_p

a doplňovala by obrázek do obdélníkového tvaru. V praxi jsou hodnoty tabulky A_d často součástí první databáze a jedná se o skutečné naměřené hodnoty, které pro nás však nemají žádné využití ať pro průběh samotné fúze, či v následujícím reportování.

Statistická interpretace dat

Je důležité si uvědomit, jak jsou naše data v tabulkách interpretována statisticky, přičemž předpokládáme znalost definic základních statistických pojmů (viz např. Anděl (2005), sekce 1.1). Označme $X = (a_1, \ldots, a_{p_A}, s_1, \ldots, s_{p_S}, b_1, \ldots, b_{p_B})^\top$ náhodný vektor, který má nějaké sdružené rozdělení a který se skládá z podvektorů $(a_1, \ldots, a_{p_A},)^\top$, $(s_1, \ldots, s_{p_S})^\top$ a $(b_1, \ldots, b_{p_B})^\top$. Pak obecný řádek tabulky $A_p \oplus_r S_p \oplus_r B_p$, případně tabulky $A_d \oplus_r S_d \oplus_r B_d$ představuje realizaci vektoru X. Obecné řádky tabulek A_p, S_p, B_p, respektive A_d, S_d, B_d pak představují realizaci podvektorů $(a_1, \ldots, a_{p_A},)^\top$, $(s_1, \ldots, s_{p_S})^\top$ a $(b_1, \ldots, b_{p_B})^\top$.

Neboť realizace vektoru X jsou nezávislé stejně rozdělené, pak všechny řádky tabulek $A_p \oplus_r S_p \oplus_r B_p$ a $A_d \oplus_r S_d \oplus_r B_d$ představují náhodný výběr z příslušného rozdělení o rozsahu $n_p + n_d$.

Klasifikace proměnných

Ve statistické analýze rozlišujeme různé typy proměnných. V literatuře se můžeme setkat s více způsoby klasifikace. My se budeme držet rozdělení zavedeného v sekcích 3.1 a 3.2 knihy Hebák a kol. (2004).

1. Podle počtu hodnot, které proměnné nabývají, rozlišujeme:
   (a) *Spojité* proměnné, které nabývají nespočetně nekonečného počtu hodnot. Vyjadřují nejčastěji váhu, míru či čas.
   (b) *Diskrétní* proměnné, které nabývají spočetného počtu hodnot. Nejčastěji vznikají počítáním, či hodnocením na vícebodové stupnici.
   (c) *Alternativní* (také *binární* či dichotomické) proměnné nabývají pouze dvou různých hodnot. Jejich častým příkladem je vyjádření postoje respondenta k nějaké otázce.
2. Z hlediska vyjádření hodnoty proměnné rozlišujeme:
   (a) *Kvalitativní* proměnné, jejichž hodnoty jsou vyjádřeny znaky a nelze s nimi tudíž provádět aritmetické operace. Podle typů vztahů mezi hodnotami je dále dělíme na:
      i. *Nominální* proměnné, o jejichž hodnotách můžeme říci pouze to, jestli jsou stejné či různé.
      ii. *Ordinální* (neboli pořadové) proměnné, které nabývají hodnot, u nichž můžeme navíc určit pořadí. To dává možnost vzestupně nebo sestupně uspořádat jednotlivé hodnoty v souboru.
   (b) *Kvantitativní* (neboli numerické či *kardinální*) proměnné, jejichž hodnoty jsou vyjádřeny číselně. Dále je dělíme opět podle typů vztahů mezi hodnotami na:

i. *Intervalové* proměnné, což jsou ordinální proměnné, pro jejichž dvě hodnoty můžeme navíc vypočítat, o kolik je jedna hodnota větší než druhá.

ii. *Poměrové* proměnné, což jsou intervalové proměnné, pro jejichž dvě hodnoty můžeme navíc vypočítat, kolikrát je jedna hodnota větší než druhá. Jedná se tedy typicky o kladné číselné hodnoty.

Nominální, ordinální a kardinální diskrétní proměnné bývají často souhrnně označovány jako *kategoriální*. Ty se pak dále dělí na dichotomické, které nabývají pouze dvou kategorií, a na vícekategoriální.

Pro naše účely budeme rozdělovat proměnné z tabulek A_p, B_d a S na kvalitativní a kardinální, případně konkrétněji na nominální, ordinální a kardinální.

Umělé indikátorové proměnné

V oblasti statistiky a ekonometrie, zejména v regresní analýze, se často objevuje pojem takzvaných umělých indikátorových proměnných, dále jen indikátorových. Jedná se v podstatě o alternativní proměnné, které nabývají hodnot 0 a 1, čímž poukazují na nepřítomnost, respektive přítomnost, určitého znaku. Tyto proměnné jsou používány pro třídění dat do vzájemně se vylučujících kategorií, například kuřák/nekuřák.

Pro použití obecných proměnných ve většině metod statistické analýzy je třeba určité typy převést na soubor indikátorových proměnných. Konkrétně všechny proměnné, jejichž hodnoty jsou znakové, neboť s nimi nelze provádět aritmetické operace, které jsou pro většinu statistických metod klíčové, a které budou převodem umožněny. V našem případě se tedy jedná o všechny kvalitativní proměnné.

Poznámka 6. V praxi se často stává, že hodnoty ordinálních proměnných jsou zastoupeny čísly. Například proměnná vzdělání může být definována jako proměnná nabývající hodnot 1 - 4, přičemž hodnoty 1 nabývá, pokud má respondent základní vzdělání, hodnoty 2, má-li středoškolské, hodnoty 3, má-li vyšší odborné a hodnoty 4, má-li vysokoškolské. V takovém případě nastává otázka, zda by měly být i tyto „číselné ordinální“ proměnné převedeny na soubor indikátorových. Tato volba závisí na konkrétních datech a z hlediska průběhu fúze není špatně, pokud převedeme i všechny tyto proměnné, popřípadě jen některé z nich. Jedná se tedy o jakési rozvětvení fúze, které může být zajímavé z hlediska diskuse výsledků na konkrétních datech. V praxi je také třeba přihlédnout na výpočetní a paměťovou náročnost algoritmu fúze, která se s každou nově převedenou proměnnou zvyšuje.

Danou proměnnou převedeme na soubor indikátorových tak, že místo ní vytvoříme právě tolik nových proměnných, kolika nabývá hodnot. Každá nová proměnná nyní reprezentuje každou jednu z nabývaných hodnot. Měla-li v i-tém pozorování původní proměnná určitou hodnotu, nabývá nyní proměnná symbolizující tuto hodnotu v i-tém pozorování hodnoty jedna, zatímco všechny ostatní proměnné reprezentující zbylé hodnoty nabývají v i-tém pozorování hodnoty nula.

Příklad. Zvolme jako původní proměnnou rodinný stav. Předpokládejme, že tato proměnná nabývá pěti znakových hodnot: svobodný/á, ženatý/vdaná,

druh/družka, rozvedený/á, vdovec/vdova. Při převodu se vytvoří pět nových proměnných, z nichž každá představuje jeden konkrétní stav. V následující Tabulce 2.1 pak na krátkém pozorování vidíme, jak se na základě původní proměnné přiřazovaly hodnoty novým proměnným.

rodinný stav		**svobodný/á**	**ženatý /vdaná**	**druh /družka**	**rozvedený/á**	**vdovec /vdova**
ženatý/vdaná		0	1	0	0	0
svobodný/á	převod	1	0	0	0	0
rozvedený/á	$\longrightarrow$	0	0	0	1	0
druh/družka		0	0	1	0	0
svobodný/á		1	0	0	0	0
...		...	...	...	...	...

Tabulka 2.1: Převod proměnné na soubor indikátorových proměnných

Tímto způsobem tedy převedeme všechny kvalitativní proměnné tabulek S a B_d, čímž se nám zvětší šířka každé tabulky, mnohdy podstatně. Pro lepší orientaci označme horním indexem *ind* tabulky, jejichž kvalitativní proměnné již prošly převodem na soubor indikátorových. Máme tedy tabulky S^{ind} a B_d^{ind}. Analogicky označme i jejich nové šířky jako p_S^{ind} a p_B^{ind}, přičemž platí $p_S^{ind} \geq p_S$ a $p_B^{ind} \geq p_B$.

Poznámka 7. Tabulku A_p nemá smysl převádět, neboť v metodách fúze využíváme pouze tabulky B_d a S.

2.1.2 Odstranění lineárních závislostí

Nyní máme nachystána data, která jsou uvnitř každé z tabulek navzájem závislá. Například proměnná, popřípadě soubor indikátorových proměnných, obsahující informace o vzdělání respondenta, může záviset na proměnné obsahující jeho věk. Pro výpočetní zjednodušení dalších kroků fúze je vhodné tyto závislosti v tabulkách S^{ind} a B_d^{ind} postupně odstranit. Tyto závislosti by totiž mohly ovlivnit správnost výsledků metod, které budeme používat dále. Mohlo by se také stát, že by určitou metodu nebylo možno vůbec provést. Navíc odstraněním závislostí mezi proměnnými se nám může zredukovat jejich počet.

Je třeba zdůraznit, že nežádoucí závislosti jsou vždy pouze ty mezi proměnnou z tabulky S^{ind}, respektive B_d^{ind} a ostatními proměnnými této tabulky. Vzájemné závislosti mezi proměnnými tabulky S_d^{ind} a proměnnými tabulky B_d^{ind} jsou pro nás naopak klíčové pro samotnou fúzi.

Faktorová analýza

Odstranit vnitřní závislosti mezi jednotlivými proměnnými tabulky S^{ind}, respektive B_d^{ind} lze více způsoby. My použijeme metodu zvanou faktorová analýza. Jedná se o vícerozměrnou techniku, která vyšetřuje vnitřní souvislosti a vztahy mezi proměnnými a odhaluje základní strukturu dat. Používá se pro kardinální proměnné, což máme díky převodu kvalitativních proměnných na soubory indikátorových proměnných splněno. Pro naše účely si uvedeme pouze základní myšlenky faktorové analýzy. Jejich matematické odvození je složitější a poněkud

rozsáhlé a je vysvětleno například v knize Anděl (1985), sekce XVII.4, popřípadě o něco více podrobněji v kapitole 19 knihy Hebák a kol. (2005b).

Jedním ze základních cílů faktorové analýzy je zjistit za pomoci posouzení struktury vztahů sledovaných proměnných, zda je možné tyto proměnné rozdělit do skupin tak, aby proměnné ze stejné skupiny spolu více korelovaly než proměnné z odlišných skupin. Dalším cílem je pak myšlenka redukce dat, která ale poněkud ustupuje před potřebou vysvětlit napozorované korelace pomocí nových nepozorovaných a svou podstatou hypotetických proměnných - takzvaných faktorů. (Hebák a kol., 2005b)

Matematicky lze postup faktorové analýzy popsat pomocí následujícího modelu. Nechť $\boldsymbol{X}$ je p_B^{ind}-rozměrný vektor představující řádky tabulky B_d^{ind}, respektive p_S^{ind}-rozměrný vektor představující řádky tabulky S^{ind}. Označme vektor jeho středních hodnot $\boldsymbol{\mu}$. Obecný model faktorové analýzy předpokládá existenci R v pozadí stojících společných faktorů F_1,F_2,$\dots F_R$, kterých je méně než p_B^{ind}, respektive p_S^{ind}. Jsou takové, že vektor $\boldsymbol{X}$ lze vyjádřit jako

$$\boldsymbol{X} = \boldsymbol{\mu} + \boldsymbol{\Gamma}\boldsymbol{f} + \boldsymbol{\varepsilon}, \tag{2.1}$$

kde $\boldsymbol{\Gamma}$ je takzvaná matice faktorových zátěží typu $p_B^{ind} \times R$, respektive $p_S^{ind} \times R$, $\boldsymbol{f}$ je R-členný vektor společných faktorů F_1,F_2,$\dots F_R$ a $\boldsymbol{\varepsilon}$ je vektor chybových složek délky p_B^{ind}, respektive p_S^{ind}, označovaných jako specifické faktory. Pro faktorový model (2.1) předpokládáme, že společné faktory F_1,F_2,$\dots F_R$ jsou nezávislé a stejně rozdělené náhodné veličiny s nulovými středními hodnotami a jednotkovými rozptyly. Výchozím úkolem analýzy pro faktorový model (2.1) je odhad faktorových zátěží a rozptylů specifických faktorů. K tomuto odhadu bylo vytvořeno mnoho postupů, které se často nazývají metody extrakce faktoru. (Hebák a kol., 2005b)

Zásadním problémem faktorové analýzy je stanovení počtu faktorů. Je totiž nutné jej stanovit dopředu ještě před provedením samotné analýzy. V nejjednodušším případě můžeme na základě zkoumaných proměnných použít teoretický předpoklad o počtu faktorů. Často je však nutné využít k tomuto účelu jiných statistických metod, například analýzu hlavních komponent (viz Anděl (1985), sekce XVII.2) či metodu maximální věrohodnosti (viz Anděl (1985), sekce XV.6). V našem případě využijeme toho, že ke každému faktoru jsme schopni určit takzvanou standardní odchylku faktoru, vyjadřující jeho důležitost vzhledem k ostatním faktorům. Na základě konkrétních dat určíme vhodnou hranici a za nové proměnné vezmeme jen takové faktory, jejichž standardní odchylka je větší než tato hranice.

Záměnou všech proměnných tabulek S^{ind} a B_d^{ind} za příslušný počet vzniklých faktorů nám vzniknou nové tabulky, označme je S^{fa} a B_d^{fa}. Analogicky označme jejich šířky jako p_S^{fa} a p_B^{fa}. Jelikož vzniklých faktorů je méně, než bylo proměnných před faktorovou analýzou, platí vztahy $p_S^{fa} \leq p_S^{ind}$ a $p_B^{fa} \leq p_B^{ind}$.

Poznámka 8. Uvažme nyní následující situaci. Nechť máme l proměnných vyjadřujících stejnou vlastnost, $l > 1$, a jednu proměnnou na nich nezávislou. Pak faktorová analýza vytvoří z l proměnných jeden faktor a z jedné nezávislé proměnné také jeden faktor. Tyto dva faktory jsou nyní rovnocenné, i když jeden z nich reprezentuje l proměnných a druhý pouze jednu proměnnou. Nabízí se otázka, zda by se tato skutečnost neměla ve fúzi nějak projevit. V praxi se tento problém řeší podle konkrétních dat, která fúzujeme. V takovýchto situacích máme

možnost volby, zda se budeme tímto problémem zabývat. Pokud se rozhodneme faktor reprezentující l proměnných před druhým upřednostnit, často to děláme pomocí vlastních čísel, která vystupují v průběhu faktorové analýzy. Pomocí nich je pak oběma faktorům přidělena takzvaná faktorová váha, přičemž váha prvního je l-krát větší než váha druhého.

Poznámka 9. Odstranit vnitřní závislosti v tabulkách B_d^{ind} a S^{ind} lze více metodami. Lze je například odstranit i ručně. V praxi se však osvědčilo používat pro odstranění těchto závislostí faktorovou analýzu.

2.1.3 Nalezení vzdáleností mezi dárci a příjemci

Nyní využijeme vzájemných závislostí mezi proměnnými tabulky S_d^{fa} a proměnnými tabulky B_d^{fa}. Pomocí těchto závislostí určíme ke každé dvojici příjemce - dárce číslo, budeme ho nazývat vzdálenost, které bude symbolizovat jak moc jsou si podobní. Tím se dostáváme do klíčové části fúze, neboť budeme-li znát tyto vzdálenosti, budeme podle nich schopni nalézt k danému příjemci vhodného dárce, a přenesením jeho hodnot na hodnoty příjemce fúzi dokončit. Možností nalezení vzdáleností je opět více. My jsme se rozhodli využít váženého průměru odlišností hodnot.

Spočítání vah spojovacích proměnných

Pro zjištění vzdáleností je nejprve nutné ke každé spojovací proměnné tabulky S_d^{fa} určit hodnotu, která bude představovat, jak moc závisí daná spojovací proměnná na proměnných z tabulky B_d^{fa}. Tuto hodnotu nazýváme váha proměnné a určíme ji následujícím postupem.

Lineární regresní modely

Závislost všech proměnných tabulky B_d^{fa} na jednotlivé spojovací proměnné lze určit pomocí mnohorozměrného lineárního regresního modelu. Jelikož se nám jedná o závislosti mezi tabulkou B_d^{fa} a k ní příslušnou tabulkou spojovacích proměnných S_d^{fa}, bereme hodnoty spojovacích proměnných pouze z této tabulky. Označme $\boldsymbol{X} = (x_1, x_2, \ldots, x_{n_d})^\top$ náhodný vektor představující obecnou spojovací proměnnou tabulky S_d^{fa}. Nechť matice $\boldsymbol{Y}$ představuje celou tabulku B_d^{fa}, je tedy rozsahu $n_d \times p_B^{fa}$. Pak lineární model popisující naši situaci zapisujeme v maticovém tvaru

$$\boldsymbol{Y}_{n_d \times p_B^{fa}} = \boldsymbol{X}_{n_d \times 1} \boldsymbol{\beta}_{1 \times p_B^{fa}} + \boldsymbol{\varepsilon}_{n_d \times p_B^{fa}}, \tag{2.2}$$

kde $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_{p_B^{fa}})$ je vektor neznámých parametrů modelu a $\boldsymbol{\varepsilon}$ je matice nepozorovatelné rušivé složky typu $n_d \times p_B^{fa}$. (Hebák a kol. (2005a), Timm (2002))

Cílem lineárního regresního modelu je objasnit vztah mezi vysvětlovanou proměnnou $\boldsymbol{Y}$ a vysvětlující proměnnou $\boldsymbol{X}$ na základě odhadů parametrů $\beta_1, \ldots, \beta_{p_B^{fa}}$. Tím se zabývají různé vícerozměrné statistické metody. My použijeme takzvanou mnohorozměrnou analýzu rozptylu, které se také zkráceně říká „MANOVA“ z anglického „Multivariate analysis of variance“ (viz Timm (2002) kapitola 4).

V praxi se osvědčilo používat pro výpočet váhy každé spojovací proměnné například takzvaný koeficient determinace R^2, známý pod anglickým názvem „R-squared“, který vyjadřuje, jakou část z celkové variability hodnot vysvětlované proměnné lineární model objasňuje. Jeho odvození je pro mnohorozměrnou analýzu rozptylu podrobně popsáno v sekci 4.2.d knihy Timm (2002). Výsledkem odvození je však matice. Abychom dostali skalár, můžeme podle zmíněného zdroje použít její determinant, případně stopu. My přiřadíme koeficientu R^2 stopu této matice. Aby se hodnota koeficientu pohybovala v intervalu [0,1], vydělíme ji ještě počtem závisle proměnných v daném modelu, což je vyjádřeno počtem prvků na diagonále matice. Čím větší hodnoty koeficientu determinace tím je obvykle model lepší.

Poznámka 10. Obecně lze za váhu proměnné vzít jakýkoliv ukazatel, který měří závislost mezi spojovacími proměnnými tabulky S_d^{fa} a proměnnými tabulky B_d^{fa}.

Algoritmus přidělování vah spojovacím proměnným

Nejdříve určíme hodnoty R^2 pro model (2.2), přičemž za $\mathbf{X}$ budeme postupně dosazovat jednotlivé spojovací proměnné. Je zřejmé, že proměnná z modelu, který má tuto hodnotu největší ze všech, závisí na proměnných tabulky B_d^{fa} nejvíce. Proto jí odpovídající hodnotu koeficientu determinace přiřadíme za váhu. Označme tuto proměnnou $\boldsymbol{Z} = (z_1, z_2, \ldots, z_{n_d})^\top$.

Tím jsme určili váhu první spojovací proměnné, kterou bychom pro další počítání ostatních vah měli do modelu zahrnout. Teoreticky bychom nyní chtěli vysvětlit závislost všech proměnných tabulky B_d^{fa} na proměnných $\boldsymbol{X}$ a $\boldsymbol{Z}$, kde vektor $\boldsymbol{X}$ nyní představuje spojovací proměnnou, které ještě nebyla přiřazena váha. Dostáváme tedy nový lineární model, který lze zapsat v maticovém tvaru

$$\boldsymbol{Y}_{n_d \times p_B^{fa}} = \begin{pmatrix} \boldsymbol{Z} & \boldsymbol{X} \end{pmatrix}_{n_d \times 2} \boldsymbol{\beta}_{2 \times p_B^{fa}} + \boldsymbol{\varepsilon}_{n_d \times p_B^{fa}}, \tag{2.3}$$

kde $\begin{pmatrix} \boldsymbol{Z} & \boldsymbol{X} \end{pmatrix}$ představuje matici typu $n_d \times 2$, jejíž první sloupec je vektor $\boldsymbol{Z}$ a druhý sloupec je vektor $\boldsymbol{X}$.

Nyní určíme hodnoty R^2 pro model (2.3), přičemž za $\mathbf{X}$ dosazujeme jen ty spojovací proměnné, které ještě nemají přiřazenou váhu. Opět z nich vybereme maximální hodnotu a proměnné, z jejíž modelu byla tato maximální hodnota získána, chceme nyní přiřadit váhu. Hodnota této váhy nebude R^2, ale od tohoto koeficientu je třeba odečíst maximum z koeficientů R^2 stanovených v předchozím kroku. Jelikož na pravé straně modelu máme nyní ve sloupcích příslušné matice dvě proměnné $\boldsymbol{Z}$ a $\boldsymbol{X}$, představuje koeficient R^2 tohoto modelu jejich společnou váhu. Ta je zřejmě větší nebo rovna váze proměnné $\boldsymbol{Z}$ a tedy pro přiřazení váhy proměnné $\boldsymbol{X}$ je nutné od jejího koeficientu R^2 odečíst koeficient R^2 stanovený na základě předchozího modelu proměnné $\boldsymbol{Z}$.

Obecně tedy v k-tém kroku, $k \in \{2, \ldots, p_S^{fa}\}$ určíme váhu následovně. Na pravou stranu modelu použitého v kroku $k-1$ přidáme zprava do příslušné matice nový sloupec představující proměnnou, jejíž váhu jsme v kroku $k-1$ určili. Pro tento model určíme hodnoty R^2, přičemž za $\mathbf{X}$ dosazujeme ty spojovací proměnné, které ještě nemají přiřazenou váhu. Tím získáme $(p_S^{fa} - k + 1)$ koeficientů determinace, označme je $R_1^2, \ldots, R^2_{p_S^{fa}-k+1}$. Z nich určíme maximum, označme jej R_k^2. Proměnné, z jejíž modelu byl tento koeficient vypočítán, přiřadíme váhu

$R^2_k - R^2_{k-1}$, kde R^2_{k-1} představuje maximum ze všech hodnot R^2 počítaných v předchozím kroku. Tak postupně přiřazujeme váhu jednotlivým spojovacím proměnným, přičemž váha proměnné přiřazená v kroku k je menší nebo rovna váze proměnné v kroku $k-1$.

Je zřejmé, že algoritmus se zastaví ve chvíli, kdy máme určené váhy ke všem spojovacím proměnným. V praxi ale často na základě konkrétních dat volíme ještě další ukončovací podmínky pro tento algoritmus. Například se určí hranice, představující jaká nejmenší váha může být proměnné přiřazena. Je zřejmé, že je-li v k-tém kroku přiřazena některé proměnné váha menší nebo rovna této hranici, bude tato nerovnost platit i pro váhy, které bychom získali ve všech následujících krocích. Potom v k-tém kroku algoritmus ukončíme, přičemž příslušné proměnné i všem zbylým proměnným přiřadíme nulovou váhu. Ta je jim obecně přiřazena vždy, když je algoritmus zastaven na základě platnosti jakékoli ukončovací podmínky.

Matice vzdáleností

Díky vahám jsme nyní schopni vytvořit matici, označme ji například $\boldsymbol{V}$, jejíž prvek v_{ij} bude představovat vzdálenost i-tého příjemce od j-tého dárce. Jedná se tedy o matici typu počet příjemců × počet dárců.

Hodnoty matice $\boldsymbol{V}$ určíme následovně. Podle již zavedeného značení představuje n_p délku tabulky A^{fa}_p, tedy počet příjemců, a n_d délku tabulky B^{fa}_d, tedy počet dárců. Označme $\boldsymbol{S} = (s_{k,l})$ matici představující tabulku S^{fa}, tedy tabulku hodnot všech spojovacích proměnných, přičemž víme, že jich je p^{fa}_S. Pak $\forall l \in \{1,\ldots,p^{fa}_S\}$ představuje vektor $(s_{1,l}, s_{2,l}, \ldots, s_{n_p+n_d,l})^\top$ l-tou spojovací proměnnou a nechť $w_l \in \mathbb{R}$ představuje její váhu. Pak se každý prvek matice $\boldsymbol{V}$ dá spočítat jako

$$v_{ij} = f((s_{i,1},s_{i,2},\ldots,s_{i,p^{fa}_S})^\top,(s_{n_p+j,1},s_{n_p+j,2},\ldots,s_{n_p+j,p^{fa}_S})^\top,(w_1,w_2,\ldots,w_{p^{fa}_S})^\top), \tag{2.4}$$

$\forall i = 1,2,\ldots,n_p$, $\forall j = 1,2,\ldots,n_d$, kde f je nějaká funkce veličin $(s_{i,1},s_{i,2},\ldots,s_{i,p^{fa}_S})^\top$, $(s_{n_p+j,1},s_{n_p+j,2},\ldots,s_{n_p+j,p^{fa}_S})^\top$ a $(w_1,w_2,\ldots,w_{p^{fa}_S})^\top$.

Mahalanobisova vzdálenost

Pro náš typ dat je vhodné zvolit za f takzvanou výběrovou váženou Mahalanobisovu vzdálenost, označme ji d^2_M. Ta je například v knize Timm (2002) definovaná jako:

$$d^2_M(\boldsymbol{X_u},\boldsymbol{X_v},\boldsymbol{W}) = (\boldsymbol{X_u} - \boldsymbol{X_v})^\top \cdot \boldsymbol{W}^\top \cdot \boldsymbol{K}^{-1} \cdot \boldsymbol{W} \cdot (\boldsymbol{X_u} - \boldsymbol{X_v}), \tag{2.5}$$

kde $\boldsymbol{X_u} = (X_{u1},\ldots,X_{uq})^\top$ a $\boldsymbol{X_v} = (X_{v1},\ldots,X_{vq})^\top$, $q \in \mathbb{N}$ jsou realizace vektorů z náhodného výběru $\boldsymbol{X_1},\ldots,\boldsymbol{X_p}$, $p \in \mathbb{N}$, $\boldsymbol{W}$ je diagonální matice typu $q \times q$, která má na diagonále váhy prvků náhodného výběru $(X_{11},X_{21},\ldots,X_{p1})^\top,(X_{12},X_{22},\ldots,X_{p2})^\top,\ldots,(X_{1q},X_{2q},\ldots,X_{pq})^\top$ a $\boldsymbol{K}$ je výběrová kovarianční matice typu $q \times q$ tohoto výběru. Ta je definována v sekci 6.2 knihy Anděl (2005).

Vzorec (2.5) nyní aplikujeme na náš problém pomocí značení zavedeného výše. Položme

$$p := n_p + n_d, \quad q := p^{fa}_S, \quad u := i, \quad v := n_p + j,$$

a dále

$$\boldsymbol{X_i} := (s_{i,1}, s_{i,2}, \ldots, s_{i,p_S^{fa}})^\top \text{ a } \boldsymbol{X_j} := (s_{n_p+j,1}, s_{n_p+j,2}, \ldots, s_{n_p+j,p_S^{fa}})^\top.$$

Realizace vektorů $\boldsymbol{X_i}$ a $\boldsymbol{X_j}$ tedy představuje hodnoty i-tého a j-tého řádku tabulky S. Vypočtené váhy spojovacích proměnných jsou pak diagonálními prvky matice $\boldsymbol{W}$ typu $p_S^{fa} \times p_S^{fa}$, tedy

$$\boldsymbol{W} := \begin{pmatrix} w_1 & 0 & \ldots & 0 \\ 0 & w_2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \ldots & 0 & w_{p_S^{fa}} \end{pmatrix}.$$

Matice $\mathbf{K}$ je výběrová kovarianční matice výběru vektorů hodnot spojovacích proměnných, je tedy typu $p_S^{fa} \times p_S^{fa}$. Z její definice (viz Anděl (2007), sekce 3.2) plyne, že její diagonální prvky jsou vždy rovny rozptylu vektoru hodnot představující příslušnou spojovací proměnnou, zatímco nediagonální prvky představují vzájemnou kovarianci těchto vektorů. Jelikož jsou vektory hodnot spojovacích proměnných faktory, z části 2.1.2 víme, že jsou navzájem nezávislé a mají jednotkové rozptyly. Tedy všechny diagonální prvky matice $\mathbf{K}$ jsou rovny jedné. Díky nezávislosti všech faktorů je jejich kovariance nulová (viz Anděl (2007), věta 3.8) a tudíž i všechny nediagonální prvky matice $\mathbf{K}$ jsou rovny nule. Matice $\mathbf{K}$ je tedy v našem případě rovna jednotkové matici. Vzorec (2.5) nyní můžeme psát jako

$$d_M^2 = \left[\begin{pmatrix} s_{i,1} \\ s_{i,2} \\ \vdots \\ s_{i,p_S^{fa}} \end{pmatrix} - \begin{pmatrix} s_{n_p+j,1} \\ s_{n_p+j,2} \\ \vdots \\ s_{n_p+j,p_S^{fa}} \end{pmatrix} \right]^\top \begin{pmatrix} w_1 & 0 & \ldots & 0 \\ 0 & w_2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \ldots & 0 & w_{p_S^{fa}} \end{pmatrix} \cdot$$

$$\cdot \begin{pmatrix} 1 & 0 & \ldots & 0 \\ 0 & 1 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \ldots & 0 & 1 \end{pmatrix} \begin{pmatrix} w_1 & 0 & \ldots & 0 \\ 0 & w_2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \ldots & 0 & w_{p_S^{fa}} \end{pmatrix} \left[\begin{pmatrix} s_{i,1} \\ s_{i,2} \\ \vdots \\ s_{i,p_S^{fa}} \end{pmatrix} - \begin{pmatrix} s_{n_p+j,1} \\ s_{n_p+j,2} \\ \vdots \\ s_{n_p+j,p_S^{fa}} \end{pmatrix} \right]$$

$$= \left[\begin{pmatrix} s_{i,1} \\ s_{i,2} \\ \vdots \\ s_{i,p_S^{fa}} \end{pmatrix} - \begin{pmatrix} s_{n_p+j,1} \\ s_{n_p+j,2} \\ \vdots \\ s_{n_p+j,p_S^{fa}} \end{pmatrix} \right]^\top \begin{pmatrix} w_1^2 & 0 & \ldots & 0 \\ 0 & w_2^2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \ldots & 0 & w_{p_S^{fa}}^2 \end{pmatrix} \cdot$$

$$\cdot \left[\begin{pmatrix} s_{i,1} \\ s_{i,2} \\ \vdots \\ s_{i,p_S^{fa}} \end{pmatrix} - \begin{pmatrix} s_{n_p+j,1} \\ s_{n_p+j,2} \\ \vdots \\ s_{n_p+j,p_S^{fa}} \end{pmatrix} \right] = \sum_{l=1}^{p_S^{fa}} (s_{i,l} - s_{n_p+j,l})^2 \cdot w_l^2$$

A tedy dosazením d_M^2 za f ve vzorci (2.4) dostáváme

$$v_{ij} = \sum_{l=1}^{p_S^{fa}} (s_{i,l} - s_{n_p+j,l})^2 \cdot w_l^2, \quad \forall i = 1,2,\ldots,n_p, \quad \forall j = 1,2,\ldots,n_d, \tag{2.6}$$

Pomocí vzorce (2.6) spočítáme všechny prvky matice vzdáleností $\boldsymbol{V}$.

2.1.4 Přidělení dárců příjemcům

Z matice vzdáleností nyní víme, jak odlišní jsou všichni příjemci a dárci. Chtěli bychom samozřejmě ke každému příjemci vybrat toho nejlepšího dárce. Přestože to individuálně může znamenat toho nejbližšího, z hlediska kvality celé fúze to nejlepší výběr být nemusí. Nabízí se také otázka, jestli je lepší použít každého dárce nejvýše jednou, či ho vzít vícekrát. Průběh fúze se tady může opět rozcházet. Nyní zmíníme čtyři základní algoritmy používané při přidělování dárců příjemcům. Z části 1.3.5 první kapitoly ale již víme, že určit nejlepší z nich na obecných datech nelze.

Připomeňme, že symbol v_{ij}, kde $i \in \{1, \ldots, n_p\}$, $j \in \{1, \ldots, n_d\}$, značí prvek matice $\boldsymbol{V}$.

Nejbližšímu

Metoda přidělování dárce nejbližšímu příjemci se může na první pohled jevit jako nejlepší možnost. Jak už je v názvu uvedeno, jedná se o algoritmus, který každému příjemci přiřadí nejbližšího dárce bez ohledu na to, jestli už byl tento dárce předtím použit. V praxi procházíme matici vzdáleností $\boldsymbol{V}$ po řádcích a v každém určíme minimální prvek. Pro každého i-tého příjemce, $i = 1, \ldots, n_p$, tedy určíme jeho dárce j podle

$$j = \underset{l \in \{1,\ldots,n_d\}}{\operatorname{argmin}} v_{il}.$$

Od minima

Přiřazování dárce příjemci takzvaně od minima je podobné předchozí metodě s tím rozdílem, že hledáme minimální prvek celé matice $\boldsymbol{V}$ a přiřadíme-li příjemci nějakého dárce, již tohoto dárce nesmíme znovu použít pro dalšího příjemce. Hledáme tedy minimum matice $\boldsymbol{V}$ a jakmile takový prvek, označme jej v_{ij}^{min}, najdeme, přiřadíme i-témů příjemci j-tého dárce. Platí

$$v_{ij}^{min} = \min_{\substack{k \in \{1,\ldots,n_p\} \\ l \in \{1,\ldots,n_d\}}} v_{kl}.$$

V dalším kroku pak hledáme znovu minimum matice $\boldsymbol{V}$, tentokrát však již bez i-tého řádku - k tomuto příjemci již máme dárce určeného, a bez j-tého sloupce - tohoto dárce jsme již použili.

Od maxima

Další možností přiřazování je takzvaně od maxima, zdůrazněme však, že se nejedná o analogii přiřazování metodou od minima, pouze se záměnou minima za maximum. Tento algoritmus nejprve prohledává matici $\boldsymbol{V}$ způsobem jako v případě přiřazování nejbližšímu, tedy hledá minimum každého řádku. Z těchto nalezených minim pak určí maximum, označme jej v_{ij}^{max}, a příslušného i -tého

dárce s j-tým příjemcem, kteří měli tuto „maximální minimální“ vzdálenost, propojí. Platí

$$v_{ij}^{max} = \max\{ \min_{l \in \{1,\dots,n_d\}} v_{kl} \mid k \in \{1, \dots, n_p\}\}.$$

V dalším kroku pokračujeme stejně, ale opět již s vynecháním i-tého řádku a j-tého sloupce, neboť každý dárce se smí použít nejvýše jednou.

Náhodně

Tato metoda pracuje naprosto náhodně. Ke každému příjemci náhodně vybereme jednoho z dárců, přitom dodržujeme pouze jednu podmínku, a to, abychom každého dárce přiřadili nejvýše jednou. Pro tento typ fúze nebyly samozřejmě nutné všechny předchozí metody, ale stačilo ji aplikovat hned na původní data. Zařazujeme ji sem pouze z důvodu kontrastu s ostatními typy přiřazování.

Poznámka 11. Pokud je při algoritmech přiřazování dárců příjemcům výběr minima, respektive maxima nejednoznačný, tedy existuje-li více než jeden minimální, respektive maximální prvek, záleží na domluvě, který vezmeme.

Někdy je vhodné vybrat tento prvek náhodným výběrem. Tím se vyvarujeme toho, že bychom nějakou část dat upřednostňovali, aniž bychom si toho byli vědomi. V praxi jsou totiž data sesbírána například po krajích České republiky, takže jsou pak i v databázi zařazena u sebe. Mohlo by se tedy stát, že bychom jeden z krajů jakýmsi způsobem zvýhodnili. Z tohoto důvodu je často nejvhodnější všechny řádky databáze ještě před samotným průběhem metod fúze zpermutovat.

Z výše uvedených definic jednotlivých metod přiřazování a poznámky 1 vyplývá, že pro algoritmy přiřazování od minima, od maxima a náhodně je nutný předpoklad $n_d \geq n_p$, tedy že dárců je alespoň tolik co příjemců. Matice $\boldsymbol{V}$ je totiž typu $n_p \times n_d$ a kdyby tento předpoklad nebyl splněn, dostali bychom se do fáze, kdy jsou všichni dárci přiřazeni právě jednou a ještě zbývají příjemci, kteří nemají určeného dárce. Tyto tři algoritmy, ve kterých se smí každý dárce použít nejvýše jednou, by tedy skončily a fúze by byla neúplná. Naopak fúze metodou nejbližšímu tento předpoklad nevyžaduje.

Pro úplnost poznamenejme, že existují i takové metody přiřazování, které po spárování příjemce s dárcem tohoto konkrétního dárce nevyškrtnou ze seznamu všech dárců, které lze ještě použít, ale nějakým způsobem dárce pouze penalizují. A to například zvětšením vzdálenosti mezi ním a všemi ostatními příjemci, kteří ještě na přidělení čekají, což ho v dalších krocích výběru znevýhodní. Tím klesne pravděpodobnost, že bude tento dárce ještě někdy použit.

Příklad. Pro lepší pochopení uveďme konkrétní příklad nalezení dvojic příjemců a dárců v matici $\boldsymbol{V}$ čtyřmi výše definovanými způsoby. Pro jednoduchost zvolme

$\boldsymbol{V}$ jako čtvercovou matici typu 5×5 definovanou následovně

$$\boldsymbol{V} = \begin{pmatrix} 3 & 6 & 1 & 4 & 7 \\ 2 & 5 & 3 & 3 & 4 \\ 4 & 5 & 1 & 4 & 6 \\ 8 & 5 & 5 & 3 & 4 \\ 6 & 4 & 2 & 4 & 5 \end{pmatrix}.$$

Souřadnice zakroužkované hodnoty pak v následujících maticích představují souřadnice propojené dvojice příjemce – dárce v metodách nejbližšímu – $\boldsymbol{V_1}$, od minima – $\boldsymbol{V_2}$, od maxima – $\boldsymbol{V_3}$ a náhodně – $\boldsymbol{V_4}$. V případě náhodného přiřazení je uveden pouze jeden příklad ze všech možných kombinací.

$$\boldsymbol{V_1} = \begin{pmatrix} 3 & 6 & ① & 4 & 7 \\ ② & 5 & 3 & 3 & 4 \\ 4 & 5 & ① & 4 & 6 \\ 8 & 5 & 5 & ③ & 4 \\ 6 & 4 & ② & 4 & 5 \end{pmatrix} \quad \boldsymbol{V_2} = \begin{pmatrix} 3 & 6 & ① & 4 & 7 \\ ② & 5 & 3 & 3 & 4 \\ 4 & 5 & 1 & 4 & ⑥ \\ 8 & 5 & 5 & ③ & 4 \\ 6 & ④ & 2 & 4 & 5 \end{pmatrix}$$

$$\boldsymbol{V_3} = \begin{pmatrix} 3 & ⑥ & 1 & 4 & 7 \\ ② & 5 & 3 & 3 & 4 \\ 4 & 5 & 1 & 4 & ⑥ \\ 8 & 5 & 5 & ③ & 4 \\ 6 & 4 & ② & 4 & 5 \end{pmatrix} \quad \boldsymbol{V_4} = \begin{pmatrix} 3 & ⑥ & 1 & 4 & 7 \\ 2 & 5 & ③ & 3 & 4 \\ 4 & 5 & 1 & ④ & 6 \\ ⑧ & 5 & 5 & 3 & 4 \\ 6 & 4 & 2 & 4 & ⑤ \end{pmatrix}$$

Díky existenci více minimálních, respektive maximálních prvků (viz poznámka 11) mohou být matice $\boldsymbol{V_2}$ a $\boldsymbol{V_3}$ také tvaru

$$\boldsymbol{V_2'} = \begin{pmatrix} 3 & 6 & 1 & 4 & ⑦ \\ ② & 5 & 3 & 3 & 4 \\ 4 & 5 & ① & 4 & 6 \\ 8 & 5 & 5 & ③ & 4 \\ 6 & ④ & 2 & 4 & 5 \end{pmatrix} \quad \boldsymbol{V_3'} = \begin{pmatrix} 3 & ⑥ & 1 & 4 & 7 \\ 2 & 5 & 3 & 3 & ④ \\ ④ & 5 & 1 & 4 & 6 \\ 8 & 5 & 5 & ③ & 4 \\ 6 & 4 & ② & 4 & 5 \end{pmatrix}.$$

2.1.5 Dokončení fúze

Máme-li určené dvojice příjemce – dárce, zbývá už jen překopírovat hodnoty z řádku tabulky B_d příslušícímu příjemci do prázdného řádku nové tabulky B_p příslušícímu dárci. Tím je celá fúze dokončena a nastává fáze jejího hodnocení.

2.2 Report

Jakmile je samotná fúze hotová, chtěli bychom zjistit, jak byla úspěšná. Abychom však mohli fúzi vyhodnotit, potřebujeme k tomu znát původní skutečná data. Zde může nastat otázka, proč jsme tedy fúzi dělali, když původní skutečná data známe? Častým cílem algoritmů spojování dat je totiž jejich hodnocení, pro které je nutné znát původní data, abychom je mohli porovnat s daty nafúzovanými. Hodnocení algoritmů fúze přispívá k nalezení stabilních algoritmů pro určité kategorie dat, na jejichž základě by pak mohla být fúze prováděna v praxi, kde již skutečnou hodnotu získaných dat neznáme.

Jak již bylo zmíněno v sekci 1.3.4, často se pro vyhodnocení algoritmu fúze používá takzvaný „Split-sample foldover test“, při kterém si rozdělíme danou databázi na dárce a příjemce a určíme část dat, které příslušný algoritmus nafúzuje. V našem případě jsou to data, která jsme v sekci 1.2 označili jako tabulku B_p délky n_p (viz Obrázek 1.1). Na základě fúzí získaných hodnot této tabulky, označme je B_p^f , a znalostí hodnot skutečných, označme je B_p^s, se pak fúze vyhodnocuje pomocí nejrůznějších statistických ukazatelů. My jsme zvolili následující ukazatele. (Soong a de Montigny, 2003b)

2.2.1 Shoda marginálních rozdělení

Jedním ze základních znaků úspěšné fúze je shoda marginálního rozdělení každé nové proměnné získané fúzí s marginálním rozdělením k ní příslušné skutečné proměnné. Shoda marginálních rozdělení se určí pomocí jednoho z následujících dvou statistických testů v závislosti na typu proměnné, přičemž testujeme vždy hodnoty proměnné z tabulky B_p^f s hodnotami z tabulky B_p^s příslušné proměnné.

Kolmogorovův-Smirnovovův dvouvýběrový test

Tento dvouvýběrový test shodnosti dvou rozdělení uvedený v sekci 11.2.4 knihy Anděl (2005) využívá takzvanou empirickou distribuční funkci. Ta je pro náhodný výběr z rozdělení $X_1, \ldots, X_k$, které má distribuční funkci F, definována jako

$$\widehat{F}_k(x) = \frac{1}{k}\sum_{i=1}^{k} \mathbb{1}\{X_i < x\},\ x \in \mathbb{R}. \tag{2.7}$$

Nyní již přejděme k samotnému testu. Nechť $X_1, \ldots, X_k$ je náhodný výběr z rozdělení se spojitou distribuční funkcí F a nechť $Y_1, \ldots, Y_l$ je na něm nezávislý náhodný výběr z rozdělení se spojitou distribuční funkcí G. Aplikací na naše data představuje $X_1, \ldots, X_k$ hodnoty proměnné tabulky B_p^s a $Y_1, \ldots, Y_l$ představuje hodnoty této proměnné, avšak z tabulky B_p^f. V našem případě tedy platí $k = l = n_p$, kde n_p je délka tabulky B_p.

Poznámka 12. Předpoklad nezávislostí výběrů $X_1, \ldots, X_k$ a $Y_1, \ldots, Y_l$ pro naše data zaručit nemůžeme.

Chceme testovat nulovou hypotézu $H_0 : F = G$ proti alternativě $H_1 : F \neq G$ (více o testování hypotéz např. v sekci 8.1 knihy Anděl (2005)). Označme $\widehat{F}_k$ empirickou distribuční funkci prvního výběru, $\widehat{G}_l$ druhého výběru, a definujme

$$D_{k,l} = \sup_x |\widehat{F}_k(x) - \widehat{G}_l(x)| \tag{2.8}$$

(Anděl (2005), sekce 11.2.4)

Praktické vyhodnocení Kolmogorovova-Smirnovova testu je následující. Jsou-li čísla k a l malá, porovná se $D_{k,l}$ ze vzorce 2.8 s přesnými kritickými hodnotami $D_{k,l}(\alpha)$, kde α je zvolená hladina testu. Platí-li nerovnost $D_{k,l} \geq D_{k,l}(\alpha)$ potom zamítneme hypotézu H_0 na hladině α. V případě větších hodnot k, l se využije Smirnovovy věty (viz Anděl (2005), věta 11.11), spočítá se limitní distribuční funkce

$$K(x) = 1 - 2\sum_{n=1}^{\infty}(-1)^{k+1} exp\{-2k^2x^2\}$$

pro $x_0 = \sqrt{\frac{kl}{k+l}} D_{k,l}$. Pokud vyjde $K(x_0) \geq 1 - \alpha$, zamítne se hypotéza H_0 na hladině, která se s rostoucími rozsahy výběru blíží číslu α. (Anděl (2005), sekce 11.2.4)

Poznámka 13. Funkce $K(x)$ je běžně tabelovaná.

Tento test lze použít pouze za předpokladu, že jsou distribuční funkce F a G spojité. Proto jej lze použít pouze pro kardinální proměnné tabulky B_p, tedy pro ty, které jsme v sekci 5 nepřeváděli na soubor indikátorových proměnných.

Test homogenity multinomických rozdělení

Zbývající proměnné tabulky B_p jsou kvalitativní, nelze s nimi tedy provádět aritmetické operace. Jsme schopni určit pouze četnosti jejich hodnot. Na těch je založen test homogenity multinomických rozdělení uvedený v sekci 13.2 knihy Anděl (2005), který nyní použijeme.

Četnost zastoupení

Nejprve pro každou proměnnou určíme četnosti zastoupení jejích hodnot. Nechť náhodný výběr $X_1, \ldots, X_{n_p}$ reprezentuje proměnnou tabulky B_p, která nabývá hodnot $h_1, \ldots, h_r$. Pak četnost zastoupení hodnoty h_i, $i = 1, \ldots, r$, této proměnné spočteme jako

$$\vartheta_{h_i} = \sum_{j=1}^{n_p} \mathbb{1}\{X_j = h_i\}. \tag{2.9}$$

Vzorcem (2.9) spočítáme četnosti hodnot vybraných proměnných z tabulek B_p^s a B_p^f. Pomocí testu homogenity multinomických rozdělení budeme dále zkoumat, zda skutečné a k nim příslušné nafúzované hodnoty pocházejí ze stejného rozdělení.

Kontingenční tabulka

Definujme nyní obecnou kontingenční tabulku. Mějme náhodný vektor $(X,Y)^\top$, který má diskrétní rozdělení, a pro jednoduchost předpokládejme, že veličina X nabývá hodnot 1,...,r a veličina Y nabývá hodnot 1,...,c, $r \in \mathbb{N}$, $c \in \mathbb{N}$. (Pokud hodnota proměnné X nabývá znakových hodnot, položíme $X = 1$ půjde-li o první hodnotu, $X = 2$ půjde-li o druhou hodnotu, atd.) Označme

$$\delta_{ij} = \mathsf{P}(X = i, Y = j), \quad \delta_{i\cdot} = \sum_j \delta_{ij}, \quad \delta_{\cdot j} = \sum_i \delta_{ij}. \tag{2.10}$$

Číslům $\delta_{i\cdot}$ a $\delta_{\cdot j}$ se říká marginální pravděpodobnosti. Nechť se uskutečnil výběr o rozsahu k z tohoto rozdělení. Nechť μ_{ij} značí počet případů, kdy se ve výběru vyskytla dvojice (i,j). Náhodné veličiny μ_{ij} pak mají multinomické rozdělení s parametrem k a s pravděpodobnostmi δ_{ij} (více o multinomickém rozdělení např. v sekci 12.1 knihy Anděl (2005)). Analogicky jako v (2.10) označme

$$\mu_{i\cdot} = \sum_j \mu_{ij}, \quad \mu_{\cdot j} = \sum_i \mu_{ij}. \tag{2.11}$$

Matice (μ_{ij}) se nazývá kontingenční tabulka a je uvedena v Tabulce 2.2, číslům $\mu_{i\cdot}$ a $\mu_{\cdot j}$ se říká marginální četnosti. (Anděl (2005), sekce 13.1)

X	Y			$\sum$
	1	.....	c	
1	μ_{11}	.....	μ_{1c}	$\mu_{1\cdot}$
...	.....			...
r	μ_{r1}	.....	μ_{rc}	$\mu_{r\cdot}$
$\sum$	$\mu_{\cdot 1}$	.....	$\mu_{\cdot c}$	k

Tabulka 2.2: Obecná kontingenční tabulka $r \times c$

V našem případě bude veličina X nabývat právě dvou hodnot $X = 1$ a $X = 2$. Bude-li $X = 1$, pak náhodný výběr $Y_1, Y_2, \ldots, Y_k$ bude představovat hodnoty proměnné tabulky B_p^s. V opačném případě bude $Y_1, Y_2, \ldots, Y_k$ představovat hodnoty té samé proměnné, avšak z tabulky B_p^f. Tedy $k = n_p$ a vzniklá kontingenční tabulka je rozsahu $2 \times c$, kde c je počet hodnot, kterých nabývá příslušná proměnná tabulky B_p. V prvním, respektive druhém řádku kontingenční tabulky se tedy nachází četnosti skutečných, respektive fúzí získaných hodnot příslušné proměnné. Ty jsme již určili pomocí vzorce (2.9), tedy i marginální řádkové četnosti $\mu_{i\cdot}$ jsou předem určeny. Toho využívá test homogenity multinomických rozdělení.

Jsou-li totiž marginální řádkové četnosti $\mu_{i\cdot}$ předem stanoveny, pak má i-tý řádek kontingenční tabulky multinomické rozdělení s parametry $\mu_{i\cdot}$, $q_{i1}, \ldots, q_{ir}$, kde $q_{i1}, \ldots, q_{ir}$ jsou nějaké pravděpodobnosti splňující $q_{i1} + \cdots + q_{ir} = 1$, $i = 1, \ldots, r$. Chceme testovat nulovou hypotézu homogenity (stejnosti rozdělení), která říká, že pravděpodobnosti $q_{i1}, \ldots, q_{ir}$ nezávisejí na řádkovém indexu i, neboli všechny řádky matice $(q_{ij})_{r \times r}$ jsou stejné. Hypotézu testujeme pomocí veličiny

$$\chi^2 = n_p \sum_{i=1}^{r} \sum_{j=1}^{r} \frac{\mu_{ij}^2}{\mu_{i\cdot}\mu_{\cdot j}} - n_p,$$

která má asymptoticky rozdělení χ^2 s $(r-1)^2$ stupni volnosti. Vyjde-li pak $\chi^2 \geq \chi^2_{(r-1)^2}(\alpha)$, zamítáme hypotézu o homogenitě. Ke shodě s limitním rozdělením se vyžaduje, aby všechny teoretické četnosti $\frac{\mu_{i\cdot}\mu_{\cdot j}}{n_p}$ byly větší než 5, což je díky rozsahu databází v praxi téměř vždy splněno. (Anděl (2005), sekce 13.1 a 13.2)

Poznámka 14. Lze dokázat (viz Cramér (1946)), že testová statistika pro test homogenity je stejná jako pro test nezávislosti v kontingenční tabulce typu $r \times r$. Ten testuje nulovou hypotézu, která říká, že veličiny X a Y jsou nezávislé.

Vyhodnocení testů, „p-hodnota“

Při testování hypotéz se nezajímáme ani tak o to, zda zamítáme či nezamítáme nulovou hypotézu na té či oné hladině α, ale o takzvanou „p-hodnotu“, neboli dosaženou hladinu testu. Jedná se o nejmenší hladinu, při které bychom ještě hypotézu zamítli, a má pro nás tedy zásadní význam při vyhodnocení testů a tudíž i samotné fúze. Je-li p-hodnota větší než α, nezamítáme nulovou hypotézu a na

základě našich dat „lze tvrdit, že došlo ke shodě rozdělení“. Tedy větší p-hodnota znamená lepší fúzi.

Poznámka 15. Shoda marginálních rozdělení fúzí získané a skutečné proměnné je vlastnost, která by měla být pro náhodnou fúzi při dostatečném počtu pozorování vždy splněna. Při dostatečně velkém rozsahu pozorování se totiž tyto marginální rozdělení asymptoticky rovnají. To plyne v případě kardinálních proměnných z Glivenkovy věty (viz Anděl (2005), věta 11.10).

2.2.2 Zachování závislostí

Dalším významným ukazatelem úspěšnosti fúze je zachování vzájemných závislostí mezi proměnnými z dvou různých tabulek A_p a B_p. K tomu se často používají korelační koeficienty, které zkoumají závislosti mezi proměnnými. Všechny typy závislostí by měla dobrá fúze co nejvíce zachovat, korelační koeficienty by se tudíž měly co nejvíce shodovat. Jelikož jsou tyto koeficienty definovány za pomoci aritmetických operací, lze je opět vypočítat jen pro kardinální proměnné. Pro zbylé, tj. kvalitativní, proměnné použijeme pro zjištění zachování závislostí takzvaná adjustovaná rezidua.

Obecně nás bude zajímat zachování závislostí mezi proměnnými tabulky B_p a proměnnými tabulky A_p. Konkrétně pro kardinální proměnnou z tabulky A_p vypočteme korelaci jejích hodnot s hodnotami kardinální proměnné tabulky B_p^s. Byla-li fúze úspěšná, pak se tato korelace zachová, zaměníme-li tyto skutečné hodnoty za k nim příslušné hodnoty získané fúzí z příslušného sloupce tabulky B_p^f. Korelační koeficienty vypočítáme pro všechny kombinace proměnných tabulky A_p s proměnnými tabulky B_p^s, respektive B_p^f. Platí, že čím je shoda větší, tím lepší byla fúze.

Pro obecný výpočet korelačních koeficientů označme hodnoty proměnné tabulky A_p jako $X_1,X_2,\ldots,X_{n_p}$ a hodnoty proměnné tabulky B_p jako $Y_1,Y_2,\ldots,Y_{n_p}$.

Pearsonův korelační koeficient

Uvažujme náhodný výběr $(X_1,Y_1)^\top,\ldots,(X_{n_p},Y_{n_p})^\top$ z dvojrozměrného rozdělení. Pro náhodný výběr $X_1,\ldots,X_{n_p}$ definujme veličiny

$$\bar{X} = \frac{1}{n_p}\sum_{i=1}^{n_p} X_i, \quad S_X^2 = \frac{1}{n_p-1}\sum_{i=1}^{n_p}(X_i-\bar{X})^2,\ n_p \geq 2.$$

A analogicky pro náhodný výběr $Y_1,\ldots,Y_{n_p}$ definujme

$$\bar{Y} = \frac{1}{n_p}\sum_{i=1}^{n_p} Y_i, \quad S_Y^2 = \frac{1}{n_p-1}\sum_{i=1}^{n_p}(Y_i-\bar{Y})^2,\ n_p \geq 2.$$

Veličiny $\bar{X}$ a $\bar{Y}$ nazýváme výběrový průměr, veličiny S_X^2 a S_Y^2 nazýváme výběrový rozptyl. Dále označme

$$S_{XY} = \frac{1}{n_p-1}\sum_{i=1}^{n_p}(X_i-\bar{X})(Y_i-\bar{Y}).$$

Výběrový korelační koeficient (také se mu říká Pearsonův) pak při předpokladu $S_X^2 > 0$ a $S_Y^2 > 0$ definujeme vzorcem

$$r_{X,Y} = \frac{S_{XY}}{\sqrt{S_X^2 S_Y^2}}. \tag{2.12}$$

(Anděl (2005), sekce 6.1)

Hodnocení výběrového korelačního koeficientu je úzce vázáno na splnění předpokladu, že náhodný výběr $(X_1,Y_1)^\top,\ldots,(X_{n_p},Y_{n_p})^\top$ pochází z dvojrozměrného normálního rozdělení. Tento předpoklad ale v praxi velmi často zaručit nemůžeme. I přes tyto a další své nedostatky bývá však Pearsonův korelační koeficient často považován za nejdůležitější míru síly vztahu dvou proměnných. (Anděl (2005), sekce 6.1)

Spearmanův korelační koeficient

Na předpokladu výběru z dvojrozměrného normálního rozdělení nezávisí koeficient pořadové korelace, také se mu říká Spearmanův. Pro jeho odvození opět předpokládejme, že $(X_1,Y_1)^\top,\ldots,(X_{n_p},Y_{n_p})^\top$ je náhodný výběr z dvojrozměrného rozdělení. Označíme pořadí veličin $X_1,\ldots,X_{n_p}$ jako $R_1,\ldots,R_{n_p}$ a pořadí veličin $Y_1,\ldots,Y_{n_p}$ jako $Q_1,\ldots,Q_{n_p}$. Spearmanův korelační koeficient poté definujeme jako výběrový korelační koeficient počítaný z dvojic $(R_1,Q_1)^\top,\ldots,(R_{n_p},Q_{n_p})^\top$. Tedy dle (2.12)

$$r_S = \frac{\frac{1}{n_p-1}\sum_{i=1}^{n_p}(R_i-\bar{R})(Q_i-\bar{Q})}{\sqrt{(\frac{1}{n_p-1}\sum_{i=1}^{n_p}(R_i-\bar{R})^2)(\frac{1}{n_p-1}\sum_{i=1}^{n_p}(Q_i-\bar{Q})^2)}}. \tag{2.13}$$

Jelikož obecně platí

$$\sum_{i=1}^{n_p}(R_i-\bar{R})^2 = \sum_{i=1}^{n_p}R_i^2 - n_p\bar{R}^2 \text{ a } \sum_{i=1}^{n_p}(R_i-\bar{R})(Q_i-\bar{Q})) = \sum_{i=1}^{m}R_iQ_i - n_p\bar{R}\bar{Q},$$

lze vztah (2.13) upravit do tvaru

$$r_S = \frac{\sum_{i=1}^{n_p}R_iQ_i - n_p\bar{R}\bar{Q}}{\sqrt{(\sum_{i=1}^{m}R_i^2 - n_p\bar{R}^2)(\sum_{i=1}^{m}Q_i^2 - n_p\bar{Q}^2)}}. \tag{2.14}$$

Při použití Spearmanova korelačního koeficientu je třeba si uvědomit, že při přechodu z dat na jejich pořadí dochází vždy ke ztrátě informace. Na druhé straně je však velkou výhodou snížení citlivosti na odchylky od normality. (Anděl (2005), sekce 11.5)

Interpretace korelačních koeficientů

Ze Schwartzovy nerovnosti (viz Anděl (2005), věta 2.16) plyne, že oba korelační koeficienty definované vztahy (2.12) a (2.14) nabývají hodnot z uzavřeného intervalu $[-1,1]$. V případě kladné korelace se hodnoty obou proměnných zároveň zvětšují, naopak při záporné korelaci se hodnota jedné proměnné zvětšuje a druhá zmenšuje.

V našem případě nás však primárně nezajímá hodnota korelačního koeficientu pro určení síly závislosti. Jde nám pouze o její číselnou hodnotu za účelem porovnání a zjištění shody.

Adjustovaná rezidua

S kvalitativními proměnnými nelze provádět aritmetické operace a nejsme proto schopni pro ně korelační koeficienty určit. Pro zjištění zachování závislostí použijeme v tomto případě takzvaná adjustovaná standardizovaná rezidua. Jedná se o ukazatele, které se počítají pro každou buňku kontingenční tabulky a vyjadřují odchylku četností skutečných a fúzovaných hodnot proměnných této tabulky.

Uvažujme nyní obecnou kontingenční tabulku (μ_{ij}), $i = 1,\ldots,r$, $j = 1,\ldots,c$ definovanou v sekci 2.2.1 a uvedenou v Tabulce 2.2, založenou na výběru o rozsahu n_p. Pro každé $i = 1,\ldots,r$ a každé $j = 1,\ldots,c$ definujme veličiny $E_{ij} = \frac{\mu_{i\cdot}\mu_{\cdot j}}{n_p}$ a nazvěme je teoretické četnosti. Pomocí nich pak definujeme adjustovaná standardizovaná rezidua

$$e_{ij} = \frac{(\mu_{ij} - E_{ij})}{\sqrt{E_{ij}}}, \quad i = 1,\ldots,r, \quad j = 1,\ldots,c.$$

(Everitt, 1992).

Zachování závislostí budeme stejně jako v případě korelačních koeficientů ověřovat mezi kvalitativními proměnnými z tabulky A_p a kvalitativními proměnnými z tabulky B_p. Pro každou kombinaci (kvalitativní) proměnné tabulky A_p a (kvalitativní) proměnné tabulky B_p^s vytvoříme kontingenční tabulku. Pro tuto kontingenční tabulku spočteme adjustovaná rezidua, čímž nám vznikne tabulka reziduí, označme ji E^s, která je stejných rozměrů jako příslušná kontingenční tabulka. Následně zaměníme skutečné hodnoty proměnné tabulky B_p^s za fúzí získané hodnoty, které se nachází v příslušném sloupci tabulky B_p^f a vytvoříme novou kontingenční tabulku. Pro ni opět spočítáme adjustovaná rezidua, jejichž tabulku označíme E^f. Platí, že čím kvalitnější byla fúze, k tím přesnějším shodám hodnot ve stejných buňkách tabulek E_s a E_f by mělo docházet.

Poznámka 16. Pro naše účely postačí testování závislostí mezi kardinální proměnnou tabulky B_p a kardinální proměnnou tabulky A_p, případně mezi kvalitativní proměnnou tabulky B_p a kvalitativní proměnnou tabulky A_p. Nicméně závislosti mezi kardinální proměnnou tabulky B_p a kvalitativní proměnnou tabulky A_p, případně mezi kvalitativní proměnnou tabulky B_p a kardinální proměnnou tabulky A_p, jsou neméně důležité. Pro zjištění zachování takovýchto závislostí lze použít například analýzu rozptylu (viz Anděl (2005), kapitola 10).

2.2.3 Shoda individuálních proměnných

Dalším ukazatelem kvality fúze je shoda skutečných a fúzích získaných hodnot individuálních proměnných tabulky B_p, která vyjadřuje, jaká část skutečných hodnot proměnné se fúzí zachovala. Na rozdíl od předchozích ukazatelů, které s hodnotami dat pracovaly spíše souhrnně, se nyní zabýváme každou hodnotou individuálně. Pro většinu fúzí je tedy těžké takovéto shody dosáhnout.

Uvažujme opět kontingenční tabulku (μ_{ij}) definovanou v sekci 2.2.1. Veličina X bude nyní představovat skutečnou hodnotu proměnné tabulky B_p^s a veličina Y k ní příslušnou fúzí získanou hodnotu, která se nachází v příslušné buňce tabulky B_p^f. Předpokládáme, že proběhl náhodný výběr z těchto hodnot o rozsahu

n_p. Protože obě veličiny nabývají stejných hodnot $1,\ldots,r$, je nyní kontingenční tabulka (μ_{ij}) čtvercová typu $r \times r$, a je uvedena v Tabulce 2.3.

X	Y			$\sum$
	1	...	r	
1	μ_{11}	...	μ_{1r}	$\mu_{1\cdot}$
...	.....			...
r	μ_{r1}	...	μ_{rr}	$\mu_{r\cdot}$
$\sum$	$\mu_{\cdot 1}$	...	$\mu_{\cdot r}$	n_p

Tabulka 2.3: Kontingenční tabulka $r \times r$

Každý diagonální prvek μ_{ii} kontingenční tabulky vyjadřuje počet těch případů, kdy se ve výběru vyskytla dvojice (i,i), tj. kolikrát se skutečná hodnota příslušící i-tému sloupci kontingenční tabulky shodovala s příslušnou hodnotou získanou fúzí. Sečteme-li nyní všechny diagonální prvky a výsledek vydělíme n_p, dostaneme číslo vyjadřující, jaká část fúzovaných hodnot proměnné se shodovala s těmi skutečnými. V praxi tedy pro každou proměnnou tabulky B_p sestavíme kontingenční tabulku jejích skutečných a fúzí získaných hodnot, a spočítáme poměr

$$\frac{\sum_{i=1}^{r} \mu_{ii}}{n_p}, \tag{2.15}$$

kde r je počet hodnot, kterých příslušná proměnná nabývá. Je zřejmé, že čím více se poměr ze vzorce (2.15) blíží k jedné, tím lepší byla fúze.

2.2.4 Závěr

V praxi je dosažení celkové shody jak marginálních rozdělení, tak individuálních proměnných i korelačních koeficientů, případně adjustovaných reziduí velmi nepravděpodobné. Nelze prohlásit, že všechno je naprosto shodné a fúze je tudíž naprosto přesná. Ani naopak, že někde dochází k odlišnostem, a proto fúze přesná nebyla. Spíše je důležité najít na základě konkrétních dat možné praktické příčiny pro případné větší odlišnosti. (Soong a de Montigny, 2003b)

Tento teoretický popis hodnocení kvality fúze lze samozřejmě stejně jako samotný algoritmus fúze různě modifikovat, upravovat, popřípadě větvit. Ať už na základě našeho uvážení či struktury konkrétních dat.

Kapitola 3

Praxe

V kapitole 2 jsme si teoreticky dopodrobna probrali celý průběh fúze i s následným vyhodnocením. Nyní se budeme zabývat praktickým provedením fúze na konkrétních datech za pomoci statistických softwarových prostředků.

Pro naprogramování našich teoretických postupů využijeme statistický program R. Jedná se o jazyk a prostředí sloužící pro statistické a grafické zpracování dat, které poskytuje kromě základních matematických výpočtů také široký výběr různých statistických technik. K samotnému programu existuje také velké množství podpůrných balíčků, ve kterých je implementována řada pokročilých funkcí. Pro průběh našeho programu je nutná instalace balíčků *dummies* a *foreign*. (R-project)

Program R je volně dostupný a lze jej stáhnout z oficiálních stránek projektu (viz R-project), kde jsou k dispozici i podrobné manuály a online nápověda. Pro základní seznámení se s programem odkažme například na skripta Konečná a Koláček. Všechny příkazy a funkce programu jsou také velmi pěkně a podrobně popsány v jeho nápovědě.

3.1 Naprogramování teorie

3.1.1 Program Metody

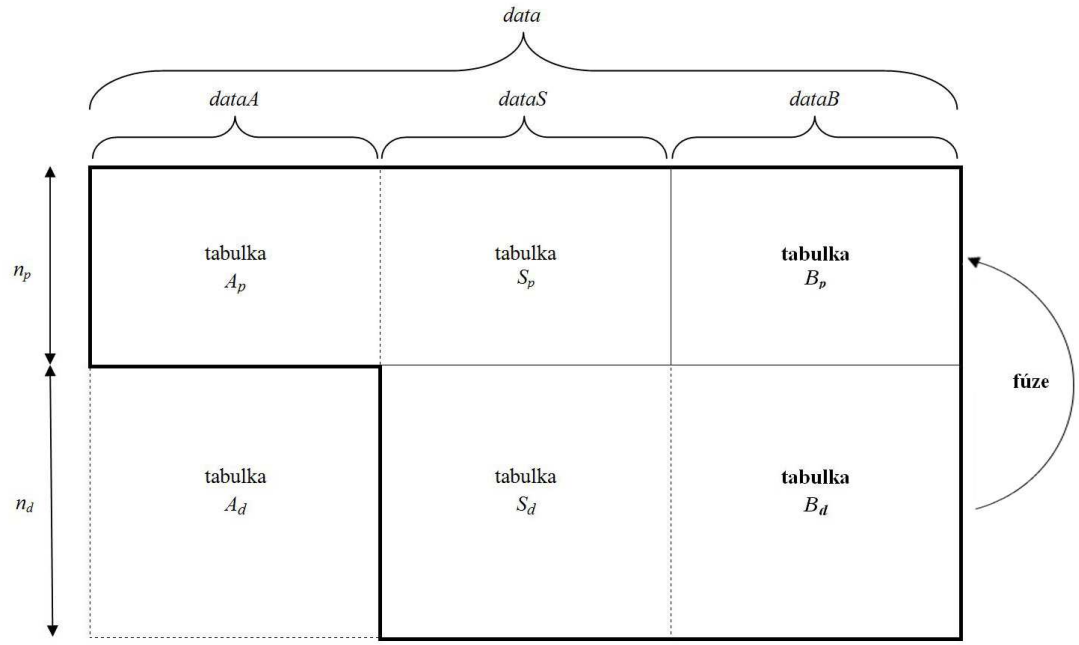
Nyní stručně popíšeme jeden ze způsobů, jak by naše teoretické metody z části 2.1 mohly být pomocí jazyka R naprogramovány. Jedná se o jednu konkrétní variantu, neboť některé problémy lze řešit různými způsoby pomocí více příkazů a funkcí, které se navíc dají mezi sebou různě kombinovat.

Uvedeme pouze základní příkazy a použité knihovny. Celý kód je pak uveden v Příloze 1 v dodatcích práce.

Načtení dat

Data ve formě datové tabulky, jejichž strukturou a generování se budeme zabývat níže v části 3.2, načteme do programu příkazem *load*. Jedná se o dvoudimenzionální strukturu, která se skládá z množiny proměnných - sloupce, a množiny pozorování - řádky. Takováto tabulka délky n_p+n_d, kterou jsme v kódu označili jako *data*, je jakýmsi sloučením tabulek A_p, S, B_p a B_d, zavedených v části 1.2 první kapitoly, a tabulky A_d, zavedené v poznámce 5, v pořadí uvedeném na Obrázku 3.1.

S využitím značení $\oplus_s$ a $\oplus_r$, zavedeného pro sloupcové a řádkové sjednocení tabulek (v tomto pořadí), lze proměnnou *data* ekvivalentně zapsat jako $(A_p \oplus_s A_d) \oplus_r S \ \oplus_r (B_p \oplus_s B_d)$. Tabulka $A_p \oplus_s A_d$ je pak v kódu programu označena jako *dataA*, analogicky tabulka $B_p \oplus_s B_d$ je označena jako *dataB* a tabulka S jako *dataS*. Příslušné podtabulky, jsou-li třeba použít, pak v kódu značíme přidáním indexu $\_p$ nebo $\_d$.



Obrázek 3.1: Vstupní data

Pro průběh programu se předpokládá, že prvky datové tabulky *data* jsou datového typu *numeric*, jde-li o číselné hodnoty, a datového typu *character*, jde-li o hodnoty znakové.

Poznamenejme ještě, že hodnoty tabulky B_p chceme fúzí získat. Je v nich proto prozatím uložena hodnota $< NA >$. Dále je v kódu programu ponecháno značení n_p, n_d, p_A, p_S i p_B pro délky a šířky příslušných tabulek. Součet $p\_A + p\_S + p\_B$ je pak uložen v proměnné p a proměnná *data* je tedy dimenze $(n\_p + n\_d) \times p$. Hodnoty n_p, n_d, p_A, p_S a p_B načítáme z vektoru v_dim, který byl vytvořen společně s daty.

Převod na soubor indikátorových proměnných

Pro převod kvalitativních proměnných na soubor indikátorových použijeme knihovnu *dummies*. Pro tyto účely je nutné si na základě typu každé proměnné dopředu připravit (viz sekce 3.2) vektor délky p, který je v kódu označen jako v_ind. Jeho i-tá složka obsahuje hodnotu 1, má-li být i -tá proměnná tabulky *data* převedena na soubor indikátorových, v opačném případě hodnotu 0. Neboli jeho i-tá složka obsahuje hodnotu 1, je-li i -tá proměnná tabulky *data* kvalitativní, a obsahuje 0, je-li i -tá proměnná tabulky *data* kardinální.

Poznámka 17. Jelikož při fúzi nepracujeme s tabulkou A_p, mohly by se zdát informace uložené v prvních p_A složkách vektoru v_ind jako nadbytečné. Tyto infor-

mace ale využijeme v následujícím reportování, kdy bude pro výpočet zachování závislostí nutné znát typy proměnných tabulky A_p.

Pro každou tabulku *dataS* a *dataB_d* pak postupně procházíme její sloupce a příslušné proměnné převádíme na soubor indikátorových pomocí funkce *dummy*. Tím se nám zvětší počet proměnných z *p_S*, respektive *p_B* na *ind_p_S*, respektive *ind_p_B*. Vzniklé tabulky šířek *ind_p_S* a *ind_p_B* jsou pak označeny jako *ind_dataS* a *ind_dataB_d*.

Faktorová analýza

Faktorovou analýzu provedeme pomocí funkce *prcomp*. Výstupem většiny výpočtů v programu R je datová struktura seznam, která se skládá z posloupností objektů zvaných složky, přičemž každá složka může obsahovat objekt jakéhokoliv typu. Stejně tak je to i u funkce *prcomp*. Z jejích výstupů nás nejvíce zajímají dvě konkrétní složky, a to zejména *$x*, ve které jsou uloženy hodnoty nových proměnných - faktorů. Ty jsou seřazeny sestupně od faktoru s největší důležitostí, neboli s největší standardní odchylkou. Tyto hodnoty odchylek jsou pak uloženy ve složce *$sdev* ve stejném pořadí, tedy od největší po nejmenší, a využíváme je při ukončovací podmínce faktorové analýzy. Určíme si hranici E (viz 2.1.2) a za nové proměnné vezmeme jen ty faktory, jejichž standardní odchylka je větší než tato hranice. Následnou normalizací faktorů dostáváme realizace nezávislých stejně rozdělených náhodných veličin s jednotkovými rozptyly.

Analýzu provedeme zvlášť pro tabulky *ind_dataS* a *ind_dataB_d*. Zaměněním jejich proměnných za příslušné počty faktorů vytvoříme nové tabulky fa_dataS a fa_dataB_d šířek fa_p_S, respektive fa_p_B, které jsou menší nebo rovny šířkám *ind_p_S*, respektive *ind_p_B*.

Manova

Váhy každé spojovací proměnné budeme postupně získávat pomocí *repeat* cyklu a ukládat do vektoru w délky fa_p_S. V každém kroku cyklu vytvoříme příslušný lineární regresní model tak, jak bylo podrobně popsáno v teoretické části 2.1.3, a aplikujeme na něj funkci *manova*. Váhu příslušné proměnné pak určíme z koeficientu R^2 odečtením hodnoty tohoto koeficientu z předchozího kroku. Cyklus se ukončí, pokud bude již všem spojovacím proměnným přidělena váha nebo pokud váha v proměnné v aktuálním kroku bude menší než zadaná hranice w_E.

Největší problém v této části je v získání samotného koeficientu determinace R^2 z výsledků funkce *manova*. My aplikujeme na náš model vzorec pro výpočet koeficientu R^2 odvozený v části 4.2.d knihy Timm (2002). Ten využívá výstupních složek funkce *manova $co*, *$x* a *$y* a funkcí *dim*, *t*, *solve* a $colMeans$. Jak jsme již zmínili v teoretické části, výsledkem je čtvercová matice. Abychom dostali skalár, přiřadíme koeficientu R^2 stopu této matice. Aby se hodnota koeficientu pohybovala mezi 0 a 1, vydělíme ji ještě počtem závisle proměnných v daném modelu, což je vyjádřeno počtem prvků na diagonále matice.

Matice vzdáleností

Pomocí vektoru w a tabulky hodnot spojovacích proměnných fa_dataS jsme nyní schopni podle vzorce (2.6) odvozeného v sekci 2.1.3 druhé kapitoly spočítat všechny prvky matice vzdáleností $\boldsymbol{V}$. Ta je dimenze $n\_p \times n\_d$ a každý její prvek $V[i,j]$ představuje vzdálenost i -tého příjemce od j-tého dárce.

Poznámka 18. Při odvozování vzorce (2.6) v části 2.1.3 jsme předpokládali, že výběrová kovarianční matice výběru vektorů hodnot spojovacích proměnných je díky tomu, že již pracujeme s faktory, jednotková. Pokud bychom ji však v programu příkazem $cov(fa\_dataS)$, respektive $cov(fa\_dataB\_d)$ spočítali, zjistíme, že diagonální prvky jsou opravdu jednotkové, avšak nediagonální prvky nabývají nenulových hodnot. Tyto hodnoty jsou však řádu 10^{-10} a menší a jejich nenulovou hodnotu lze přisuzovat zaokrouhlovacím chybám.

Přidělení dárců příjemcům

V tomto kroku vystačíme pouze s jednoduchými cykly a základními matematickými výpočty a funkcemi jako jsou *min* a *max*, *sample*, či *which*.

Dvojice příjemce - dárce určíme podle čtyř různých algoritmů popsaných v sekci 2.1.4 druhé kapitoly. Přiřadíme-li každému příjemci jeho nejbližšího dárce, vytvoříme přenesením hodnot dárce na hodnoty příjemce novou tabulku $data\_NEJ$. Analogicky v případě od minima dostaneme tabulku $data\_MIN$, v případě od maxima $data\_MAX$ a v případě náhodného přiřazení $data\_NAH$. Tím je celá fúze dokončena, přičemž nafúzovaná data se nacházejí vždy v podtabulce [1:n_p,(p_A+p_S+1):p] příslušné datové tabulky.

Pro grafické účely je ještě v této části definována matice H, pomocí níž se nám pak v grafickém výstupu programu zobrazí příkazem *hist* histogramy vzdáleností dvojic příjemce - dárce přiřazených k sobě příslušnou metodou.

Poznámka 19. Časová náročnost programu Metody závisí na rozsahu dat a na nastavení hranice E ve faktorové analýze, případně na nastavení hranice w_E při počítání vah spojovacích proměnných. Je také ovlivněna počtem proměnných, které se převádí na soubor indikátorových, na což je třeba přihlédnout při tvorbě vektoru v_ind ve smyslu poznámky 6.

Větší časová náročnost je v našem případě patrná při fúzi na datech společnosti Median. Předpokládáme-li nastavení hranic E, w_E a vektoru v_ind uvedené v kódu programu, pak se celková doba výpočtu na průměrném počítači pohybuje kolem osmi minut.

3.1.2 Program Report

Následný report fúze probíhá podle teoretické části 2.2 druhé kapitoly a jeho celý kód je uveden v Příloze 2 v dodatcích práce. V každém kroku je vždy potřeba vyhodnocovat fúzi čtyřikrát, neboť máme čtyři různé nafúzované databáze $data\_NEJ$, $data\_MIN$, $data\_MAX$ a $data\_NAH$.

S těmito databázemi načteme na začátku programu pro report navíc ještě vektory v_dim a v_ind a datovou tabulku $data\_komplet$, která na pozici fúzovaných hodnot obsahuje hodnoty skutečné, a bez které by tudíž report nebylo možno

provést. Ve všech datových tabulkách se v reportu zajímáme pouze o hodnoty nové tabulky vzniklé fúzí, tedy o hodnoty na pozicích $[1 : n\_p,]$.

Shoda marginálních rozdělení

Při zjišťování shody marginálních rozdělení skutečné a fúzí získané proměnné využijeme zavedeného vektoru *v_ind*. Je-li jeho hodnota $v\_ind[i]$ rovna 0, znamená to, že příslušná i-tá proměnná je kardinální. V takovémto případě pak použijeme pomocí příkazu *ks.test* Kolmogorovův-Smirnovův test. V opačném případě vytvoříme pomocí funkce *table* četnosti fúzovaných, respektive skutečných hodnot příslušné kvalitativní proměnné a jejich řádkovým spojením (funkce *rbind*) vytvoříme kontingenční tabulku. Na ní pak příkazem *chisq.test* provedeme test homogenity multinomických rozdělení.

Poznámka 20. Díky funkcím *levels* a $factor$ jsou v kontingenční tabulce zastoupeny i ty hodnoty proměnné, kterých proměnná sice aktuálně nenabývá, ale z definice nabývat může. Pro další práci s kontingenční tabulkou je totiž důležité, aby v ní byly všechny tyto hodnoty proměnné zahrnuty, byť je jejich četnost nulová. To platí i pro všechny další kontingenční tabulky v tomto kódu.

Při obou testech se zajímáme o p-hodnotu, která je uložena ve výstupní složce *$p.value*. Tyto hodnoty postupně ukládáme do vektorů $p.value\_NEJ$, respektive $p.value\_MIN$, $p.value\_MAX$ a $p.value\_NAH$. Ty pak tvoří složky výsledného seznamu *p.value*.

Zachování závislostí

Vektor *v_ind* využijeme i v případě volby mezi počítáním korelačních koeficientů či adjustovaných reziduí. Ty počítáme pro všechny kombinace skutečné, respektive fúzí získané proměnné s každou kardinální proměnnou tabulky A_p, která je v programu Report symbolizována datovou podtabulkou $data\_komplet[1 : n\_p,1 : p\_A]$. Jsou-li obě proměnné z této kombinace kardinální, spočítáme pro ně korelační koeficienty pomocí funkce *cor*. Zadáním argumentu *method* této funkce určíme, zda se bude počítat Spearmanův či Pearsonův korelační koeficient.

Tímto způsobem vypočteme korelaci každé kardinální proměnné tabulky A_p s každou kardinální proměnnou tabulky B_p, obsahující skutečné hodnoty. Její hodnoty poté zaměníme za k ní příslušné nafúzované a spočteme korelaci znovu. Tyto dvě korelace pak mezi sebou v absolutní hodnotě odečteme. Výsledné rozdíly postupně ukládáme do datových tabulek, které tvoří složky seznamu r a jejich řádky představují proměnné tabulky B_p a sloupce proměnné tabulky A_p. Pro proměnné, pro které se korelace nepočítala, je příslušný řádek (případně sloupec) vektor s hodnotami $< NA >$.

Zachování závislostí mezi kvalitativní proměnnou tabulky B_p a kvalitativní proměnnou tabulky A_p počítáme pomocí adjustovaných reziduí. Funkcí *table* vytvoříme kontingenční tabulku pro kombinaci proměnných tabulky B_p, obsahujících skutečné hodnoty, s proměnnými z tabulky A_p. Na kontingenční tabulku následně aplikujeme funkci *chisq.test*. Hodnoty adjustovaných reziduí jsou jedním z výsledků testu a jsou uloženy ve složce *$residuals* ve formě tabulky stejné dimenze jako původní kontingenční. Záměnou skutečných hodnot za nafúzované hodnoty příslušné proměnné v kombinaci za skutečnou spočítáme tyto hodnoty znova, přičemž nás opět zajímá absolutní hodnota jejich rozdílu. Ty po-

stupně tiskneme na obrazovku. Průměrné hodnoty adjustovaných reziduí jsou pak uloženy v konstantách $adj.rezidua\_NEJ$, $adj.rezidua\_MIN$, $adj.rezidua\_MAX$ a $adj.rezidua\_NAH$, které tvoří složky seznamu $adj.rezidua$.

Shoda individuálních proměnných

Pro výpočet poměru, udávajícího jaká část skutečných hodnot proměnné tabulky B_p se fúzí zachovala, vytvoříme opět příkazem *table* kontingenční tabulku. Tentokrát to bude čtvercová kontingenční tabulka skutečných hodnot proměnné a k ní příslušných fúzovaných hodnot. Poměr spočítáme vydělením součtu diagonálních prvků tabulky a hodnoty n_p a uložíme ho na příslušnou pozici do vektoru $shoda\_NEJ$, popřípadě $shoda\_MIN$, $shoda\_MAX$ a $shoda\_NAH$. Tyto vektory pak tvoří složky seznamu $shoda$.

Poznámka 21. Pro účely interpretace výsledků fúze na rozsáhlých databázích (ve smyslu velkého počtu proměnných) jsou v programu Report uvedeny příkazy *mean* pro průměr a *sd* pro směrodatnou odchylku, které shrnují výsledky příslušné metody u každého ukazatele. Toho využijeme v části 4.2 při reportování fúze provedené na datech společnosti Median.

Poznámka 22. U každého ukazatele lze také pomocí funkce *hist* vykreslit histogramy četností výsledných hodnot.

Poznámka 23. Větší časová náročnost programu Report se opět projeví na datech společnosti Median.

3.2 Vstupní data

Vstupní data do našeho programu pro fúzi si nejprve vytvoříme sami pomocí funkcí programu R, umožňujících vygenerovat náhodné výběry z různých rozdělení. Dále si fúzi vyzkoušíme také na datech sesbíraných v praxi, která nám poskytla firma Median.

3.2.1 Vygenerovaná data

Vygenerovaná data nám budou sloužit k hlubšímu pochopení fúze. Nebudou proto nijak rozsáhlá. Vygenerujeme je tak, aby byly na první pohled patrné všechny závislosti a vztahy mezi jednotlivými proměnnými a abychom díky nim mohli lehce vypozorovat a odůvodnit vzniklé odchylky ve výsledcích.

Definice dat

Vytvoříme data zabývající se čteností tištěných médií na jedné straně a stravováním na straně druhé. Protože se jedná pouze o ukázková data, postačí zvolit příslušné šířky $p\_A = 6$, $p\_S = 8$ a $p\_B = 6$ a délky $n\_p = 400$ a $n\_d = 600$. Datová tabulka $dataA$ obsahuje proměnné typu četnost čtení, předplatné či zda respondent čte deník Metro. Proměnné $dataB$ jsou typu nejčastější místo obědu, četnost vaření či zda respondent dodržuje pravidelnou stravu. Jako spojovací proměnné $dataS$ byly zvoleny klasické sociodemografické proměnné jako je věk, počet dětí, rodinný stav, či mzda. Není důležité vystihnout, aby se vygenerované hodnoty co

nejvíce blížily skutečnosti, spíše je podstatné vytvoření vzájemných vazeb mezi nimi.

Program Generování dat

Celý kód programu pro vygenerování je opět uveden v dodatcích práce v Příloze 3. Data jsou generována jednoduchými cykly a podmíněnými funkcemi, které jsou do sebe různě vnořeny za účelem vytvoření co nejvíce vazeb. Samotné generování pak probíhá pomocí funkce *sample* s využitím jejího argumentu *prob*, ve kterém zadáváme každé hodnotě z množiny, ze které generujeme, pravděpodobnost, že nastane. Vygenerované hodnoty proměnných následně uložíme do datové tabulky *data*.

Na základě definice každé proměnné určíme její typ, podle kterého zadáme příslušnou hodnotu vektoru *v_ind*. Ten následně uložíme příkazem *save*. Také uložíme vygenerovaná *data* jako datovou tabulku *data_komplet*, kterou využijeme při reportování. Zbývá už jen smazat data, která budeme chtít nafúzovat a to přepsáním hodnot podtabulky $data[1 : n\_p,(p\_A + p\_S + 1) : p]$ na hodnoty $< NA >$. Tím je tabulka *data* připravena k fúzi a příkazem *save* ji uložíme.

Poznámka 24. Díky tomu, že data jsou generována zcela náhodně, získáme po opětovném průběhu programem vždy odlišná data. Pro aplikaci fúze na těchto datech a následné vyhodnocení bychom však chtěli, aby se data vygenerovala sice zcela náhodně, ale vždy jednoznačně. To zajistí příkaz *set.seed*, který se nachází vždy před použitím funkce *sample*. Hodnota čísla v argumentu této funkce může být libovolná.

3.2.2 Data společnosti Median

Fúzi také aplikujeme na skutečných, tj. v praxi sesbíraných, datech, které nám poskytla firma Median (viz Median). Tato společnost se specializuje na kvalitativní i kvantitativní výzkumy trhu, sociologické výzkumy, výzkumy sledovanosti médií a výzkumy veřejného mínění.

Firma realizuje v České i Slovenské republice projekt „MML - Market & Media & Lifestyle“. Stejný výzkum se provádí v dalších zemích světa také pod zkratkou TGI, kterou jsme zmiňovali v úvodu v sekci 1.3.1. Databáze MML je vysoce akceptovaným zdrojem údajů v oblasti cíleného marketingu, nákupu médií a v oblasti reklamy. Projekt je kontinuálním výzkumem s výstupem dat čtyřikrát ročně. Nám byla poskytnuta část dat z 3.-4. čtvrtletí roku 2013. (Median)

Popis dat

Jedná se o data v rozsahu počtu 3000 pozorování. Po rozdělení dat do našich tabulek *dataA*, *dataS* a *dataB* dostáváme $p\_A = 269$ proměnných zaměřených na čtenost tištěných médií, $p\_S = 108$ sociodemografických proměnných a $p\_B = 178$ proměnných týkajících se využívání bankovních služeb. Konkrétně v tabulce *dataA* máme převážně proměnné typu zda respondent v určitém období četl určitý deník, týdeník, popřípadě měsíčník. Dále pak spojovací proměnné obsahují vedle klasických proměnných typu věk, pohlaví, vzdělání atd. i různé méně časté sociodemografické údaje například pohlaví hospodyně, či osobní vlastnosti respondenta. Tabulka dat, která budeme fúzovat, pak obsahuje informace typu

kým je respondentovi poskytován běžný účet, kolik zaplatí respondent v průměru kreditní kartou za měsíc, jakým způsobem splácí dluh na kreditní kartě či zda využívá pojištění domácnosti.

Program Nachystání MML dat

Sesbíraná data jsou uložena v souboru typu SAV. Je potřeba vytvořit kód, který tyto data do programu R načte a upraví tak, abychom na ně mohli následně aplikovat program pro fúzi. Tento kód je celý uveden v Příloze 4. Data načteme pomocí knihovny *foreign* a funkce *read.spss* do datové tabulky *data*, přičemž dle výchozího nastavení funkce budou všechny hodnoty proměnných datového typu *factor*. Toto nastavení prozatím ponecháme. Hranice mezi tabulkami *dataA*, *dataS* a *dataB* musíme určit ručně zadáním hodnot *p_A* a *p_S*. Dále je nutno ručně zadat hodnotu *n_p*, tedy kolik řádků budeme fúzovat. Hodnoty *n_p*, *n_d*, *p_A*, *p_S* a *p_B* vložíme do vektoru *v_dim*, který následně uložíme.

Jak bylo řečeno v poznámce 11, v praxi mohou být data sesbírána například po krajích České republiky a mohou pak být i v databázi zařazeny pod sebou. Proto je vhodné všechny řádky na začátku náhodně zpermutovat. K tomu využijeme funkce *sample.int*.

Dále je potřeba vytvořit již několikrát zmíněný vektor *v_ind*. To uděláme ručně na základě určení typu každé proměnné z datové tabulky. Tento vektor následně uložíme. Proměnné, které se nebudou převádět na indikátorové, jsou sice číselné, ale v tabulce *data* jsou uloženy jako datový typ *factor*. Proto je příkazem *as.numeric* převedeme na numerický datový typ.

V datech sesbíraných v praxi narážíme velmi často na jeden zásadní problém. Občas se v datech vyskytne hodnota, která nebyla pozorována, tedy kterou v našem případě respondent nezodpověděl. Takovéto buňky tabulky mají v sobě hodnotu $< NA >$ a nelze kvůli nim provést většinu metod použitých v celé fúzi. Pro naše účely postačí, pokud v případě kardinálních proměnných nahradíme chybějící buňky průměrem všech známých hodnot příslušné proměnné. U kvalitativních proměnných nahradíme jejich hodnotu znakem *nezodpovězeno*, čímž se nám z nich po převodu proměnné na soubor indikátorových proměnných stane vždy samostatná kategorie.

Poznámka 25. Problém chybějících dat v databázích je obecně známý pod názvem „Missing value problem“, neboli problém chybějících hodnot, a vyskytuje se v praxi velmi často. V literatuře lze najít různé metody a algoritmy doplňování chybějících dat, odkažme například na článek Pigott, případně na knihu Little a Rubin (2002).

Proměnné, které se budou převádět na soubor indikátorových proměnných, jsou stále ještě uloženy jako datový typ *factor*. Proto je příkazem *as.character* převedeme na znakový datový typ. Nyní již máme v tabulce *data* pouze hodnoty datového typu *numeric* nebo *character*. Tabulku *data* uložíme do datové tabulky *data_komplet*, abychom si ji uchovali pro reportování. Tu část dat, kterou budeme chtít nafúzovat, smažeme přepsáním hodnot podtabulky $data[1 : n\_p,(p\_A+p\_S+1) : p]$ na hodnoty $< NA >$. Následně tabulku *data*, která je připravena k fúzi, uložíme.

Kapitola 4

Výsledky a diskuse

V této kapitole se budeme zabývat výsledky programů pro fúzi a následný report (viz Přílohy 1 a 2) aplikovaných na námi vygenerovaných datech (viz 3.2.1) a datech poskytnutých společností Median (viz 3.2.2). Uvedeme si tabulky s konkrétními hodnotami ukazatelů definovaných v sekci 2.2 a pokusíme se odlišit a prodiskutovat jednotlivé typy fúzí, případně okomentovat vzniklé větší neshody.

Budeme se opět držet již zavedeného značení, přičemž nás pro reportování zajímají pouze hodnoty příjemců, tedy tabulky A_p, S_p a B_p, délek n_p.

Poznámka 26. Cílem této kapitoly není porovnání použitých čtyř metod přiřazování, tedy čtyř typů fúze, na konkrétních datech ve smyslu určit nejlepší, druhý nejlepší, třetí nejlepší a nejhorší typ. Takovéto porovnání by bylo značně individuální, neboť jeden ukazatel se může jevit významnější než druhý a naopak. V praxi lze tento problém vyřešit například tím, že na základě konkrétní situace a zadání fúze přiřadíme každému ukazateli váhu.

Stejný problém nastává u určování pořadí metod fúze u jednotlivých ukazatelů, neboť nemáme nástroj, který by nám výsledné hodnoty pro příslušný typ fúze shrnul do jednoho údaje. V praxi opět záleží na konkrétní situaci. Vhodným nástrojem může někdy být například medián, kvadratické odchylky výsledných hodnot, či průměr s přihlédnutím na směrodatnou odchylku. Při výběru je také nutné přihlédnout k faktorům jako je rozsah databáze či jak velké jsou výkyvy ve výsledcích.

Díky tomu, že jedním z našich typů fúze je metoda náhodného přiřazování dárců a příjemců, měli bychom ji být schopni ve výsledcích námi vybraných ukazatelů odlišit od ostatních typů a její výsledky by měly dosahovat obecně horších hodnot než výsledky ostatních metod. Výjimkou je shoda marginálních rozdělení, ke které by mělo docházet při dostatečně velkém počtu pozorování i při náhodné fúzi (viz poznámka 15 z části 2.2.1).

4.1 Výsledky fúze na vygenerovaných datech

Tato data jsme definovali v části 3.2.1 kapitoly 3. Aplikací programu Report (viz 3.1.2) na fúzi proběhlou na těchto datech dostáváme následující výsledky.

Poznámka 27. Výsledné hodnoty jednotlivých ukazatelů jsou zaokrouhleny na čtyři desetinná místa.

4.1.1 Shoda marginálních rozdělení

Shodu marginálních rozdělení skutečných a fúzí získaných hodnot (viz sekce 2.2.1) testujeme pro každou proměnnou tabulky B_p. V Tabulce 4.1 jsou uvedeny výsledné p-hodnoty Kolmogorovova-Smirnovova testu pro případ, že se jednalo o kardinální proměnnou - proměnné 2, 3, 4 a 5. U zbylých, tedy kvalitativních proměnných - proměnné 1 a 6, jsou uvedeny výsledky testu homogenity multinomických rozdělení.

proměnné tabulky B_p	**metoda přiřazování**			
	nejbližšímu	**od minima**	**od maxima**	**náhodně**
prom. 1	0.8830	0.9688	0.9958	0.4230
prom. 2	0.6399	1.0000	1.0000	0.9841
prom. 3	1.0000	1.0000	1.0000	0.5806
prom. 4	1.0000	0.9982	0.9996	1.0000
prom. 5	1.0000	1.0000	1.0000	0.9841
prom. 6	0.9999	0.9382	0.9639	0.5551

Tabulka 4.1: Výsledné p-hodnoty

Jak vidíme z Tabulky 4.1, výsledné p-hodnoty pro metody přiřazování nejbližšímu, od minima i od maxima jsou velmi pěkné. Metoda náhodné fúze se zde jeví jako nejhorší, neboť rozsah námi vygenerovaných dat není dostatečně velký.

4.1.2 Zachování závislostí

Zachování závislostí mezi proměnnými tabulky B_p a A_p ověřujeme pomocí korelačních koeficientů, případně adjustovaných reziduí (viz sekce 2.2.2).

Korelační koeficienty

Pro kardinální proměnné vypočteme korelaci hodnot každé proměnné tabulky A_p - proměnné 2, 3 a 4, s každou fúzí získanou hodnotou proměnné tabulky B_p - proměnné 2, 3, 4 a 5. Fúzí získané hodnoty poté zaměníme za k nim příslušné skutečné hodnoty a spočteme korelaci znovu. Tyto dvě korelace pak mezi sebou v absolutní hodnotě odečteme. Výsledné rozdíly jsou uvedeny v Tabulkách 4.2 - 4.5, a to jak pro Pearsonův korelační koeficient tak pro Spearmanův.

metoda přiřazování nejbližšímu		*metoda*	proměnné tabulky B_p			
			prom. 2	**prom. 3**	**prom. 4**	**prom. 5**
proměnné tabulky A_p	**prom. 2**	*Pearson*	0.0648	0.0158	0.0614	0.0618
		Spearman	0.0648	0.0158	0.0614	0.0784
	prom. 3	*Pearson*	0.0459	0.0522	0.0073	0.0002
		Spearman	0.0459	0.0522	0.0073	0.0055
	prom. 4	*Pearson*	0.0797	0.0591	0.0387	0.0360
		Spearman	0.0828	0.0466	0.0691	0.0414

Tabulka 4.2: Výsledné absolutní hodnoty rozdílu korelačních koeficientů - metoda přiřazování nejbližšímu

metoda přiřazování od minima		*metoda*	proměnné tabulky B_p			
			prom. 2	**prom. 3**	**prom. 4**	**prom. 5**
proměnné tabulky A_p	**prom. 2**	*Pearson*	0.0777	0.0543	0.0712	0.0434
		Spearman	0.0777	0.0543	0.0712	0.0581
	prom. 3	*Pearson*	0.0613	0.0958	0.0701	0.0016
		Spearman	0.0613	0.0958	0.0701	0.0001
	prom. 4	*Pearson*	0.0420	0.0610	0.0579	0.0624
		Spearman	0.0382	0.0459	0.0818	0.0235

Tabulka 4.3: Výsledné absolutní hodnoty rozdílu korelačních koeficientů - metoda přiřazování od minima

metoda přiřazování od maxima		*metoda*	proměnné tabulky B_p			
			prom. 2	**prom. 3**	**prom. 4**	**prom. 5**
proměnné tabulky A_p	**prom. 2**	*Pearson*	0.0440	0.0207	0.0599	0.0165
		Spearman	0.0440	0.0207	0.0599	0.0335
	prom. 3	*Pearson*	0.0613	0.0958	0.0723	0.0019
		Spearman	0.0612	0.0958	0.0723	0.0024
	prom. 4	*Pearson*	0.0555	0.0319	0.0112	0.0862
		Spearman	0.0459	0.0279	0.0015	0.0329

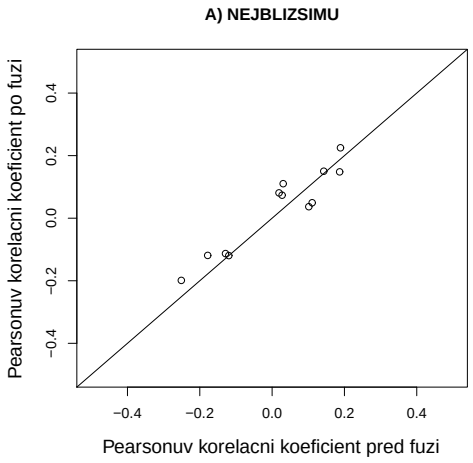
Tabulka 4.4: Výsledné absolutní hodnoty rozdílu korelačních koeficientů - metoda přiřazování od maxima

metoda náhodného přiřazování		*metoda*	proměnné tabulky B_p			
			prom. 2	**prom. 3**	**prom. 4**	**prom. 5**
proměnné tabulky A_p	**prom. 2**	*Pearson*	0.1052	0.1804	0.0694	0.0698
		Spearman	0.1052	0.1804	0.0694	0.0435
	prom. 3	*Pearson*	0.0311	0.2327	0.1736	0.1471
		Spearman	0.0311	0.2327	0.1736	0.2046
	prom. 4	*Pearson*	0.0321	0.1797	0.2217	0.2677
		Spearman	0.0432	0.0930	0.1635	0.1071

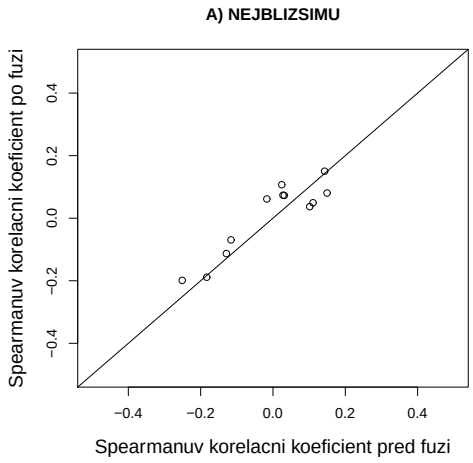
Tabulka 4.5: Výsledné absolutní hodnoty rozdílu korelačních koeficientů - metoda náhodného přiřazování

Jak vidíme z Tabulek 4.2 - 4.5, výsledné absolutní hodnoty rozdílů korelačních koeficientů jsou malé. Metody přiřazování nejbližšímu, od minima i od maxima lze tedy z hlediska zachování závislostí považovat za úspěšné. A to jak u Pearsonova korelačního koeficientu, tak i u Spearmanova koeficientu. O něco větších absolutních hodnot rozdílu korelačních koeficientů dosahovala dle očekávání fúze metodou náhodného přiřazování.

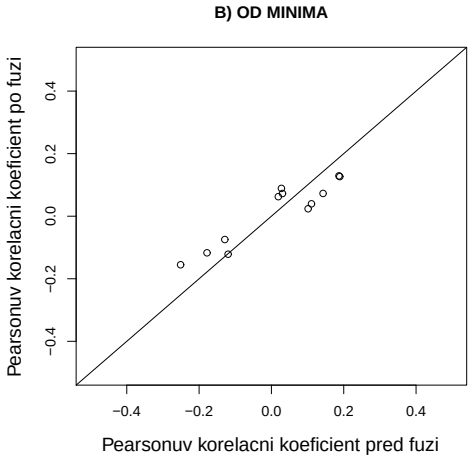
Na Obrázcích 4.1 - 4.8 jsou pro příslušné metody přiřazování graficky znázorněny korelační koeficienty před fúzí a po fúzi. Grafy jsou pro lepší představu doplněny osou prvního a třetího kvadrantu. Z obrázků je patrné, že nejvíce mimo osu jsou body v grafech příslušejících náhodné fúzi. Tím lze „na první pohled“ velmi pěkně rozpoznat neúspěšnost této metody oproti zbývajícím metodám.



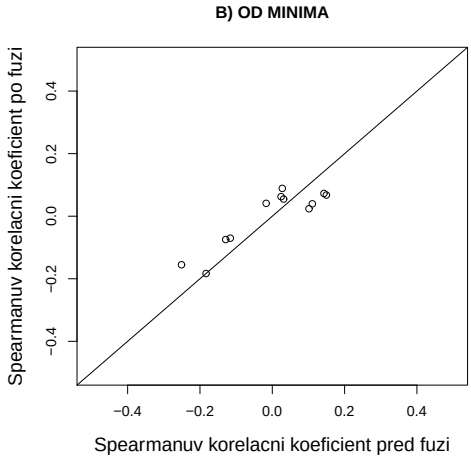
Obrázek 4.1: Graf závislostí Pearsonových korelačních koeficientů před fúzí a po fúzi - metoda přiřazování nejbližšímu



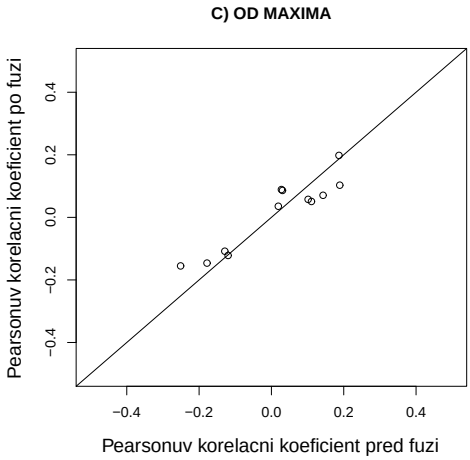
Obrázek 4.2: Graf závislostí Spearmanových korelačních koeficientů před fúzí a po fúzi - metoda přiřazování nejbližšímu



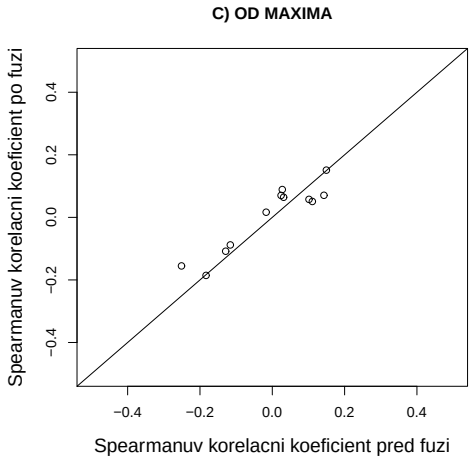
Obrázek 4.3: Graf závislostí Pearsonových korelačních koeficientů před fúzí a po fúzi - metoda přiřazování od minima



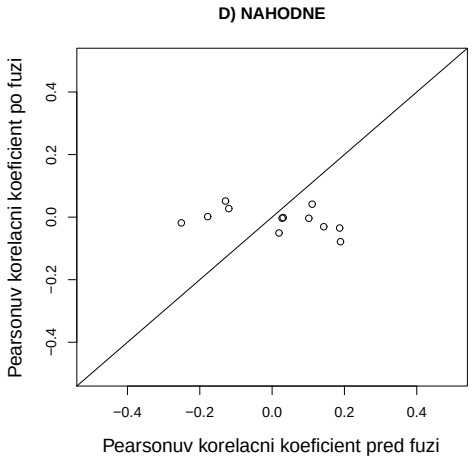
Obrázek 4.4: Graf závislostí Spearmanových korelačních koeficientů před fúzí a po fúzi - metoda přiřazování od minima



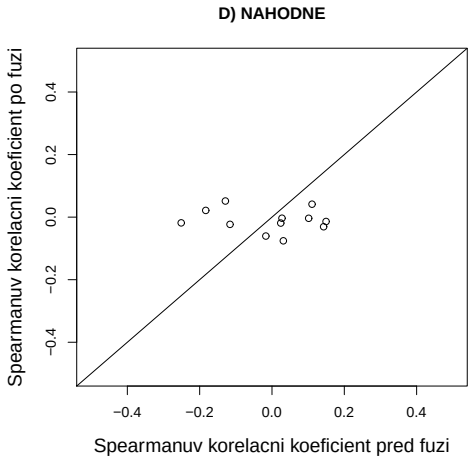
Obrázek 4.5: Graf závislostí Pearsonových korelačních koeficientů před fúzí a po fúzi - metoda přiřazování od maxima



Obrázek 4.6: Graf závislostí Spearmanových korelačních koeficientů před fúzí a po fúzi - metoda přiřazování od maxima



Obrázek 4.7: Graf závislostí Pearsonových korelačních koeficientů před fúzí a po fúzi - metoda náhodného přiřazování



Obrázek 4.8: Graf závislostí Spearmanových korelačních koeficientů před fúzí a po fúzi - metoda náhodného přiřazování

Adjustovaná rezidua

V případě kvalitativních proměnných počítáme absolutní hodnoty rozdílů adjustovaných reziduí v kontingenčních tabulkách, které vznikly kombinací každé proměnné z tabulky B_p - proměnné 1 a 6, s proměnnými z tabulky A_p - proměnné 1, 5 a 6. Výsledné hodnoty pro každou kombinaci jsou ve formě tabulky stejných rozměrů jako je příslušná kontingenční a jsou tedy rozsáhlejší. Proto se při interpretaci výsledků omezíme pouze na průměry hodnot jednotlivých tabulek výsledných absolutních hodnot adjustovaných reziduí, které jsou uvedeny v Tabulkách 4.6 - 4.9. Celé tabulky jsou pak vypisovány v programu Report (viz Příloha 2).

přiřazování metodou nejbližšímu		**proměnné tabulky B_p**	
		prom. 1	**prom. 6**
proměnné tabulky A_p	**prom. 1**	0.4330	0.2511
	prom. 5	0.4178	0.2224
	prom. 6	0.2879	0.2476

Tabulka 4.6: Průměry absolutních hodnot rozdílů adjustovaných reziduí - metoda přiřazování nejbližšímu

přiřazování metodou od minima		proměnné tabulky B_p	
		prom. 1	prom. 6
proměnné tabulky A_p	prom. 1	0.3221	0.4852
	prom. 5	0.5174	0.5054
	prom. 6	0.2983	0.5224

Tabulka 4.7: Průměry absolutních hodnot rozdílů adjustovaných reziduí - metoda přiřazování od minima

přiřazování metodou od maxima		proměnné tabulky B_p	
		prom. 1	prom. 6
proměnné tabulky A_p	prom. 1	0.4571	0.5650
	prom. 5	0.6833	0.4223
	prom. 6	0.3858	0.4204

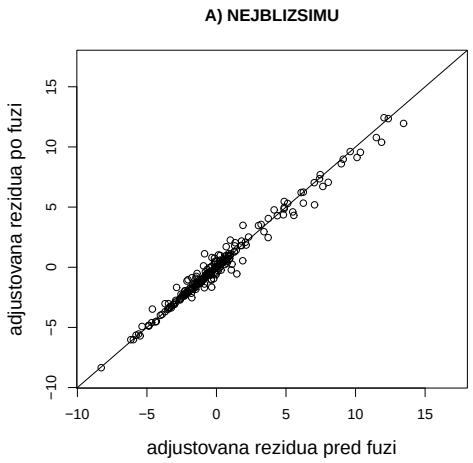
Tabulka 4.8: Průměry absolutních hodnot rozdílů adjustovaných reziduí - metoda přiřazování od maxima

metoda náhodného přiřazování		proměnné tabulky B_p	
		prom. 1	prom. 6
proměnné tabulky A_p	prom. 1	0.9350	1.0574
	prom. 5	4.4514	3.6258
	prom. 6	3.5438	3.3100

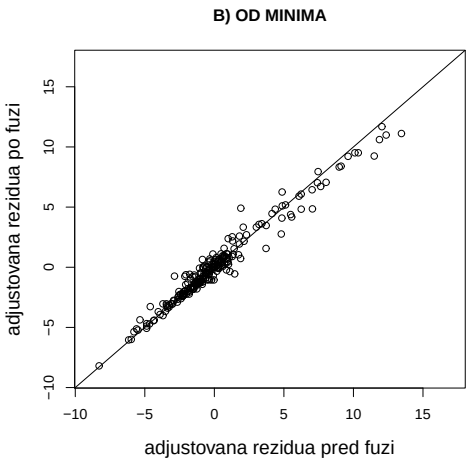
Tabulka 4.9: Průměry absolutních hodnot rozdílů adjustovaných reziduí- metoda náhodného přiřazování

Z Tabulek 4.6 - 4.9 vidíme, že průměry absolutních rozdílů hodnot adjustovaných reziduí vycházejí pro metody přiřazování nejbližšímu, od minima a od maxima relativně malé. Abychom však mohli tyto tři metody prohlásit za úspěšné, bylo by potřeba projít jednotlivé tabulky absolutních hodnot rozdílů adjustovaných reziduí kvůli případným větším výkyvům hodnot. Hodnoty průměrů v Tabulce 4.9 příslušící metodě náhodného přiřazování jsou dle očekávání viditelně větší než u ostatních metod.

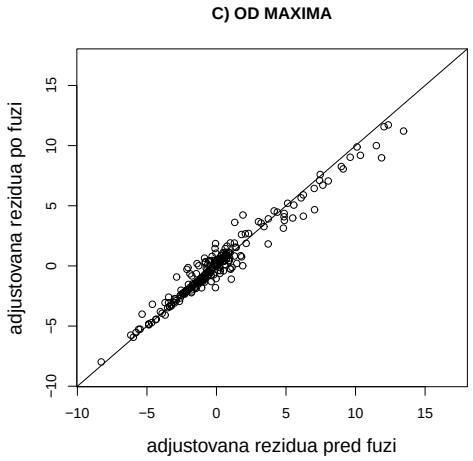
Na Obrázcích 4.9 - 4.12 jsou pro příslušné metody přiřazování graficky znázorněny závislosti adjustovaných reziduí před fúzí a po fúzi. V každém z grafů je navíc pro lepší představu vykreslena osa prvního a třetího kvadrantu. Stejně jako u korelačních koeficientů je i zde z obrázků „na první pohled“ velmi pěkně patrné, že nejvíce mimo osu jsou body v grafech příslušejících náhodné fúzi. To poukazuje na neúspěšnost této metody oproti zbývajícím metodám.



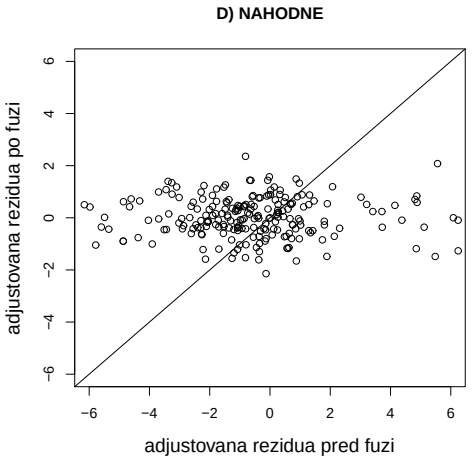
Obrázek 4.9: Graf závislostí adjustovaných reziduí před fúzí a po fúzi - metoda přiřazování nejbližšímu



Obrázek 4.10: Graf závislostí adjustovaných reziduí před fúzí a po fúzi - metoda přiřazování od minima



Obrázek 4.11: Graf závislostí adjustovaných reziduí před fúzí a po fúzi - metoda přiřazování od maxima



Obrázek 4.12: Graf závislostí adjustovaných reziduí před fúzí a po fúzi - metoda náhodného přiřazování

4.1.3 Shoda individuálních proměnných

Při počítání shody individuálních proměnných (viz sekce 2.2.3) nás zajímá poměr, udávající jaká část skutečných hodnot proměnné tabulky B_p se fúzí zachovala. Ten je uveden v Tabulce 4.10.

Jak již bylo zmíněno, na rozdíl od předchozích ukazatelů, které s hodnotami dat pracovaly spíše souhrnně, se nyní zabýváme každou hodnotou individuálně. Tudíž je obecně pro většinu fúzí těžké takovéto shody dosáhnout. Tento fakt je patrný z výsledků uvedených v Tabulce 4.10, kde se objevují větší odlišnosti i při přiřazování metodami nejbližšímu, od minima a od maxima. Přesto je zde

stále patrný rozdíl mezi těmito metodami a náhodnou fúzí, která dosahuje dle očekávání nejhorších výsledků.

proměnné tabulky B_p	**metoda přiřazování**			
	nejbližšímu	**od minima**	**od maxima**	**náhodně**
prom. 1	0.9250	0.9050	0.8750	0.2325
prom. 2	0.6075	0.5725	0.5325	0.5025
prom. 3	0.9575	0.9325	0.9475	0.5100
prom. 4	0.8175	0.8275	0.8200	0.7250
prom. 5	0.7625	0.7325	0.7200	0.2375
prom. 6	0.9450	0.9050	0.8675	0.2750

Tabulka 4.10: Výsledný poměr shody individuálních proměnných

4.1.4 Shrnutí výsledků

Jak již bylo řečeno v poznámce 26, porovnání jednotlivých metod může být individuální a závislé na konkrétní situaci a zadání fúze. Přesto se dle očekávání jeví jako nejhorší metoda náhodná fúze.

Při přiřazování metodami nejbližšímu, od minima a od maxima dosahovaly výsledky většiny ukazatelů velmi pěkných hodnot, které se blížily k ideálním hodnotám ukazatelů. Všechny tyto tři typy fúzí lze proto prohlásit za velmi úspěšné.

4.1.5 Doplňující diskuse

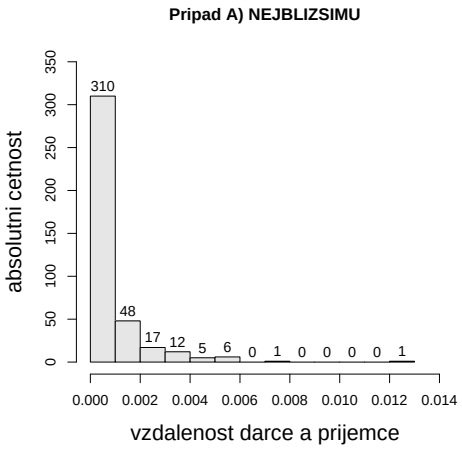
1. Při přiřazování metodou nejbližšímu nebyl na rozdíl od zbývajících metod přiřazování kladen požadavek na to, aby každý dárce byl přiřazen nejvýše jednou. Je zajímavé si všimnout četností počtu použití dárců při přiřazování příjemcům, které jsou uvedeny v Tabulce 4.11.

počet, kolikrát byl dárce použit	**1**	**2**	**3**	**4**	**5**	**6**
počet dárců s příslušnou četností	126	66	20	11	4	3

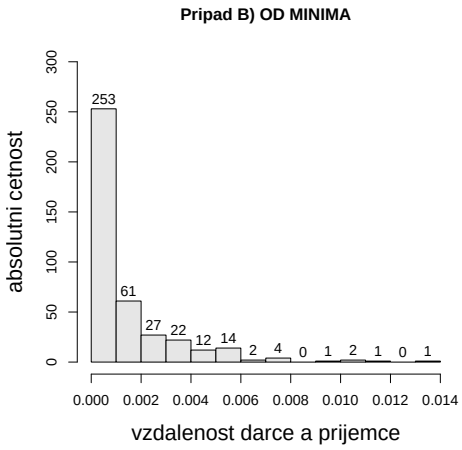
Tabulka 4.11: Četnosti počtu použití dárců - metoda nejbližšímu

Poznámka 28. Hodnoty tabulky získáme pomocí proměnné G z kódu Programu Metody (viz Příloha 1) příkazem $table(table(G[,1]))$.

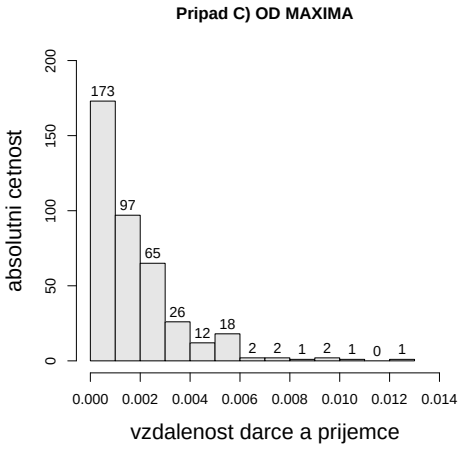
2. Zajímavé jsou také histogramy vzdáleností přidělených dárců a příjemců příslušnými metodami vykreslené na Obrázcích 4.13 - 4.16.



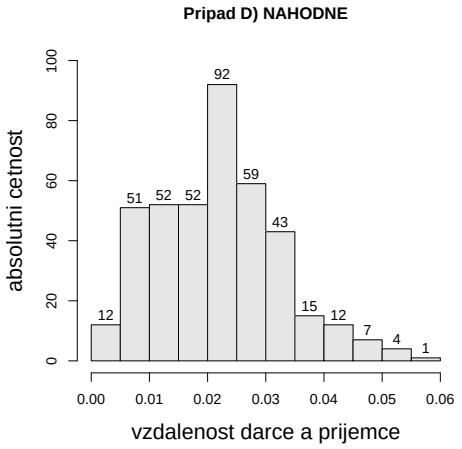
Obrázek 4.13: Histogram vzdáleností přidělených dárců a příjemců - metoda přiřazování nejbližšímu



Obrázek 4.14: Histogram vzdáleností přidělených dárců a příjemců - metoda přiřazování od minima



Obrázek 4.15: Histogram vzdáleností přidělených dárců a příjemců - metoda přiřazování od maxima



Obrázek 4.16: Histogram vzdáleností přidělených dárců a příjemců - metoda náhodného přiřazování

Poznámka 29. Vzdálenosti dárců a příjemců jsou myšleny ve smyslu definice matice vzdáleností - viz sekce 2.1.3.

3. Při testování shody individuálních proměnných jsme výsledné poměry uvedené v Tabulce 4.10 uvedli pro všechny proměnné. Mohlo by se stát, že bychom v datech měli proměnnou, které by nabývala velkého množství hodnot. Pak by poměr, vyjadřující shodu skutečných a fúzí získaných hodnot, byl oproti ostatním hodnotám pravděpodobně velmi malý. Počítat tento poměr by tedy pro takové proměnné nemělo smysl a bylo by vhodné je z tabulky odstranit, případně počítat poměr pouze pro kategoriální proměnné.

4.2 Výsledky fúze na reálných datech společnosti Median

Data, která nám poskytla společnost Median, jsou popsána v sekci 3.2.2. Aplikací programu Report (viz 3.1.2) na fúzi těchto dat dostáváme následující výsledky.

Jelikož se jedná o data obsahující vetší počet proměnných, není možné je kvůli velkému rozsahu výsledků jednotlivých ukazatelů interpretovat všechny, jako tomu bylo v předchozí části při reportování fúze na námi vygenerovaných datech. Je proto třeba shrnout výsledky příslušného ukazatele u jednotlivých typů fúzí. Jedním z nástrojů pro toto shrnutí je například průměr výsledných hodnot doplněný o jejich směrodatnou odchylku. Dalším možným nástrojem je například medián.

Poznámka 30. Interpretované hodnoty jsou zaokrouhleny na čtyři desetinná místa.

4.2.1 Shoda marginálních rozdělení

Shodu marginálních rozdělení skutečných a fúzí získaných hodnot (viz sekce 2.2.1) testujeme pro každou proměnnou tabulky B_p. Pro případ kardinálních proměnných jsou v Tabulce 4.12 uvedeny průměry výsledných p-hodnot Kolmogorovova-Smirnovova testu, které jsou doplněny směrodatnou odchylkou těchto hodnot. U zbylých, tedy kvalitativních proměnných, jsou uvedeny průměry a směrodatné odchylky výsledných p-hodnot testu homogenity multinomických rozdělení.

	metoda přiřazování			
	nejbližšímu	**od minima**	**od maxima**	**náhodně**
průměrné hodnoty p-hodnot	0.8558	0.8804	0.8676	0.8422
směr. odchylky p-hodnot	0.2920	0.2471	0.2649	0.2982

Tabulka 4.12: Interpretace výsledných p-hodnot

Jak bylo řečeno v poznámce 26 na začátku této kapitoly, jediným ukazatelem, při kterém by neměl být při dostatečně velkém počtu pozorování patrný rozdíl ve výsledcích mezi náhodnou fúzí a ostatními metodami, je shoda marginálních rozdělení. Z Tabulky 4.12 není skutečně patrný rozdíl mezi náhodným přiřazováním a ostatními metodami. Je však nutné zohlednit fakt, že jsou v tabulce uvedeny pouze průměry a směrodatné odchylky výsledků. Pro plné ověření neexistence rozdílu je nutné projít všechny jednotlivé výsledné p-hodnoty (vypsané programem Report).

4.2.2 Zachování závislostí

Zachování závislostí mezi proměnnými tabulky B_p a A_p ověřujeme pomocí korelačních koeficientů, případně adjustovaných reziduí (viz sekce 2.2.2). Jelikož ale v tabulce A_p nejsou žádné kvalitativní proměnné, adjustovaná rezidua nepočítáme a zachování závislostí ověřujeme pouze pomocí korelačních koeficientů.

Korelační koeficienty

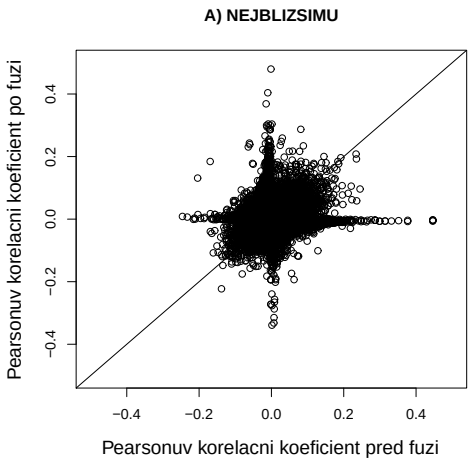
Pro kardinální proměnné vypočteme korelaci hodnot každé proměnné tabulky A_p s každou fúzí získanou hodnotou proměnné tabulky B_p. Fúzí získané hodnoty poté zaměníme za k nim příslušné skutečné hodnoty a spočteme korelaci znovu. Tyto dvě korelace pak mezi sebou v absolutní hodnotě odečteme. Výsledky jsou interpretovány pomocí průměru a směrodatné odchylky příslušných absolutních hodnot rozdílů korelačních koeficientů a jsou uvedeny v Tabulce 4.13, jak pro Pearsonův korelační koeficient tak pro Spearmanův koeficient.

		metoda přiřazování			
	metoda	**nej.**	**od min.**	**od max.**	**náh.**
průměrné hodnoty absolutních hodnot rozdílů korelačních koeficientů	*Pearson*	0.0290	0.0290	0.0286	0.0297
	Spearman	0.0290	0.0289	0.0286	0.0299
směr. odchylky absolutních hodnot rozdílů korelačních koeficientů	*Pearson*	0.0384	0.0414	0.0390	0.0391
	Spearman	0.0377	0.0400	0.0384	0.0390

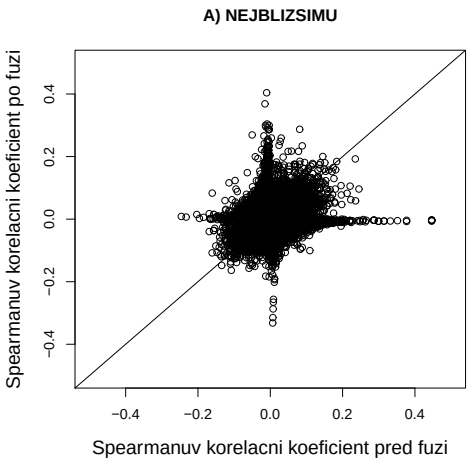
Tabulka 4.13: Interpretace výsledných absolutních hodnot rozdílů korelačních koeficientů

Jak vidíme z Tabulky 4.13, výsledné průměry a směrodatné odchylky absolutních hodnot rozdílů korelačních koeficientů jsou pro všechny čtyři typy fúzí velmi podobné. Kvůli velkému rozsahu databáze nemají průměry a směrodatné odchylky velkou informační hodnotu.

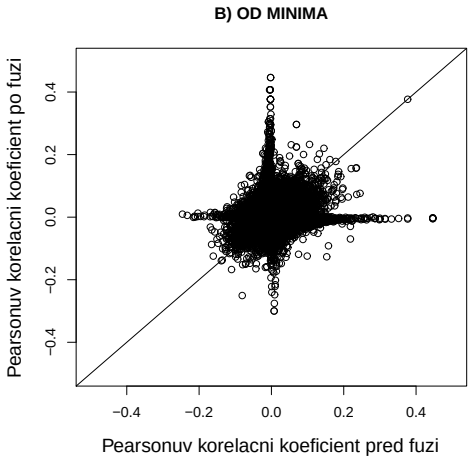
Mnohem lepší interpretací výsledků jsou v tomto případě grafy znázorňující korelační koeficienty před fúzí a po fúzi, které jsou uvedené na Obrázcích 4.1 - 4.8. Grafy jsou pro lepší představu doplněny osou prvního a třetího kvadrantu. Z obrázků je při hlubším zkoumání patrné, že při metodách přiřazování nejbližšímu, od minima a od maxima se body grafů seskupují ve tvaru elipsy kolem osy, zatímco při náhodném přiřazování vyplňují body grafu kružnici se středem v počátku. Tím lze rozpoznat neúspěšnost náhodného přiřazování oproti zbývajícím metodám.



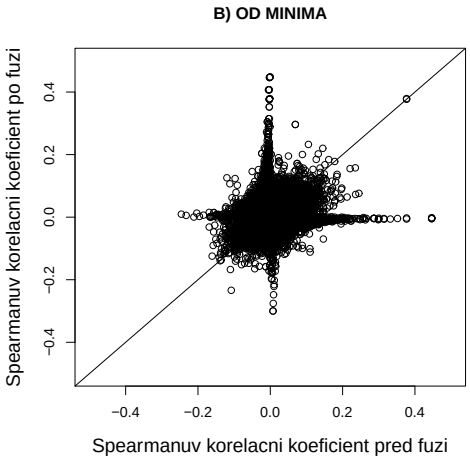
Obrázek 4.17: Graf závislostí Pearsonových korelačních koeficientů před fúzí a po fúzi - metoda přiřazování nejbližšímu



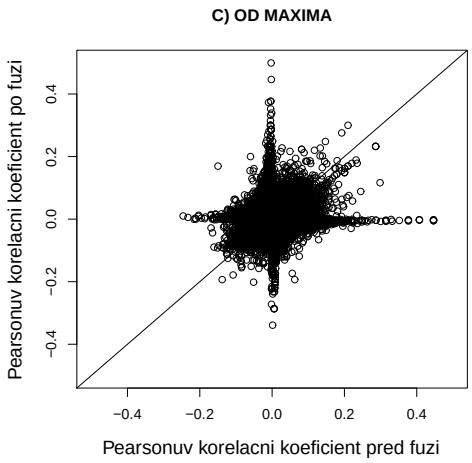
Obrázek 4.18: Graf závislostí Spearmanových korelačních koeficientů před fúzí a po fúzi - metoda přiřazování nejbližšímu



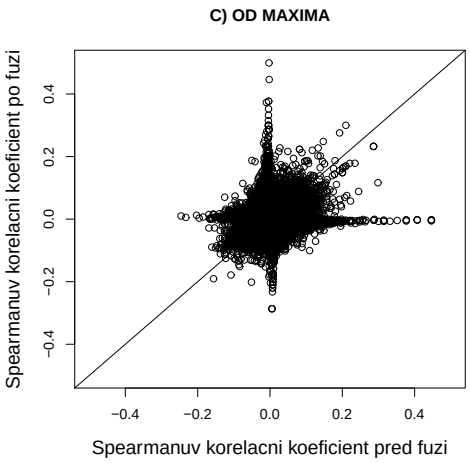
Obrázek 4.19: Graf závislostí Pearsonových korelačních koeficientů před fúzí a po fúzi - metoda přiřazování od minima



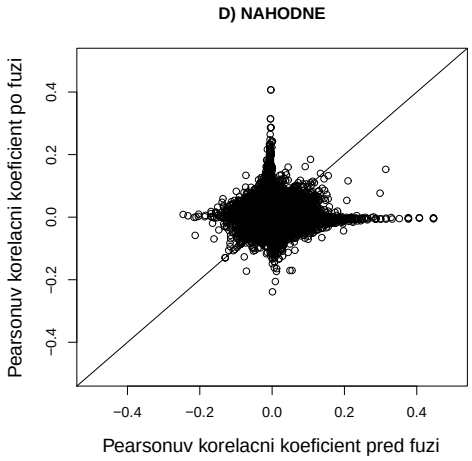
Obrázek 4.20: Graf závislostí Spearmanových korelačních koeficientů před fúzí a po fúzi - metoda přiřazování od minima



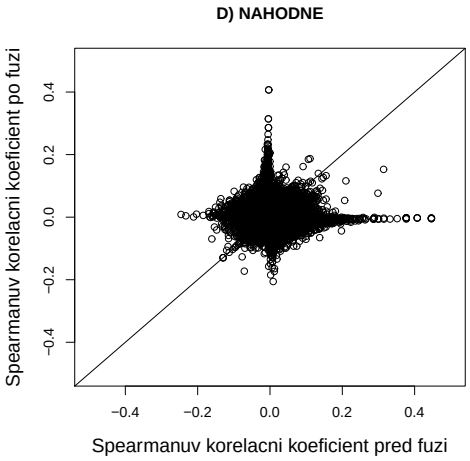
Obrázek 4.21: Graf závislostí Pearsonových korelačních koeficientů před fúzí a po fúzi - metoda přiřazování od maxima



Obrázek 4.22: Graf závislostí Spearmanových korelačních koeficientů před fúzí a po fúzi - metoda přiřazování od maxima



Obrázek 4.23: Graf závislostí Pearsonových korelačních koeficientů před fúzí a po fúzi - metoda náhodného přiřazování



Obrázek 4.24: Graf závislostí Spearmanových korelačních koeficientů před fúzí a po fúzi - metoda náhodného přiřazování

Další možností, jak prokázat rozdíly mezi jednotlivými metodami, je například použití lineárního regresního modelu (viz Zvára (2008)), který by vždy pro příslušnou metodu zkoumal závislost hodnot Pearsonových korelačních koeficientů před fúzí na hodnotách Pearsonových korelačních koeficientů po fúzi. Konkrétně nás zajímá testování nulové hypotézy, že koeficient regresní přímky je nulový, tedy že hodnoty korelačních koeficientů před fúzí nezávisí na hodnotách korelačních koeficientů po fúzi. Tento test bývá také často označován jako test o sklonu regresní přímky a jeho výsledky jsou shrnuty v Tabulce 4.14.

	metoda přiřazování			
	nejbližšímu	**od minima**	**od maxima**	**náhodně**
p-hodnota testu	$< 2.2 \cdot 10^{-16}$	$< 2.2 \cdot 10^{-16}$	$< 2.2 \cdot 10^{-16}$	0.258
hodnota odhadu koeficientu regresní přímky	0.1850	0.1403	0.1825	-0.0081
odhad chyby hodnoty odhadu koef. regresní přímky	0.0058	0.0054	0.0057	0.0072

Tabulka 4.14: Výsledky testu o sklonu regresní přímky

Z Tabulky 4.14 je již rozdíl mezi náhodným přiřazováním a ostatními metodami patrný. Vidíme, že p-hodnota testu je v případě náhodné fúze rovna 0.258, naproti tomu v ostatních případech přiřazování jsou p-hodnoty menší než $2.2 \cdot 10^{-16}$. Tedy zvolíme-li například hladinu testu klasicky $\alpha = 0.05$, nezamítáme v případě náhodného přiřazení nulovou hypotézu a naše data nejsou v rozporu s tvrzením, že regresní přímka má nulový sklon. Na základě našich dat tedy lze tvrdit, že při náhodném přiřazování nezávisí hodnoty korelačních koeficientů před fúzí na hodnotách korelačních koeficientů po fúzi. Naopak v případě přiřazování metodou nejbližšímu, od minima a od maxima zamítáme nulovou hypotézu na hladině $\alpha = 0.05$. Z hodnot odhadu koeficientu regresní přímky, které jsou doplněny odhadem chyby těchto hodnot, je rozdíl mezi náhodnou fúzí a ostatními metodami také dobře patrný.

Poznámka 31. V programu Report (viz Příloha 2) jsou na konci uvedeny kódy pro příslušné lineární modely, přičemž je využita funkce *summary* pro vypsání kompletních výsledků. P-hodnota testu se nachází vždy na posledním řádku výsledků pod parametrem $p - value$, hodnota odhadu koeficientu regresní přímky a odhad její chyby (v tomto pořadí) se nachází v části *Coefficients* v první a druhé buňce druhého řádku výsledné tabulky.

4.2.3 Shoda individuálních proměnných

Při počítání shody individuálních proměnných (viz sekce 2.2.3) nás zajímá poměr, udávající jaká část skutečných hodnot proměnné tabulky B_p se fúzí zachovala. Ten je interpretován pomocí průměru a směrodatné odchylky v Tabulce 4.15.

	metoda přiřazování			
	nejbližšímu	**od minima**	**od maxima**	**náhodně**
průměrné hodnoty shody indiv. prom.	0.8618	0.8606	0.8600	0.8356
směr. odchylky shody indiv. prom.	0.1740	0.1761	0.1783	0.2126

Tabulka 4.15: Interpretace výsledných shody individuálních proměnných

Z Tabulky 4.15 opět nejsou rozdíly mezi výslednými hodnotami nijak zvlášť patrné. Jelikož má náhodné přiřazování o něco větší směrodatnou odchylku, mohli bychom se domnívat, že se u něj vyskytuje větší množství výkyvů ve výsledcích. Pro ověření by však bylo nutné projít všechny jednotlivé výsledné shody individuálních proměnných (vypsané programem Report).

4.2.4 Shrnutí výsledků

Jak již bylo řečeno v poznámce 26, porovnání jednotlivých metod může být individuální a závislé na konkrétní situaci a zadání fúze. Přesto by se dle očekávání měla jevit jako nejhorší metoda náhodná fúze s výjimkou shody marginálních rozdělení. To je patrné z grafické interpretace korelačních koeficientů. Zásadní rozdíl mezi náhodnou fúzí a ostatními metodami také vyplynul při použití lineárního modelu zkoumajícího závislost hodnot Pearsonových korelačních koeficientů před fúzí na hodnotách Pearsonových korelačních koeficientů po fúzi.

Výsledky většiny ukazatelů při přiřazování metodami nejbližšímu, od minima a od maxima nedosahovaly zpravidla tak pěkných hodnot jako při reportování generovaných dat (viz sekce 4.1). Přesto lze všechny tyto tři metody přiřazování považovat za použitelné pro praxi.

Jednou z možností, jak zvýšit úspěšnost těchto fúzí, by mohlo být například snížení hranice pro standardní odchylky faktorů u faktorové analýzy (viz 2.1.2). Tím by se nám však podstatně zvýšila časová náročnost algoritmů zbývajících metod fúze (viz poznámka 19).

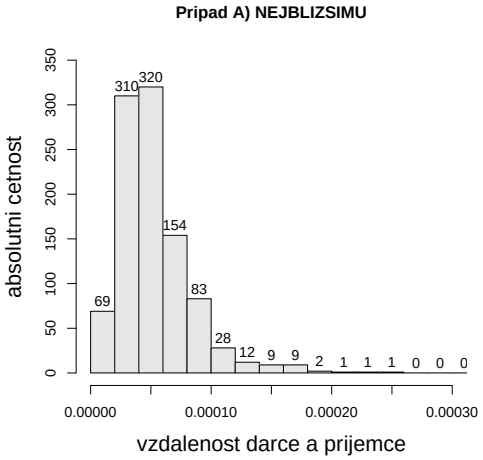
4.2.5 Doplňující diskuse

1. Při přiřazování metodou nejbližšímu nebyl na rozdíl od zbývajících metod přiřazování kladem požadavek, že každý dárce smí být přiřazen nejvýše jednou. Je zajímavé si všimnout četností počtu použití dárců při přiřazování příjemcům, které jsou uvedeny v Tabulce 4.16. (Dále viz poznámka 28.)

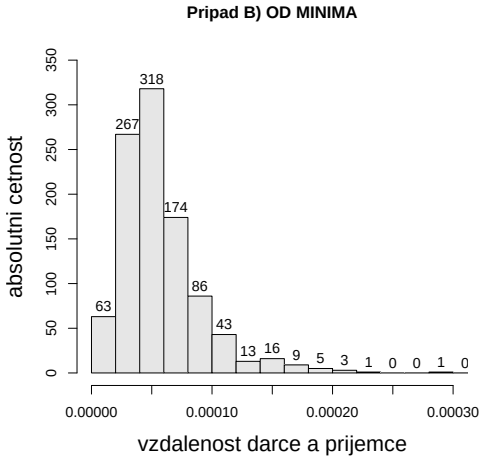
počet, kolikrát byl dárce použit	**1**	**2**	**3**	**4**	**5**
počet dárců s příslušnou četností	528	153	44	6	2

Tabulka 4.16: Četnosti počtu použití dárců - metoda nejbližšímu

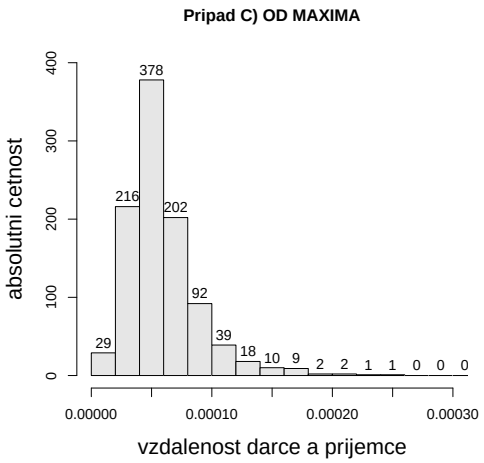
2. Zajímavé jsou také histogramy vzdáleností přidělených dárců a příjemců příslušnými metodami vykreslené na Obrázcích 4.25 - 4.28. (Dále viz poznámka 29.)



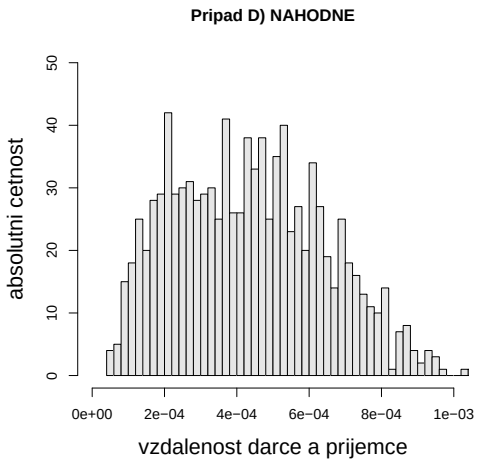
Obrázek 4.25: Histogram vzdáleností přidělených dárců a příjemců - metoda přiřazování nejbližšímu



Obrázek 4.26: Histogram vzdáleností přidělených dárců a příjemců - metoda přiřazování od minima



Obrázek 4.27: Histogram vzdáleností přidělených dárců a příjemců - metoda přiřazování od maxima



Obrázek 4.28: Histogram vzdáleností přidělených dárců a příjemců - metoda náhodného přiřazování

3. Při testování shody individuálních proměnných jsme výsledné poměry uvedené v Tabulce 4.15 uvedli pro všechny proměnné. Mohlo by se stát, že bychom v datech měli proměnnou, které by nabývala velkého množství hodnot. Pak by poměr, vyjadřující shodu skutečných a fúzí získaných hodnot, byl oproti ostatním hodnotám pravděpodobně velmi malý. Počítat tento poměr by tedy pro takové proměnné nemělo smysl a bylo by vhodné je z tabulky odstranit, případně počítat poměr pouze pro kategoriální proměnné.

4.3 Závěr

Provedli jsme čtyři různé typy statické fúze bez omezení na dvou různých databázích. Na námi vygenerovaných datech se dle očekávání jeví jako nejhorší metoda náhodná fúze. Při přiřazování ostatními metodami dosahovaly výsledky většiny ukazatelů velmi pěkných hodnot, které se blížily k ideálním hodnotám ukazatelů. Všechny tyto tři typy fúzí můžeme proto považovat za velmi úspěšné.

Na datech společnosti Median nebyly rozdíly mezi náhodnou fúzí a ostatními metodami jednoznačně patrné. Přesto se nám povedlo prokázat horší výsledky této fúze zejména díky interpretaci korelačních koeficientů. Náhodná fúze se tedy i na této databázi jeví dle očekávání jako nejhorší metoda. Výsledky většiny ukazatelů při přiřazování ostatními metodami nedosahovaly zpravidla tak pěkných hodnot jako při reportování generovaných dat. Přesto lze všechny tyto tři metody přiřazování považovat za použitelné pro praxi.

Kapitola 5

Závěr

5.1 Shrnutí práce

V této práci jsme se zabývali spojováním databází, jakožto jednou z cest řešící velmi častý problém dostupnosti dat v praxi. Zmínili jsme konkrétní poptávky po fúzi, zejména v oblasti marketingu, a základní algoritmy spojování dat. Tímto jsme narazili na problém zobecnění, který poukazuje na nemožnost určení nejlepšího algoritmu pro obecná data. Ve zbylých kapitolách práce jsme se pak zabývali takzvanou metodou „statistické fúze bez omezení“.

V teoretické části práce jsme si podrobně popsali metody průběhu statistické fúze bez omezení a následný report. Nejprve bylo nutné připravit si data, poté pomocí faktorové analýzy odstranit jejich vnitřní lineární závislosti. Na základě přiřazení váhy každé spojovací proměnné pomocí mnohorozměrné analýzy rozptylu se spočítala matice teoretických vzdáleností mezi příjemcem a dárcem. Pomocí této matice byly pak k sobě čtyřmi různými metodami přiřazování nazvanými nejbližšímu, od minima, od maxima a náhodně, přiřazeny dvojice příjemce - dárce, čímž byla fúze dokončena.

Následné vyhodnocení probíhalo na základě různých statistických ukazatelů. Nejprve jsme pomocí Kolmogorovova-Smirnovova testu a testu homogenity multinomických rozdělení testovali shodu marginálních rozdělení skutečných a fúzovaných hodnot, přičemž jsme se v každé kombinaci zajímali o p-hodnotu testu. Dalším ukazatelem bylo zachování vzájemných závislostí v databázi, které vyjadřují Pearsonův a Spearmanův korelační koeficient, popřípadě adjustovaná rezidua. Posledním ukazatelem pak byla shoda individuálních proměnných, vyjadřující jaká část skutečných hodnot fúzované proměnné se fúzí zachovala.

Praktický průběh fúze a následného vyhodnocení byl naprogramován pomocí statistického programu R. Pro fúzi jsme si sami zadefinovali a vygenerovali nepříliš rozsáhlá vlastní data tak, aby byly vidět vzájemné závislosti a mohli jsme na nich fúzi lépe pochopit. Navíc nám byla poskytnuta rozsáhlá data od společnosti Median, sesbíraná pro projekt „MML - Market & Media & Lifestyle“, díky kterým jsme mohli naprogramovanou fúzi aplikovat na skutečná data z praxe. Celé kódy příslušných programů jsou uvedeny v přílohách práce.

Výsledky fúze aplikované na obě databáze jsme uvedli formou grafů a tabulek hodnot příslušných ukazatelů z reportu, na jejichž základě jsme pak naše čtyři typy fúze diskutovali. Z výsledků je patrné, že nejhůře dopadla fúze metodou náhodného přiřazování, jak bylo očekáváno.

Seznam použité literatury

Anděl, J. (1985). *Matematická statistika*. Vydání druhé. SNTL - Nakladatelství technické literatury / ALFA, vydavateľstvo technickej a ekonomickej literatúry, Praha. ISBN 04-003-85.

Anděl, J. (2005). *Základy matematické statistiky*. Vydání první. Matfyzpress, Praha. ISBN 80-86732-40-1.

Anděl, J. (2007). *Statistické metody*. Čtvrté upravené vydání. Matfyzpress, Praha. ISBN 80-7378-003-8.

Cramér, H. (1946). *Mathematical Methods of Statistics*. Princeton University Press.

Everitt, B. S. (1992). *The Analysis of Contingency Tables*. Second edition. CRC Press, Florida. ISBN 0-412-39850-8.

Hebák, P., Hustopecký, J., Jarošová, E. a Pecáková, I. (2004). *Vícerozměrné statistické metody [1]*. Vydání první. Informatorium. ISBN 80-7333-025-3.

Hebák, P., Hustopecký, J. a Malá, I. (2005a). *Vícerozměrné statistické metody [2]*. Vydání první. Informatorium. ISBN 80-7333-036-9.

Hebák, P., Hustopecký, J., Průša, M., Řezanková, H., Svobodová, A. a Vlach, P. (2005b). *Vícerozměrné statistické metody [3]*. Vydání první. Informatorium. ISBN 80-7333-039-3.

Ingram, D. D., O'Hare, J., Scheuren, F. a Turek, J. (2000). Statistical Matching: A New Validation Case Study. In *Survey Research Methods Section (SRMS 2000), American Statistical Association*, pages 746–751. URL `http://www.amstat.org/sections/srms/Proceedings/y2000f.html`. [cit. červen 2014].

KantarMedia. TGI. [online]. URL `http://globaltgi.kantarmedia.com/home/` [cit. červen 2014].

Konečná, K. a Koláček, J. Jak pracovat s jazykem R. [online]. URL `http://www.nti.tul.cz/cz/images/7/7d/Navod_R_cesky.pdf`. [cit. červen 2014].

Little, R. J. A. a Rubin, D. B. (2002). *Statistical Analysis with Missing Data*. 2 edition. Wiley-Interscience. ISBN 978-0471183860.

Median. [online]. URL www.median.cz/. [cit. červen 2014].

Nielsen. TAM. [online]. URL http://www.agbnielsen.com/aboutus/whatistam.asp [cit. červen 2014].

Pigott, T. D. A Review of Methods for Missing Data. [online]. URL http://www.stat.uchicago.edu/~eichler/stat24600/Admin/MissingDataReview.pdf [cit. červen 2014].

R-project. [online]. URL http://www.r-project.org/. [cit. červen 2014].

Soong, R. a de Montigny, M. (2001). The anatomy of data fusion. In *10th Worldwide Readership Research Symposium (WRRS 2001)*, pages 87–109. URL http://www.zonalatina.com/WRRSfusion.pdf. [cit. červen 2014].

Soong, R. a de Montigny, M. (2003a). Does fusion on-the-fly really fly? In *Week of Audience Measurement (Mixed Media Session) 2003*, pages 183–204. URL http://www.magazine.org/does-fusion-fly-really-fly. [cit. červen 2014].

Soong, R. a de Montigny, M. (2003b). Foundations of split-sample foldover tests. In *11th Worldwide Readership Research Symposium (WRRS 2003)*, pages 453–466. URL http://www.zonalatina.com/WRRS2003split.pdf. [cit. červen 2014].

Soong, R. a de Montigny, M. (2004). No free lunch in data fusion / integration. In *Worldwide Audience Measurement WAM 2004*. URL http://www.zonalatina.com/WAM2004.pdf. [cit. červen 2014].

Timm, N. H. (2002). *Applied Multivariate Analysis*. Springer, New York. ISBN 978-0-387-95347-2.

van der Putten, P. (2000). Data Fusion: A Way to Provide More Data to Mine in? In *12th Belgian-Dutch Artificial Intelligence (BNAIC 2000)*. URL http://www.liacs.nl/~putten/library/fusiebnaic00.htm. [cit. červen 2014].

Zvára, K. (2008). *Regrese*. Vydání první. Matfyzpress, Praha. ISBN 978-80-7378-041-8.

Seznam obrázků

Seznam tabulek

Přílohy

Příloha 1: Program Metody

```
setwd(" ") #nutno zadat cestu do prislusneho adresare
rm(list=ls())

                                        ### METODY ###

## Nacteni dat a vektoru dimenzi:
# - znaceni: dim(A_p): n_p, p_A
#            dim(S): n_p+n_d, p_S
#            dim(B_d): n_d, p_B
#            => p_A+p_S+p_B=p

load("data.txt")
load("v_dim.txt")

p <- dim(data)[2]
n_p <- v_dim[1]; n_d <- v_dim[2]
p_A <- v_dim[3]; p_S <- v_dim[4]; p_B <- v_dim[5]

# - rozdeleni dat:
dataA_p <- data[(1:n_p),(1:p_A)]
dataS <- data[1:(n_p+n_d),(p_A+1):(p_A+p_S)]
dataB_d <- data[(n_p+1):(n_p+n_d),(p_A+p_S+1):p]

## Prevod danych promennych na soubor ind:
# - znaceni: dim(S): n_p+n_d, ind_p_S
#            dim(B_d): n_d, ind_p_B
ind_p_S <- p_S; ind_p_B <- p_B
ind_dataS <- dataS
ind_dataB_d <- dataB_d

library(dummies)
load("v_ind.txt")

# - prevod tabulky S:
k <- p_A
i <- 0
while (k!=p_A+p_S)
      {
      k <- k+1; i <- i+1
      if (v_ind[k]==1)
         {
         x <- dummy(ind_dataS[1:(n_p+n_d),i], data = NULL, sep = "=",
         drop = TRUE, fun = as.integer, verbose = FALSE)
         j <- dim(x)[2]
         if ((i>1)&(i<ind_p_S))
            {ind_dataS <- cbind(ind_dataS[1:(n_p+n_d),(1:(i-1))],x,
            ind_dataS[1:(n_p+n_d),((i+1):ind_p_S)])}
         if (i==1)
            {ind_dataS <- cbind(x,ind_dataS[1:(n_p+n_d),((i+1):ind_p_S)])}
         if (i==ind_p_S)
            {ind_dataS <- cbind(ind_dataS[1:(n_p+n_d),(1:(i-1))],x)}
         i <- (i-1+j); ind_p_S <- (ind_p_S-1+j)
         }
      }

# - prevod tabulky B_d:
#(pozn.: mame spravne k)
```

```
i<-0
while (k!=p)
      {
      k <- k+1; i <- i+1
      if (v_ind[k]==1)
         {
         x <- dummy(ind_dataB_d[1:n_d,i], data = NULL, sep = "=", drop = TRUE,
         fun = as.integer, verbose = FALSE)
         j <- dim(x)[2]
         if ((i>1)&(i<ind_p_B))
            {ind_dataB_d <- cbind(ind_dataB_d[1:n_d,(1:(i-1))],x,
            ind_dataB_d[1:n_d,((i+1):ind_p_B)])}
         if (i==1)
            {ind_dataB_d <- cbind(x,ind_dataB_d[1:n_d,((i+1):ind_p_B)])}
         if (i==ind_p_B)
            {ind_dataB_d <- cbind(ind_dataB_d[1:n_d,(1:(i-1))],x)}
         i <- i-1+j; ind_p_B <- (ind_p_B-1+j)
         }
      }

## Faktorova analyza:
# - znaceni: dim(S): n_p+n_d,fa_p_S
#            dim(B_d): n_d, fa_p_B

# - zadani hranice:
E <- 0.001          #vhodna hranice pro vygenerovana data
#E <- 1.6           #vhodna hranice pro MML data

# - tabulka S:
pc_S <- prcomp(ind_dataS, scale.=TRUE)
i <- 1; fa_p_S <- 0
while (pc_S$sdev[i]>E) {i<-i+1; fa_p_S<-fa_p_S+1}      #zjisteni poctu faktoru
fa_dataS <- pc_S$x[1:(n_p+n_d), 1:fa_p_S]

# - tabulka B_d:
pc_B <- prcomp(ind_dataB_d, scale.=TRUE)
i <- 1; fa_p_B <- 0
while (pc_B$sdev[i]>E) {i<-i+1; fa_p_B<-fa_p_B+1}      #zjisteni poctu faktoru
fa_dataB_d <- pc_B$x[1:n_d, 1:fa_p_B]

# - normalizace faktoru:
for (j in 1:fa_p_S) {fa_dataS[,j] <- (fa_dataS[,j])/sd((fa_dataS[,j]))}
for (j in 1:fa_p_B) {fa_dataB_d[,j] <- (fa_dataB_d[,j])/sd((fa_dataB_d[,j]))}

## Manova:
# - vektor vah w:
w <- vector(m="numeric", length=fa_p_S)          #vektor vah
P <- matrix(0,nr=n_d ,nc=1)                      #pouzite prom. co uz maji vahu
r <- vector(m="numeric", length=fa_p_S)          #vektor vsech r-squered
                                                 #v prislusnem kroku
pouzite <- vector(m="numeric", length=fa_p_S)    #pouzite[i]=0 if w[i] neurcena
                                                 #pouzite[i]=1 if w[i] urcena
r_predch <- 0                                    #r-squared pro predchozi krok
w_E <- 0                                  #hranice pro vahu (nejmensi mozna
                                          #vaha), lze ponechat pro obe databaze

# - nalezeni prvni vahy:
for (i in 1:fa_p_S)
      {
      model <- manova(as.matrix(fa_dataB_d)~fa_dataS[(n_p+1):(n_p+n_d),i],
               x=T,y=T)
      genR <- (t(model$co)%*%t(model$x)%*%model$y-dim(model$x)[1]
               *colMeans(model$y)%*%t(colMeans(model$y)))%*%solve(t(model$y)%*%
               model$y-dim(model$x)[1]*colMeans(model$y)%*%t(colMeans(model$y)))
      r[i] <- sum(diag(genR))/length(diag(genR))
      }
j <- which(r == max(r), arr.ind = TRUE)
w[j] <- r[j]-r_predch
r_predch <- r[j]
r <- vector(m="numeric", length=fa_p_S)
pouzite[j] <- 1
P <- fa_dataS[(n_p+1):(n_p+n_d),j]
```

```
# - nalezeni zbylych vah:
repeat
{
for (i in 1:fa_p_S)
       {
       if (pouzite[i]==0)
          {
          model <- manova(as.matrix(fa_dataB_d)^P+fa_dataS[(n_p+1):(n_p+n_d),i],
                   x=T,y=T)
          genR <- (t(model$co)%*%t(model$x)%*%model$y-dim(model$x)[1]*colMeans
                  (model$y)%*%t(colMeans(model$y)))%*%solve(t(model$y)%*%model$y
                   -dim(model$x)[1]*colMeans(model$y)%*%t(colMeans(model$y)))
          if (is.na(sum(diag(genR))/length(diag(genR)))==FALSE)
             {r[i] <-  sum(diag(genR))/length(diag(genR))}
          else {r[i] <- -1}
          }
       }
j <- which(r == max(r), arr.ind = TRUE)[1]
konec <- r[j]-r_predch
if (r[j]-r_predch >= w_E)
       {
       w[j] <- r[j]-r_predch
       r_predch <- r[j]
       r <- vector(m="numeric", length=fa_p_S)
       pouzite[j] <- 1
       P <- cbind(P,fa_dataS[(n_p+1):(n_p+n_d),j])
       }
if (min(pouzite)==1  |  konec < w_E) {break}
}

## Matice vzdalenosti V:
V <- matrix(0,nr=n_p ,nc=n_d )              #matice vzdalenosti
W_w <- matrix(0,nr=fa_p_S ,nc=fa_p_S )      #diagonalni matice druhych mocnin vah
diag(W_w) <- w^2
for (i in 1:n_p)
       {
       for (j in 1:n_d)
         {
         V[i,j] <- t(fa_dataS[i,1:fa_p_S]-fa_dataS[n_p+j,1:fa_p_S])%*%W_w%*%
                   (fa_dataS[i,1:fa_p_S]-fa_dataS[n_p+j,1:fa_p_S])
         }
       }

## Prirazeni darcu prijemcum:
G <- matrix (0,nr=n_p ,nc=4)  #Gij obsahuje ke kazdemu prijemci i jeho darce j,
                              #pro pripad A-prvni sloupec, B-druhy atd.
H <- matrix (0,nr=n_p ,nc=4)  #Hij obsahuje ke kazdemu prijemci vzdalenost jeho
                              #darce, pro pripad A-prvni sloupec, B-druhy atd.

# A) NEJBLIZSIMU:
data_NEJ <- data
for (i in 1:n_p)
       {
       j <- which(V[i,] == min(V[i,]), arr.ind = TRUE)[1]
       data_NEJ[i,(p_A+p_S+1):p] <- data_NEJ[(n_p+j),(p_A+p_S+1):p]
       G[i,1] <- j; H[i,1] <- V[i,j]
       }

# B) OD MINIMA:
V_MIN <- V; data_MIN <- data
for (i in 1:n_p)
       {
       k <- which(V_MIN == min(V_MIN), arr.ind = TRUE)[1,1]
       j <- which(V_MIN == min(V_MIN), arr.ind = TRUE)[1,2]
       data_MIN[k,(p_A+p_S+1):p] <- data_MIN[(n_p+j),(p_A+p_S+1):p]
       G[k,2] <- j; H[k,2] <- V[k,j]
       V_MIN[k,1:n_d] <- max(V)+1
       V_MIN[1:n_p,j] <- max(V)+1
       }

# C) OD MAXIMA:
data_MAX <- data; V_MAX <- V
for (a in 1:n_p)
```

```
      {
      max <- 0
      for (i in 1:n_p)
         {
         if (is.na(min(V_MAX[i,]))==F)
            {
            min <- which(V_MAX[i,]==min(V_MAX[i,]),arr.ind = TRUE)[1]
            #min ... sloupec ve kterem je min i-teho radku
            if (V_MAX[i,min]>=max) {max <- V_MAX[i,min]; k <- i; j <- min}
            }
         }
      data_MAX[k,(p_A+p_S+1):p] <- data_MAX[(n_p+j),(p_A+p_S+1):p]
      H[k,3] <- V[k,j]; G[k,3] <-j
      V_MAX[1:n_p,j] <- max(V)+1
      V_MAX[k,j] <- NA
      }

# D) NAHODNE:
data_NAH <- data
set.seed(100)
s <- sample(1:n_d, n_p, replace=FALSE)
for (i in 1:n_p)
      {
      data_NAH[i,(p_A+p_S+1):p] <- data_NAH[(n_p+s[i]),(p_A+p_S+1):p]
      G[i,4] <- s[i]; H[i,4] <- V[i,s[i]]
      }

# - histogramy vzdalenosti pridelenych darcu a prijemcu:
split.screen(c(1,4))
# A )
screen(1)
hist(H[,1], xlab = "vzdalenost darce a prijemce", ylab = "absolutni cetnost",
main = "Pripad A) NEJBLIZSIMU", xlim=c(0,0.014), ylim=c(0,350), col=gray(0.9))

# B)
screen(2)
hist(H[,2], xlab = "vzdalenost darce a prijemce", ylab = "absolutni cetnost",
main = "Pripad B) OD MINIMA", xlim=c(0,0.014), ylim=c(0,300), col=gray(0.9))

# C)
screen(3)
hist(H[,3], xlab = "vzdalenost darce a prijemce", ylab = "absolutni cetnost",
main = "Pripad C) OD MAXIMA", xlim=c(0,0.014), ylim=c(0,200), col=gray(0.9))

# D)
screen(4)
hist(H[,4], xlab = "vzdalenost darce a prijemce", ylab = "absolutni cetnost",
main = "Pripad D) NAHODNE", xlim=c(0,0.06), ylim=c(0,100), col=gray(0.9))

# - cetnosti pouziti darcu pri prirazovani prijemcum metodou nejblizsimu:
# pozn.: pri ostatnich metodach je kazdy darce pouzit max. jednou
table(table(G[,1]))

## Ulozeni nafuzovanych dat (pro report):
save(data_NEJ, file="data_NEJ.txt"); save(data_MIN, file="data_MIN.txt")
save(data_MAX, file="data_MAX.txt"); save(data_NAH, file="data_NAH.txt")
```

Příloha 2: Program Report

```
setwd(" ") #nutno zadat cestu do prislusneho adresare
rm(list=ls())

                                ### REPORT ###

## Nacteni dat a vektoru ind a vektoru dimenzi:
# - znaceni: dim(A_p): n_p, p_A
#            dim(S): n_p+n_d,p_S
#            dim(B_d): n_d, p_B
#            => p_A+p_S+p_B=p
load("data_komplet.txt")
load("data_NEJ.txt"); load("data_MIN.txt")
load("data_MAX.txt"); load("data_NAH.txt")
load("v_ind.txt")
```

```
p <- dim(data_komplet)[2]
load("v_dim.txt")
n_p <- v_dim[1]; n_d <- v_dim[2]
p_A <- v_dim[3]; p_S <- v_dim[4]; p_B <- v_dim[5]

## Shoda marginalnich rozdeleni:
# - Kolmogorovuv-Smirnovovuv test / test homogenity multinomickych rozdeleni:

# A) NEJBLIZSIMU:
p.value_NEJ <- vector(m="numeric", length=p_B)
for (i in 1:p_B)
      {
      if (v_ind[p_A+p_S+i]==1)
                    # nemusi jit nutne o znak. promennou (muze byt ord. cis.)
         {
         l <- union(labels(table(data_komplet[1:n_p,p_A+p_S+i]))[[1]],
                    labels(table(data_NEJ[1:n_p,p_A+p_S+i]))[[1]])
         kt <- rbind(table(factor(data_komplet[1:n_p,p_A+p_S+i],levels=l)),
                     table(factor(data_NEJ[1:n_p,p_A+p_S+i],levels=l)))
         p.value_NEJ[i] <- chisq.test(kt)$p.value
         }
      else {p.value_NEJ[i] <- ks.test(data_komplet[1:n_p,p_A+p_S+i],data_NEJ
           [1:n_p,p_A+p_S+i], alternative = "two.sided",exact = NULL)$p.value}
      }
# - histogram p-hodnot:
dev.new(); split.screen(c(1,4))
screen(1)
hist(p.value_NEJ, main="A) NEJBLIZSIMU", xlab="p-hodnota",
     ylab="absolutni cetnosti")

# B) OD MINIMA:
p.value_MIN <- vector(m="numeric", length=p_B)
for (i in 1:p_B)
      {
      if (v_ind[p_A+p_S+i]==1)
           # nemusi jit nutne o znak. promennou (muze byt ord. cis.)
         {
         l <- union(labels(table(data_komplet[1:n_p,p_A+p_S+i]))[[1]],
                    labels(table(data_MIN[1:n_p,p_A+p_S+i]))[[1]])
         kt <- rbind(table(factor(data_komplet[1:n_p,p_A+p_S+i],levels=l)),
                     table(factor(data_MIN[1:n_p,p_A+p_S+i],levels=l)))
         p.value_MIN[i] <- chisq.test(kt)$p.value
         }
      else {p.value_MIN[i] <- ks.test(data_komplet[1:n_p,p_A+p_S+i],data_MIN
           [1:n_p,p_A+p_S+i], alternative = "two.sided",exact = NULL)$p.value}
      }
# - histogram p-hodnot:
screen(2)
hist(p.value_MIN, main="B) OD MINIMA", xlab="p-hodnota",
     ylab="absolutni cetnosti")

# C) OD MAXIMA:
p.value_MAX <- vector(m="numeric", length=p_B)
for (i in 1:p_B)
      {
      if (v_ind[p_A+p_S+i]==1)
           # nemusi jit nutne o znak. promennou (muze byt ord. cis.)
         {
         l <- union(labels(table(data_komplet[1:n_p,p_A+p_S+i]))[[1]],
                    labels(table(data_MAX[1:n_p,p_A+p_S+i]))[[1]])
         kt <- rbind(table(factor(data_komplet[1:n_p,p_A+p_S+i],levels=l)),
                     table(factor(data_MAX[1:n_p,p_A+p_S+i],levels=l)))
         p.value_MAX[i] <- chisq.test(kt)$p.value
         }
      else {p.value_MAX[i] <- ks.test(data_komplet[1:n_p,p_A+p_S+i],data_MAX
           [1:n_p,p_A+p_S+i], alternative = "two.sided",exact = NULL)$p.value}
      }
# - histogram p-hodnot:
screen(3)
hist(p.value_MAX, main="C) OD MAXIMA", xlab="p-hodnota",
     ylab="absolutni cetnosti")
```

```
# D) NAHODNE:
p.value_NAH <- vector(m="numeric", length=p_B)
for (i in 1:p_B)
      {
      if (v_ind[p_A+p_S+i]==1)
            # nemusi jit nutne o znak. promennou (muze byt ord. cis.)
         {
         l <- union(labels(table(data_komplet[1:n_p,p_A+p_S+i]))[[1]],
                    labels(table(data_NAH[1:n_p,p_A+p_S+i]))[[1]])
         kt <- rbind(table(factor(data_komplet[1:n_p,p_A+p_S+i],levels=l)),
                     table(factor(data_NAH[1:n_p,p_A+p_S+i],levels=l)))
         p.value_NAH[i] <- chisq.test(kt)$p.value
         }
      else {p.value_NAH[i] <- ks.test(data_komplet[1:n_p,p_A+p_S+i],data_NAH
           [1:n_p,p_A+p_S+i], alternative = "two.sided",exact = NULL)$p.value}
      }
# - histogram p-hodnot:
screen(4)
hist(p.value_NAH, main="D) NAHODNE", xlab="p-hodnota",
     ylab="absolutni cetnosti")

# - vysledny prehled p-hodnot:
p.value <- list(NEJBLIZSIMU=p.value_NEJ, PRUMER_NEJ=mean(p.value_NEJ,
           na.rm=TRUE), SMER.ODCHYLKA_NEJ=sd(p.value_NEJ,na.rm=TRUE),
           OD_MINIMA=p.value_MIN, PRUMER_MIN=mean(p.value_MIN, na.rm=TRUE),
           SMER.ODCHYLKA_MIN=sd(p.value_MIN,na.rm=TRUE), OD_MAXIMA=p.value_MAX,
           PRUMER_MAX=mean(p.value_MAX,na.rm=TRUE), SMER.ODCHYLKA_MAX=
           sd(p.value_MAX,na.rm=TRUE), NAHODNE=p.value_NAH, PRUMER_NAH=
           mean(p.value_NAH, na.rm=TRUE), SMER.ODCHYLKA_NAH=sd(p.value_NAH,na.rm
    =TRUE))
p.value



## Zachovani zavisloti:
# - Pearsonuv a Spearmanuv korelacni koeficient:
r.pearson_skut <- matrix(NA,nr=p_A ,nc=p_B)  #staci spocitat jen v A (nemeni se)
r.spearman_skut <- matrix(NA,nr=p_A ,nc=p_B) #staci spocitat jen v A (nemeni se)

# A) NEJBLIZSIMU
r.pearson_fuz_NEJ <- matrix(NA,nr=p_A ,nc=p_B)
r.spearman_fuz_NEJ <- matrix(NA,nr=p_A ,nc=p_B)
for (j in 1:p_B)
      {
      for (i in 1:p_A)
         {
         if (v_ind[p_A+p_S+j]==0 & v_ind[i]==0)
            {
            r.pearson_fuz_NEJ[i,j] <- cor(data_komplet[1:n_p,i], data_NEJ
                                          [1:n_p,p_A+p_S+j], method="pearson")
            r.pearson_skut[i,j] <- cor(data_komplet[1:n_p,i], data_komplet
                                       [1:n_p,p_A+p_S+j], method="pearson")
            r.spearman_fuz_NEJ[i,j] <- cor(data_komplet[1:n_p,i], data_NEJ
                                          [1:n_p,p_A+p_S+j], method="spearman")
            r.spearman_skut[i,j] <- cor(data_komplet[1:n_p,i], data_komplet
                                       [1:n_p,p_A+p_S+j], method="spearman")
            }
         }
      }

# - vysledny prehled v tabulce:
r.pearson_NEJ <- abs(r.pearson_fuz_NEJ - r.pearson_skut)
r.spearman_NEJ <- abs(r.spearman_fuz_NEJ - r.spearman_skut)

# - grafy zavislosti kor. koef. pred a po fuzi:
dev.new(); split.screen(c(2,4))
screen(1)
plot(c(r.pearson_skut),c(r.pearson_fuz_NEJ),xlim=c(-0.5,0.5),ylim=c(-0.5,0.5),
      main="A) NEJBLIZSIMU", xlab="Pearsonuv korelacni koeficient pred fuzi",
      ylab="Pearsonuv korelacni koeficient po fuzi")
abline(0,1)
screen(5)
plot(c(r.spearman_skut),c(r.spearman_fuz_NEJ),xlim=c(-0.5,0.5),ylim=c(-0.5,0.5),
```

```
      main="A) NEJBLIZSIMU", xlab="Spearmanuv korelacni koeficient pred fuzi",
      ylab="Spearmanuv korelacni koeficient po fuzi")
abline(0,1)

# B) OD MINIMA:
r.pearson_fuz_MIN <- matrix(NA, nr=p_A ,nc=p_B)
r.spearman_fuz_MIN <- matrix(NA, nr=p_A ,nc=p_B)
for (j in 1:p_B)
      {
      for (i in 1:p_A)
         {
         if (v_ind[p_A+p_S+j]==0 & v_ind[i]==0)
            {
            r.pearson_fuz_MIN[i,j] <- cor(data_komplet[1:n_p,i], data_MIN
                                          [1:n_p,p_A+p_S+j], method="pearson")
            r.spearman_fuz_MIN[i,j] <- cor(data_komplet[1:n_p,i], data_MIN
                                          [1:n_p,p_A+p_S+j], method="spearman")
            }
         }
      }

# - vysledny prehled v tabulce:
r.pearson_MIN <- abs(r.pearson_fuz_MIN - r.pearson_skut)
r.spearman_MIN <- abs(r.spearman_fuz_MIN - r.spearman_skut)

# - grafy zavislosti kor. koef. pred a po fuzi:
screen(2)
plot(c(r.pearson_skut),c(r.pearson_fuz_MIN),xlim=c(-0.5,0.5),ylim=c(-0.5,0.5),
      main="B) OD MINIMA", xlab="Pearsonuv korelacni koeficient pred fuzi",
      ylab="Pearsonuv korelacni koeficient po fuzi")
abline(0,1)
screen(6)
plot(c(r.spearman_skut),c(r.spearman_fuz_MIN),xlim=c(-0.5,0.5),ylim=c(-0.5,0.5),
      main="B) OD MINIMA", xlab="Spearmanuv korelacni koeficient pred fuzi",
      ylab="Spearmanuv korelacni koeficient po fuzi")
abline(0,1)


# C) OD MAXIMA:
r.pearson_fuz_MAX <- matrix(NA, nr=p_A ,nc=p_B)
r.spearman_fuz_MAX <- matrix(NA, nr=p_A ,nc=p_B)
for (j in 1:p_B)
      {
      for (i in 1:p_A)
         {
         if (v_ind[p_A+p_S+j]==0 & v_ind[i]==0)
            {
            r.pearson_fuz_MAX[i,j] <- cor(data_komplet[1:n_p,i], data_MAX
                                          [1:n_p,p_A+p_S+j], method="pearson")
            r.spearman_fuz_MAX[i,j] <- cor(data_komplet[1:n_p,i], data_MAX
                                          [1:n_p,p_A+p_S+j], method="spearman")
            }
         }
      }

# - vysledny prehled v tabulce:
r.pearson_MAX <- abs(r.pearson_fuz_MAX - r.pearson_skut)
r.spearman_MAX <- abs(r.spearman_fuz_MAX - r.spearman_skut)

# - grafy zavislosti kor. koef. pred a po fuzi:
screen(3)
plot(c(r.pearson_skut),c(r.pearson_fuz_MAX),xlim=c(-0.5,0.5),ylim=c(-0.5,0.5),
      main="C) OD MAXIMA", xlab="Pearsonuv korelacni koeficient pred fuzi",
      ylab="Pearsonuv korelacni koeficient po fuzi")
abline(0,1)
screen(7)
plot(c(r.spearman_skut),c(r.spearman_fuz_MAX),xlim=c(-0.5,0.5),ylim=c(-0.5,0.5),
      main="C) OD MAXIMA", xlab="Spearmanuv korelacni koeficient pred fuzi",
      ylab="Spearmanuv korelacni koeficient po fuzi")
abline(0,1)

# D) NAHODNE:
r.pearson_fuz_NAH <- matrix(NA, nr=p_A ,nc=p_B)
```

```
r.spearman_fuz_NAH <- matrix(NA, nr=p_A ,nc=p_B)
for (j in 1:p_B)
      {
      for (i in 1:p_A)
         {
         if (v_ind[p_A+p_S+j]==0 & v_ind[i]==0)
            {
            r.pearson_fuz_NAH[i,j] <- cor(data_komplet[1:n_p,i], data_NAH
                                          [1:n_p,p_A+p_S+j], method="pearson")
            r.spearman_fuz_NAH[i,j] <- cor(data_komplet[1:n_p,i], data_NAH
                                          [1:n_p,p_A+p_S+j], method="spearman")
            }
         }
      }


# - vysledny prehled v tabulce:
r.pearson_NAH <- abs(r.pearson_fuz_NAH - r.pearson_skut)
r.spearman_NAH <- abs(r.spearman_fuz_NAH - r.spearman_skut)

# - grafy zavislosti kor. koef. pred a po fuzi:
screen(4)
plot(c(r.pearson_skut),c(r.pearson_fuz_NAH),xlim=c(-0.5,0.5),ylim=c(-0.5,0.5),
      main="D) NAHODNE", xlab="Pearsonuv korelacni koeficient pred fuzi",
      ylab="Pearsonuv korelacni koeficient po fuzi")
abline(0,1)
screen(8)
plot(c(r.spearman_skut),c(r.spearman_fuz_NAH),xlim=c(-0.5,0.5),ylim=c(-0.5,0.5),
      main="D) NAHODNE", xlab="Spearmanuv korelacni koeficient pred fuzi",
      ylab="Spearmanuv korelacni koeficient po fuzi")
abline(0,1)


# - kompletni prehled korelacnich koeficientu:
r <- list(NEJBLIZSIMU_Pearson=r.pearson_NEJ, prumer_NEJP=mean(r.pearson_NEJ, na.
    rm=TRUE),
          smer.odchylka_NEJP=sd(as.vector(r.pearson_NEJ), na.rm=TRUE),
          NEJBLIZSIMU_Spearman=r.spearman_NEJ,
          prumer_NEJS=mean(r.spearman_NEJ, na.rm=TRUE),
          smer.odchylka_NEJS=sd(as.vector(r.spearman_NEJ), na.rm=TRUE),
          OD_MINIMA_Pearson=r.pearson_MIN,
          prumer_MINP=mean(r.pearson_MIN, na.rm=TRUE),
          smer.odchylka_MINP=sd(as.vector(r.pearson_MIN), na.rm=TRUE),
          OD_MINIMA_Spearman=r.spearman_MIN,
          prumer_MINS=mean(r.spearman_MIN, na.rm=TRUE),
          smer.odchylka_MINS=sd(as.vector(r.spearman_MIN), na.rm=TRUE),
          OD_MAXIMA_Pearson=r.pearson_MAX,
          prumer_MAXP=mean(r.pearson_MAX, na.rm=TRUE),
          smer.odchylka_MAXP=sd(as.vector(r.pearson_MAX), na.rm=TRUE),
          OD_MAXIMA_Spearman=r.spearman_MAX,
          prumer_MAXS=mean(r.spearman_MAX, na.rm=TRUE),
          smer.odchylka_MAXS=sd(as.vector(r.spearman_MAX), na.rm=TRUE),
          NAHODNE_Pearson=r.pearson_NAH,
          prumer_NAHP=mean(r.pearson_NAH, na.rm=TRUE),
          smer.odchylka_NAHP=sd(as.vector(r.pearson_NAH), na.rm=TRUE),
          NAHODNE_Spearman=r.spearman_NAH,
          prumer_NAHS=mean(r.spearman_NAH, na.rm=TRUE),
          smer.odchylka_NAHS=sd(as.vector(r.spearman_NAH), na.rm=TRUE))
r

# - histogramy absolutnich hodnot rozdilu korelacnich koeficientu:
dev.new(); split.screen(c(2,4))
screen(1); hist(r.pearson_NEJ, main="A) NEJBLIZSIMU",
xlab="abs. hodnoty rozdilu Pearsonova kor. koef.", ylab="absolutni cetnosti")
screen(5); hist(r.spearman_NEJ, main="A) NEJBLIZSIMU",
xlab="abs. hodnoty rozdilu Spearmanova kor. koef.", ylab="absolutni cetnosti")

screen(2); hist(r.pearson_MIN, main="B) OD MINIMA",
xlab="abs. hodnoty rozdilu Pearsonova kor. koef.", ylab="absolutni cetnosti")
screen(6); hist(r.spearman_MIN, main="B) OD MINIMA",
xlab="abs. hodnoty rozdilu Spearmanova kor. koef.", ylab="absolutni cetnosti")

screen(3); hist(r.pearson_MAX, main="C) OD MAXIMA",
```

```
xlab="abs. hodnoty rozdilu Pearsonova kor. koef.", ylab="absolutni cetnosti")
screen(7); hist(r.spearman_MAX, main="C) OD MAXIMA",
xlab="abs. hodnoty rozdilu Spearmanova kor. koef.", ylab="absolutni cetnosti")

screen(4); hist(r.pearson_NAH, main="D) NAHODNE",
xlab="abs. hodnoty rozdilu Pearsonova kor. koef.", ylab="absolutni cetnosti")
screen(8); hist(r.spearman_NAH, main="D) NAHODNE",
xlab="abs. hodnoty rozdilu Spearmanova kor. koef.", ylab="absolutni cetnosti")



# - adjustovana rezidua:

if (any(v_ind[1:p_A]==1)==TRUE & any(v_ind[p_A+p_B+1:p]==1)==TRUE ) {
#ma smysl pouze pokud mame nejakou kvalitativni prom. v A_p & v B_p

# A) NEJBLIZSIMU:
adj.rezidua_NEJ <- 0                                    #prumerna hodnota
c <- 0
adj_f_NEJ<- 0; adj_s_NEJ <- 0
for (i in 1:p_B)                                        #kazda nafuzovana/skutecna
   {
   if (v_ind[p_A+p_S+i]==1)                             #ktera sla na prevod na ind.
      {
      for (j in 1:p_A)                                  #s kazdou z A_p
         {
         if (v_ind[j]==1)                               #ktera sla na prevod na ind.
            {
            l1 <- labels(table(data_komplet[,p_A+p_S+i]))[[1]]
            l2 <- labels(table(data_komplet[,j]))[[1]]
            kt_f <- table(factor(data_NEJ[1:n_p,p_A+p_S+i],levels=l1),
                    factor(data_komplet[1:n_p,j],levels=l2))
            kt_s <- table(factor(data_komplet[1:n_p,p_A+p_S+i],levels=l1),
                    factor(data_komplet[1:n_p,j],levels=l2))
            print(abs(chisq.test(kt_f)$residuals-chisq.test(kt_s)$residuals))
            adj_f_NEJ<- c(adj_f_NEJ,c(chisq.test(kt_f)$residuals))
            adj_s_NEJ <- c(adj_s_NEJ,c(chisq.test(kt_s)$residuals))
            c <- c+1
            print(mean(abs(chisq.test(kt_f)$residuals-chisq.test(kt_s)$
            residuals),na.rm=TRUE))
            adj.rezidua_NEJ <- adj.rezidua_NEJ + mean(abs(chisq.test(kt_f)$
                               residuals-chisq.test(kt_s)$residuals),na.rm=TRUE)
            }
         }
      }
    }
adj.rezidua_NEJ <- adj.rezidua_NEJ/c

# - graf zavislosti adjustovanych rezidui pred a po fuzi:
dev.new(); split.screen(c(1,4))
screen(1)
plot(adj_s_NEJ,adj_f_NEJ,xlim=c(-9,17), ylim=c(-9,17),
      main="A) NEJBLIZSIMU", xlab="adjustovana rezidua pred fuzi",
      ylab="adjustovana rezidua po fuzi")
abline(0,1)


# B) OD MINIMA:
adj.rezidua_MIN <- 0                                    #prumerna hodnota
c <- 0
adj_f_MIN<- 0; adj_s_MIN <- 0
for (i in 1:p_B)                                        #kazda nafuzovana/skutecna
   {
   if (v_ind[p_A+p_S+i]==1)                             #ktera sla na prevod na ind.
      {
      for (j in 1:p_A)                                  #s kazdou z A_p
         {
         if (v_ind[j]==1)                               #ktera sla na prevod na ind.
            {
            l1 <- labels(table(data_komplet[,p_A+p_S+i]))[[1]]
            l2 <- labels(table(data_komplet[,j]))[[1]]
            kt_f <- table(factor(data_MIN[1:n_p,p_A+p_S+i],levels=l1),
                    factor(data_komplet[1:n_p,j],levels=l2))
```

```
            kt_s <- table(factor(data_komplet[1:n_p,p_A+p_S+i],levels=l1),
                    factor(data_komplet[1:n_p,j],levels=l2))
            print(abs(chisq.test(kt_f)$residuals-chisq.test(kt_s)$residuals))
            adj_f_MIN <- c(adj_f_MIN,c(chisq.test(kt_f)$residuals))
            adj_s_MIN <- c(adj_s_MIN,c(chisq.test(kt_s)$residuals))
            c <- c+1
            print(mean(abs(chisq.test(kt_f)$residuals-chisq.test(kt_s)$
            residuals),na.rm=TRUE))
            adj.rezidua_MIN <- adj.rezidua_MIN + mean(abs(chisq.test(kt_f)$
                               residuals-chisq.test(kt_s)$residuals),na.rm=TRUE)
            }
         }
      }
   }
adj.rezidua_MIN <- adj.rezidua_MIN/c

# - graf zavislosti adjustovanych rezidui pred a po fuzi:
screen(2)
plot(adj_s_MIN,adj_f_MIN,xlim=c(-9,17), ylim=c(-9,17),
      main="B) OD MINIMA", xlab="adjustovana rezidua pred fuzi",
      ylab="adjustovana rezidua po fuzi")
abline(0,1)

# C) OD MAXIMA:
adj.rezidua_MAX <- 0                                    #prumerna hodnota
c <- 0
adj_f_MAX<- 0; adj_s_MAX <- 0
for (i in 1:p_B)                                        #kazda nafuzovana/skutecna
   {
   if (v_ind[p_A+p_S+i]==1)                             #ktera sla na prevod na ind.
      {
      for (j in 1:p_A)                                  #s kazdou z A_p
         {
         if (v_ind[j]==1)                               #ktera sla na prevod na ind.
            {
            l1 <- labels(table(data_komplet[,p_A+p_S+i]))[[1]]
            l2 <- labels(table(data_komplet[,j]))[[1]]
            kt_f <- table(factor(data_MAX[1:n_p,p_A+p_S+i],levels=l1),
                    factor(data_komplet[1:n_p,j],levels=l2))
            kt_s <- table(factor(data_komplet[1:n_p,p_A+p_S+i],levels=l1),
                    factor(data_komplet[1:n_p,j],levels=l2))
            print(abs(chisq.test(kt_f)$residuals-chisq.test(kt_s)$residuals))
            adj_f_MAX <- c(adj_f_MAX,c(chisq.test(kt_f)$residuals))
            adj_s_MAX <- c(adj_s_MAX,c(chisq.test(kt_s)$residuals))
            c <- c+1
            print(mean(abs(chisq.test(kt_f)$residuals-chisq.test(kt_s)$
            residuals),na.rm=TRUE))
            adj.rezidua_MAX <- adj.rezidua_MAX + mean(abs(chisq.test(kt_f)$
                               residuals-chisq.test(kt_s)$residuals),na.rm=TRUE)
            }
         }
      }
   }
adj.rezidua_MAX <- adj.rezidua_MAX/c

# - graf zavislosti adjustovanych rezidui pred a po fuzi:
screen(3)
plot(adj_s_MAX,adj_f_MAX,xlim=c(-9,17), ylim=c(-9,17),
      main="C) OD MAXIMA", xlab="adjustovana rezidua pred fuzi",
      ylab="adjustovana rezidua po fuzi")
abline(0,1)

# D) NAHODNE:
adj.rezidua_NAH <- 0                                    #prumerna hodnota
c <- 0
adj_f_NAH<- 0; adj_s_NAH <- 0
for (i in 1:p_B)                                        #kazda nafuzovana/skutecna
   {
   if (v_ind[p_A+p_S+i]==1)                             #ktera sla na prevod na ind.
      {
      for (j in 1:p_A)                                  #s kazdou z A_p
         {
         if (v_ind[j]==1)                               #ktera sla na prevod na ind.
```

```
            {
            l1 <- labels(table(data_komplet[,p_A+p_S+i]))[[1]]
            l2 <- labels(table(data_komplet[,j]))[[1]]
            kt_f <- table(factor(data_NAH[1:n_p,p_A+p_S+i],levels=l1),
                    factor(data_komplet[1:n_p,j],levels=l2))
            kt_s <- table(factor(data_komplet[1:n_p,p_A+p_S+i],levels=l1),
                    factor(data_komplet[1:n_p,j],levels=l2))
            print(abs(chisq.test(kt_f)$residuals-chisq.test(kt_s)$residuals))
            adj_f_NAH <- c(adj_f_NAH,c(chisq.test(kt_f)$residuals))
            adj_s_NAH <- c(adj_s_NAH,c(chisq.test(kt_s)$residuals))
            c <- c+1
            print(mean(abs(chisq.test(kt_f)$residuals-chisq.test(kt_s)$
            residuals),na.rm=TRUE))
            adj.rezidua_NAH <- adj.rezidua_NAH + mean(abs(chisq.test(kt_f)$
                               residuals-chisq.test(kt_s)$residuals),na.rm=TRUE)
            }
          }
      }
    }
adj.rezidua_NAH <- adj.rezidua_NAH/c

# - graf zavislosti adjustovanych rezidui pred a po fuzi:
screen(4)
plot(adj_s_NAH,adj_f_NAH,xlim=c(-6,6), ylim=c(-6,6),
      main="D) NAHODNE", xlab="adjustovana rezidua pred fuzi",
      ylab="adjustovana rezidua po fuzi")
abline(0,1)

# - vysledny prehled v seznamu:
adj.rezidua <- list(NEJBLIZSIMU=adj.rezidua_NEJ, OD_MINIMA=adj.rezidua_MIN,
                OD_MAXIMA=adj.rezidua_MAX, NAHODNE=adj.rezidua_NAH)
adj.rezidua


}


## Shoda individualnich promennych:
# A) NEJBLIZSIMU:
shoda_NEJ <- vector(m="numeric", length=p_B)
for (i in 1:p_B)
      {
      l <- labels(table(data_komplet[,p_A+p_S+i]))[[1]]
      kt <- table(factor(data_NEJ[1:n_p,p_A+p_S+i],levels=l),
            factor(data_komplet[1:n_p,p_A+p_S+i],levels=l))
      shoda_NEJ[i] <- sum(diag(kt))/n_p
      }
# - histogram shod:
dev.new(); split.screen(c(1,4))
screen(1)
hist(shoda_NEJ, main="A) NEJBLIZSIMU", xlab="shoda ind. promennych",
     ylab="absolutni cetnosti", breaks=30, ylim=c(0,10))

# B) OD MINIMA:
shoda_MIN <- vector(m="numeric", length=p_B)
for (i in 1:p_B)
      {
      l <- labels(table(data_komplet[,p_A+p_S+i]))[[1]]
      kt <- table(factor(data_MIN[1:n_p,p_A+p_S+i],levels=l),
            factor(data_komplet[1:n_p,p_A+p_S+i],levels=l))
      shoda_MIN[i] <- sum(diag(kt))/n_p
      }
# - histogram shod:
screen(2)
hist(shoda_MIN, main="B) OD MINIMA", xlab="shoda ind. promennych",
     ylab="absolutni cetnosti", breaks=30, ylim=c(0,10))

# C) OD MAXIMA:
shoda_MAX <- vector(m="numeric", length=p_B)
for (i in 1:p_B)
      {
      l <- labels(table(data_komplet[,p_A+p_S+i]))[[1]]
      kt <- table(factor(data_MAX[1:n_p,p_A+p_S+i],levels=l),
```

```
            factor(data_komplet[1:n_p,p_A+p_S+i],levels=l))
      shoda_MAX[i] <- sum(diag(kt))/n_p
      }
# - histogram shod:
screen(3)
hist(shoda_MAX, main="C) OD MAXIMA", xlab="shoda ind. promennych",
     ylab="absolutni cetnosti", breaks=30, ylim=c(0,10))

# D) NAHODNE:
shoda_NAH <- vector(m="numeric", length=p_B)
for (i in 1:p_B)
      {
      l <- labels(table(data_komplet[,p_A+p_S+i]))[[1]]
      kt <- table(factor(data_NAH[1:n_p,p_A+p_S+i],levels=l),
            factor(data_komplet[1:n_p,p_A+p_S+i],levels=l))
      shoda_NAH[i] <- sum(diag(kt))/n_p
      }
# - histogram shod:
screen(4)
hist(shoda_NAH, main="D) NAHODNE", xlab="shoda ind. promennych",
     ylab="absolutni cetnosti", breaks=30, ylim=c(0,10))

# - vysledny prehled v seznamu:
shoda <- list(NEJBLIZSIMU=shoda_NEJ, prumerNEJ=mean(shoda_NEJ,
             na.rm=TRUE),smer.odchylkaNEJ=sd(shoda_NEJ, na.rm=TRUE),
             OD_MINIMA=shoda_MIN, prumerMIN=mean(shoda_MIN,na.rm=TRUE),
             smer.odchylkaMIN=sd(shoda_MIN, na.rm=TRUE), OD_MAXIMA=shoda_MAX,
             prumerMAX=mean(shoda_MAX,na.rm=TRUE), smer.odchylkaMAX=sd(shoda_MAX,
             na.rm=TRUE),NAHODNE=shoda_NAH, prumerNAH=mean(shoda_NAH,na.rm=TRUE),
             smer.odchylkaNAH=sd(shoda_NAH, na.rm=TRUE))
shoda


## Linearni model zkoumajici zavislost hodnot korelacnich koeficientu
## pred fuzi na hodnotach korelacnich koeficientu po fuzi:

# - vyuzivame pro lepsi moznosti porovnani metod prirazovani
# (zejmena v pripade MML dat)
#lm_NEJ <- summary(lm(c(r.pearson_skut)~c(r.pearson_fuz_NEJ)))
#lm_MIN <- summary(lm(c(r.pearson_skut)~c(r.pearson_fuz_MIN)))
#lm_MAX <- summary(lm(c(r.pearson_skut)~c(r.pearson_fuz_MAX)))
#lm_NAH <- summary(lm(c(r.pearson_skut)~c(r.pearson_fuz_NAH)))

#lm_NEJ; lm_MIN; lm_MAX; lm_NAH
```

Příloha 3: Program Generování dat

```
setwd(" ") #nutno zadat cestu do prislusneho adresare
rm(list=ls())
                              ### GENEROVANI DAT ###

## Zadani dimenze tabulek A_p, S, B_d:
# - znaceni: dim(A_p): n_p, p_A
#            dim(S): n_p+n_d,p_S
#            dim(B_d): n_d, p_B
#            => p_A+p_S+p_B=p
n_p <- 400
n_d <- 600
p_A <- 6; p_S <- 8; p_B <- 6
p <- p_A+p_S+p_B

# - ulozeni vektoru dimenzi:
v_dim <- c(n_p,n_d,p_A,p_S,p_B)
save(v_dim, file="v_dim.txt")

## Generovani dat:

# - dataS (spojovaci):
# 1) pohlavi (0-muz, 1-zena):
set.seed(100); pohlavi <- sample(0:1, (n_p+n_d), replace=TRUE)

# 2) vek (18-50):
set.seed(100); vek <- sample(18:50, (n_p+n_d), replace=TRUE)
```

```
# 3) rodinny stav ("svobodny", "svobodna", "zenaty", "vdana", "druh", "druzka",
#    "rozvedeny", "rozvedena", "vdovec", "vdova"):
rodinny_stav <- vector(m="character", length=(n_p+n_d)); set.seed(300)
for (i in 1:(n_p+n_d))
      {
      if (pohlavi[i]==0)
         {
         if (vek[i]>35)
           {rodinny_stav[i] <- sample(c("svobodny","zenaty","druh","rozvedeny",
           "vdovec"), 1, prob=c(15,30,20,30,10))}
         if (vek[i]<=35 & vek[i]>=25)
           {rodinny_stav[i] <- sample(c("svobodny","zenaty","druh","rozvedeny",
           "vdovec"), 1, prob=c(15,35,25,23,2))}
         if (vek[i]<25)
           {rodinny_stav[i] <- sample(c("svobodny","zenaty","druh","rozvedeny",
           "vdovec"), 1, prob=c(40,5,45,4,1))}
         }
      else
         {
         if (vek[i]>35)
          {rodinny_stav[i] <- sample(c("svobodna","vdana","druzka","rozvedena",
           "vdova"),1, prob=c(10,30,30,23,7))}
         if (vek[i]<=35 & vek[i]>=25)
          {rodinny_stav[i] <- sample(c("svobodna","vdana","druzka","rozvedena",
           "vdova"),1, prob=c(25,30,30,13,2))}
         if (vek[i]<25)
          {rodinny_stav[i] <- sample(c("svobodna","vdana","druzka","rozvedena",
           "vdova"),1, prob=c(50,10,35,4,1))}
         }
      }

# 4) pohlavi partnera (0-muz, 1-zena):
pohlavi_partnera <- vector(m="numeric", length=(n_p+n_d)); set.seed(100)
for (i in 1:(n_p+n_d))
      {pohlavi_partnera[i] <- sample(c((pohlavi[i] + 1)%%2,pohlavi[i]), 1,
       prob=c(96,4))}

# 5) pocet deti ("zadne", "1", "2", "3 az 4", "5 a vice"):
pocet_deti <- vector(m="character", length=(n_p+n_d)); set.seed(100)
for (i in 1:(n_p+n_d))
      {
      if (vek[i]>28 & (rodinny_stav[i]=="svobodny"|rodinny_stav[i]=="svobodna"))
            {pocet_deti[i] <- sample(c("zadne","1","2","3 az 4","5 a vice"), 1,
             prob=c(60,35,3,1,1))}
      else {pocet_deti[i] <- sample(c("zadne","1","2","3 az 4","5 a vice"), 1,
            prob=c(5,30,45,15,5))}
      if (vek[i]<=28 & (rodinny_stav[i]=="zenaty"|rodinny_stav[i]=="vdana"))
            {pocet_deti[i] <- sample(c("zadne","1","2","3 az 4","5 a vice"), 1,
            prob=c(22,50,20,6,2))}
      else {pocet_deti[i] <- sample(c("zadne","1","2","3 az 4","5 a vice"), 1,
            prob=c(40,40,16,3,1))}
      }

# 6) pocet vnoucat ("zadne", "1", "2", "3 az 4", "5 a vice"):
pocet_vnoucat <- vector(m="character", length=(n_p+n_d)); set.seed(100)
for (i in 1:(n_p+n_d))
      {
      if (pocet_deti[i]=="zadne" | vek[i]<35) {pocet_vnoucat[i] <- "zadne"}
      else
         {
         if ((pocet_deti[i]=="1" | pocet_deti[i]=="2") & vek[i]>45)
            {pocet_vnoucat[i] <- sample(c("zadne","1","2","3 az 4","5 a vice"),
            1, prob=c(5,15,30,35,15))}
         else
            {
            if ((pocet_deti[i]=="3 az 4" | pocet_deti[i]=="5 a vice") &
               vek[i]>45)
               {pocet_vnoucat[i] <- sample(c("zadne","1","2","3 az 4",
               "5 a vice"), 1, prob=c(3,12,20,35,40))}
            else {pocet_vnoucat[i] <- sample(c("zadne","1","2","3 az 4",
               "5 a vice"), 1, prob=c(40,40,15,3,2))}
            }
```

```
        }
    }

# 7) kraj ( "PHA", "STC", "JHC", "PLK", "KVK", "ULK", "LBK", "HKK", "PAK",
#           "VYS","JHM", "OLK", "ZLK", "MSK"):
set.seed(100)
kraj <- sample(c("PHA","STC","JHC","PLK","KVK","ULK","LBK","HKK","PAK","VYS",
      "JHM","OLK","ZLK","MSK"), (n_p+n_d), replace=TRUE)

# 8) hruba mzda zaokrouhlena na tisice (7000-50000):
# (pozn.: u studentu kapesne ci vydelek pri skole)
mzda <- vector(m="numeric", length=(n_p+n_d)); set.seed(100)
for (i in 1:(n_p+n_d))
    {
    if (vek[i]>35)
        {
        if (kraj[i]=="PHA") {mzda[i] <- sample(c(15:50), 1,prob=c(rep(30,11),
                            rep(40,10),rep(40,15)))}
        else {mzda[i] <- sample(c(7:50), 1,prob=c(rep(30,9),rep(50,10),
                         rep(10,10),rep(10,15)))}
        }
    else
        {
        if (kraj[i]=="PHA") {mzda[i] <- sample(c(15:50), 1,prob=c(rep(30,11),
                            rep(40,10),rep(40,15)))}
        else {mzda[i] <- sample(c(7:50), 1,prob=c(rep(50,9),rep(35,10),
                         rep(10,10),rep(5,15)))}
        }
    }
mzda <- mzda*1000

# - slouceni:
dataS <- data.frame(pohlavi,vek,rodinny_stav,pohlavi_partnera,pocet_deti,
         pocet_vnoucat,kraj,mzda)

# - dataA (ctenost tistenych medii):
# 1) cetnost cteni medii (0 necte, 1 - 1x za tyden, ... 7 - kazdy den):
set.seed(300); cetnost_cteni <- sample( 0:7, (n_p+n_d), replace=TRUE);

# 2) predplatne (0 nema, 1 ma):
predplatne <- vector(m="numeric", length=(n_p+n_d)); set.seed(100)
for (i in 1:(n_p+n_d))
    {
    if(cetnost_cteni[i]>2)
        {
        if (mzda[i]>18000){predplatne[i]<- sample(c(0,1), 1, prob=c(40,60))}
        else {predplatne[i]<- sample(c(0,1), 1, prob=c(70,30))}
        }
    }

# 3) Metro (0 necte, 1 cte):
metro <- vector(m="numeric", length=(n_p+n_d)); set.seed(100)
for (i in 1:(n_p+n_d))
    {
    if (cetnost_cteni[i]>0 & kraj[i]=="PHA") {metro[i] <- sample(c(0,1), 1,
          prob=c(10,90))}
    }

# 4) tydenni utrata za noviny a casopisy zaokrouhlena na stovky (0-1000):
tydenni_utrata <- vector(m="numeric", length=(n_p+n_d)); set.seed(100)
for (i in 1:(n_p+n_d))
    {
    if (cetnost_cteni[i]==0) {tydenni_utrata[i] <- 0}
    else
        {
        if (mzda[i]>35000 & cetnost_cteni[i]>3 & predplatne[i]==1)
           {tydenni_utrata[i] <- sample(c(0:10), 1, prob=c(2,rep(5,4),
           rep(16,3),rep(10,3)))}
        else
           {tydenni_utrata[i] <- sample(c(0:10), 1, prob=c(20,30,20,20,3,2,
           rep(1,5)))}
        }
    }
```

```
tydenni_utrata <- tydenni_utrata * 100

# 5) Nejcastejsi oblast zajmu - noviny ("zpravy", "politika a ekonomie",
#    "sport", "kultura", "jine"):
noviny <- vector(m="character", length=(n_p+n_d)); set.seed(100)
for (i in 1:(n_p+n_d))
      {
      if (pohlavi[i]==0)
            {noviny[i] <- sample(c("zpravy","politika a ekonomie","sport",
             "kultura","jine"), 1, prob=c(30,25,30,5,10))}
      else
            {noviny[i] <- sample(c("zpravy","politika a ekonomie","sport",
             "kultura","jine"), 1, prob=c(40,10,10,30,10))}
      }

# 6) Nejcastejsi oblast zajmu - casopisy ("moda", "dum a zahrada", "auto-moto",
#    "vareni", "jine"):
casopisy <- vector(m="character", length=(n_p+n_d)); set.seed(100)
for (i in 1:(n_p+n_d))
      {
      if (pohlavi[i]==0)
         {casopisy[i] <- sample(c("moda","dum a zahrada","auto-moto",
          "vareni","jine"), 1, prob=c(4,10,50,10,26))}
      else
         {
         if (rodinny_stav[i]=="vdana" | rodinny_stav[i]=="druzka" |
         pocet_deti[i]!="zadne")
            {casopisy[i] <- sample(c("moda","dum a zahrada","auto-moto",
             "vareni","jine"), 1, prob=c(25,24,1,40,10))}
         else
             {casopisy[i] <- sample(c("moda","dum a zahrada","auto-moto",
             "vareni","jine"), 1, prob=c(50,10,1,30,9))}
         }
      }

# - slouceni:
dataA <- data.frame(cetnost_cteni,predplatne,metro,tydenni_utrata,noviny,
         casopisy)

# - dataB (stravovani):
# 1) nejcastejsi misto obedu ("doma", "restaurace", "menza", "jidelna",
#    "neobedva"):
obed <- vector(m="character", length=(n_p+n_d)); set.seed(100)
for (i in 1:(n_p+n_d))
      {
      if (vek[i]<26){obed[i] <- sample(c("doma","restaurace","menza","jidelna",
                     "neobedva"), 1, prob=c(15,5,55,10,15))}
      else
         {
         if (mzda[i]>25000)
            {obed[i] <- sample(c("doma","restaurace","menza","jidelna",
             "neobedva"), 1, prob=c(20,50,2,10,18))}
         else
            {obed[i] <- sample(c("doma","restaurace","menza","jidelna",
             "neobedva"), 1, prob=c(30,25,2,25,18))}
         }
      }

# 2) pravidelna strava (0 ne, 1 ano):
pravidelna_strava <- vector(m="numeric", length=(n_p+n_d)); set.seed(100)
for (i in 1:(n_p+n_d))
      {
      if (obed[i]=="neobedva") {pravidelna_strava[i] <- 0}
      else
         {
         if (predplatne[i]==1)
            {pravidelna_strava[i] <- sample(c(0,1), 1,
             prob=c(30,70))}
         else
            {pravidelna_strava[i] <- sample(c(0,1), 1,
             prob=c(40,60))}
         }
      }
```

```
# 3) fast food (0 ne, 1 ano):
fast_food <- vector(m="numeric", length=(n_p+n_d)); set.seed(100);
for (i in 1:(n_p+n_d))
      {
      if (metro[i]==1)
         {fast_food[i] <- sample(c(0,1), 1, prob=c(70,30))}
      else
         {fast_food[i] <- sample(c(0,1), 1, prob=c(40,60))}
      }

# 4) zdrava strava (0 ne, 1 ano):
zdrava_strava <- vector(m="numeric", length=(n_p+n_d)); set.seed(100)
for (i in 1:(n_p+n_d))
      {
      if(pravidelna_strava[i]==1 & fast_food[i]==0)
         {zdrava_strava[i] <- sample(c(0,1), 1, prob=c(30,70))}
      }

# 5) denni utrata za jidlo zaokrouhlena na stovky (100-1200):
denni_utrata <- vector(m="numeric", length=(n_p+n_d)); set.seed(100)
for (i in 1:(n_p+n_d))
      {
      if (obed[i]=="menza") {denni_utrata[i] <- sample(c(1:4), 1,
                                                prob=c(38,45,15,1))}
      else
         {
         if (tydenni_utrata[i]>400)
            {denni_utrata[i] <- sample(c(1:12), 1,
                              prob=c(5,5,15,20,15,10,rep(5,6)))}
         else
            {
             if (mzda[i]>35000 &  obed[i]=="restaurace")
                  {denni_utrata[i] <- sample(c(2:12), 1,
                                    prob=c(rep(9,3),rep(15,4),rep(1,4)))}
              else
                  {denni_utrata[i] <- sample(c(1:12), 1,
                                    prob=c(10,40,30,10,3,2,rep(1,6)))}
             }
         }
      }
denni_utrata <- denni_utrata * 100

# 6) cetnost vareni dotycneho (0 nevari, 1 - 1x za tyden, ... 7 - kazdy den):
cetnost_vareni <- vector(m="numeric", length=(n_p+n_d)); set.seed(100)
for (i in 1:(n_p+n_d))
      {
      if (casopisy[i]=="vareni")
         {cetnost_vareni[i] <- sample(c(0:7), 1, prob=
          c(1,2,2,15,15,20,20,25))}
      else
         {
         if (pohlavi[i]==1)
            {
            if (pocet_deti[i]!="zadne")
               {cetnost_vareni[i] <- sample(c(0:7), 1, prob=
                c(3,5,25,8,8,20,6,25))}
            else
               {cetnost_vareni[i] <- sample(c(0:7), 1, prob=
                c(10,5,40,10,10,10,10,10))}
            }
         else
            {
            if (rodinny_stav[i]=="rozvedeny" & pocet_deti[i]!="zadne")
               {cetnost_vareni[i] <- sample(c(0:7), 1, prob=
               c(35,5,30,5,10,5,5,5))}
            else
               {cetnost_vareni[i] <- sample(c(0:7), 1, prob=
                c(50,30,10,2,2,2,2,2))}
            }
         }
      }
```

```
# - slouceni:
dataB <- data.frame(obed, pravidelna_strava, fast_food,
          zdrava_strava, denni_utrata, cetnost_vareni)

## Slouceni tabulek:
data <- data.frame(dataA, dataS, dataB)

## Rozliseni promennych, ktere je potreba prevest na soubor ind:
# prevadime vsechny nominalni a ordinalni (nebinarni)
# v_ind[i] = 1, pokud ma byt dana promenna prevedena
#          = 0, jinak
v_ind <- c(1,0,0,0,1,1,0,0,1,0,1,1,1,0,1,0,0,0,0,1)

# - ulozeni vekoru v_ind:
save(v_ind, file="v_ind.txt")

## Ulozeni kompletnich dat (pro report):
data_komplet <- data
save(data_komplet, file="data_komplet.txt")

## "Umazani" dat, ktere se budou fuzovat:
data[1:n_p,(p_A+p_S+1):p] <- NA
data[(n_p+1):(n_p+n_d),1:p_A] <- NA

## Ulozeni dat pripravenych k fuzi:
save(data, file="data.txt")
```

Příloha 4: Program Nachystání MML dat

```
setwd(" ") #nutno zadat cestu do prislusneho adresare
rm(list=ls())

                              ### NACHYSTANI MML DAT ###

## Nacteni dat:
library(foreign)
data <- read.spss("MML3.SAV", to.data.frame=T) #bunky datove tabulky typu factor
data <- data[1:3000,]


## Zadani dimenze tabulek A_p, S, B_d:
# - znaceni: dim(A_p): n_p, p_A
#            dim(S): n_p+n_d, p_S
#            dim(B_d): n_d, p_B
#            => p_A+p_S+p_B=p

p <- dim(data)[2]
p_A <- 269; p_S <- 108; p_B <- (p-p_A-p_S)

# - zvolime kolik radku budeme fuzovat (predpokladame n_d>n_p):
n_p <- dim(data)[1]/3
n_p <- trunc(n_p)             # ... zaokrouhlime na cele cislo
n_d <- dim(data)[1]-n_p

# ulozeni vektoru dimenzi c(n_p,n_d,p_A,p_S,p_B):
v_dim <- c(n_p,n_d,p_A,p_S,p_B)
save(v_dim, file="v_dim.txt")

## Permutace dat:
set.seed(100)
q <- sample.int(n_p+n_d, size =n_p+n_d, replace = FALSE, prob = NULL)
data <- data[q,]

## Rozliseni promennych, ktere je potreba prevest na soubor ind:
# v_ind[i] = 1, pokud ma byt dana promenna prevedena
#          = 0, jinak

v_ind <- vector(mode="numeric", length=p)

# - oznacime vsechny nominalni a ordinalni (nebinarni):
c <- c(270:280,283:284,287:293,296:297,312:314,335,348:376,379:381,
       405:407,428:432,456,474:475,491:496,511:512,529:535,538,540,
       542:543,545:546,548:549,551:555)
```

```
v_ind[c] <- 1

# - ulozeni vekoru v_ind:
save(v_ind, file="v_ind.txt")


## Prevod ciselnych faktoru na typ numeric:
for (j in 1:p) { if (v_ind[j]==0) {data[,j] <- as.numeric(data[,j])}}

# posunuti hodnot promennych, aby sedely s puvodnimi hodnotami ze souboru:
d <- c(282,286,337)
e <- c(1:18,27:51,68:109,136:269,295,298:300,302:310,316:331,339:346,
       382:404,408:427,433:455,463:473,476:490,499:509,513:527)
data[,d] <- data[,d]+11
data[,e] <- data[,e]-1


## Prepsani <NA> hodnot na prumernou hodnotu \ "nezodpovezeno":
for (j in 1:p)
      {
      if(any(is.na(data[,j])))
         {
         if (v_ind[j]==1)
            {
            levels(data[,j]) <- c(levels(data[,j]),"neodpovezeno")
            data[is.na(data[,j]),j] <- "neodpovezeno"
            }
         else {data[is.na(data[,j]),j] <- mean(data[,j], na.rm=TRUE)}
         }
      }

#overeni - musi platit: any(is.na(data))==FALSE

## Prevod znakovych faktoru na typ character:
for (j in 1:p)
      { if (v_ind[j]==1) {data[,j] <- as.character(data[,j])}}

## Ulozeni fuzovane casti dat (pro report):
data_komplet <- data
save(data_komplet, file="data_komplet.txt")

## "Umazani" dat, ktere se budou fuzovat:
data[1:n_p,(p_A+p_S+1):p] <- NA
data[n_p+1:n_p+n_d,1:p_A] <- NA

## Ulozeni dat pripravenych k fuzi:
save(data, file="data.txt")
```