Kontrola kvality vstupních dat a jejich použitelnost

Ministerstvo práce a sociálních věcí České republiky

EY

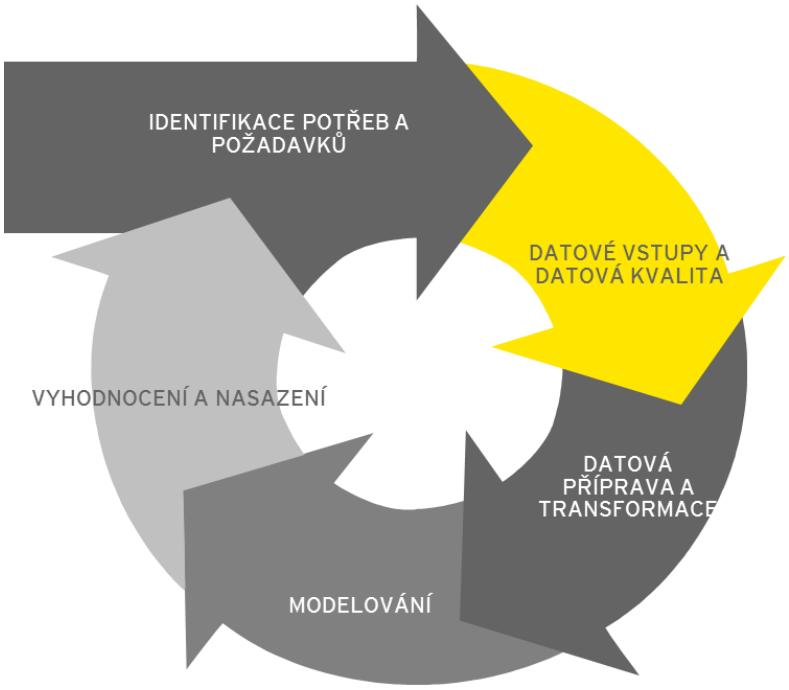
Building a better working world

Obsah

1. Úvod 2
2. Datové vstupy a datová kvalita 3
2.1 Sběr dat 3
2.2 Popis dat 9
2.3 Průzkum dat 13
2.4 Verifikace dat 18
3. Datová příprava a transformace 30
3.1 Výběr dat 30
3.2 Čištění dat 34
3.3 Transformace dat 39
3.4 Integrace a formátování 44
4. Závěr 47

1. Úvod

Tento dokument popisuje metodiku datové kvality pro vytvoření klíčových podmínek pro datovou přípravu a následné úspěšné vytvoření mikrosimulačního modelu. Na základě našich získaných zkušeností víme, že v rámci procesu konstrukce prediktivních statistických modelů je často fáze kontroly datové kvality a přípravy dat podceněna, a není ji věnován dostatečný čas. To má poté negativní vliv na výsledný model. Primárně je nutné vytvoření podmínek pro dosažení odpovídající kvality dat. Námi použitý postup vychází z metodologie CRISP-DM (CRoss-Industry Standard Process for Data Mining) a teoretických přístupů z různých odvětví. Tento proces se skládá z pěti částí (viz obrázek 1). Pořadí jednotlivých fází není pevně stanoveno, často totiž mohou nastat situace, které vyvolají potřebu návratu k předchozí aktivitě. Znázorněný obrázek postihuje cyklickou podstatu samotného data-miningu a modelování - tento proces typicky nekončí nalezením jediného správného řešení, ale získané zkušenosti poslouží k efektivnějšímu zpracování dat v budoucnosti.



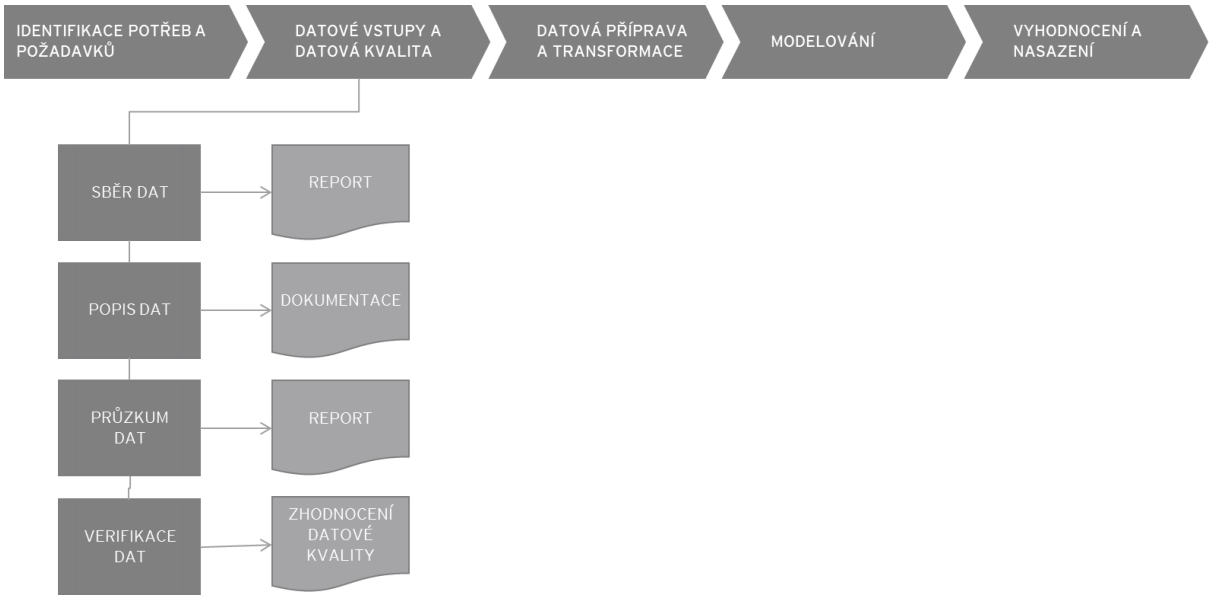
Obrázek 1: Znázornění vztahů mezi jednotlivými fázemi použitého postupu

Pro účely hodnocení kvality vstupních dat využijeme především druhou a třetí fázi znázorněného postupu, které se týkají porozumění nasbíraných dat a jejich přípravou k dalšímu použití. Každá z těchto fází se skládá z několika dílčích úloh.

2. Datové vstupy a datová kvalita

Předchozí příprava a pochopení obsahu zdrojových dat je nezbytná k provedení vlastního hodnocení datové kvality pro další testování. Kromě vlastního obsahu vstupních dat se v první části našeho procesu budeme zabývat také jejich zdroji a způsobem jejich sběru.

Následující obrázek znázorňuje jednotlivé fáze procesu hodnocení datových vstupů a datové kvality. Tento proces se skládá ze 4 dílčích částí, z nichž každá poskytne výstup (report nebo jiný druh dokumentace), který je nezbytný pro potřeby následujících analýz a poté i samotného modelování. Fáze porozumění datům začíná s úvodním sběrem dat a pokračuje aktivitami, které umožňují lepší poznání zpracovávaných dat, identifikaci problémů datové kvality a odhalení zajímavých podmnožin z hlediska následného modelování, které umožňují formulaci prvních hypotéz.



Obrázek 2: Dílčí úlohy a výstupy týkající se fáze datových vstupů a datové kvality

2.1 Sběr dat

Data pro vstup do mikrosimulačního modelu podle nám dostupných informací pocházejí z více zdrojů, hlavními z nich jsou existující databáze státních organizací (především databáze České správy sociálního zabezpečení, Ministerstva práce a sociálních věcí a Českého statistického úřadu).

2.1.1 Cíl

Cílem této fáze je:

- Zajištění přístupu k datům a jejich stažení
- Načtení dat do používaného nástroje k dalšímu zpracování
- Ověření propojení a identifikace případných nedostatků při použití dat z různých zdrojů

2.1.2 Výstup

Výstupem této fáze je vytvoření reportu sběru dat. Tento report:

- Je vytvořen na základě analýzy nasbíraných dat
- Obsahuje dostupné datové zdroje pro modelování a jejich základní popis (obsah, základní zobrazovanou jednotku, formu obsažených dat, …)

- Zárodek tohoto reportu existuje v rámci provedené rešerše dat *Rešerše zdrojů dat o důchodových právech*, která byla ze strany MPSV poskytnuta při zahájení projektu, nicméně tento dokument je potřeba upravit a udržovat aktualizovaný
- Zároveň obsahuje návrhy na další rozvoj
    - Zlepšení podkladových databází
        - Rozšíření podkladových databází
        - Změny struktury a granularity dat
    - Návrhy na zefektivnění metod sběru

Poznámka: Základní report sběru dat je nezbytný z pohledu dalšího využití dat a vhodný ke kombinaci s reporty z dalších částí (popis, průzkum a verifikace dat).

2.1.3 Činnost

Před samotným sběrem dat je potřeba specifikovat:

- Jaké informace budou pro modelování potřeba
- Zda se z uvažovaných zdrojů dají tyto informace získat
- Pokud uvažované zdroje všechny potřebné informace neobsahují, je potřeba zvážit následující možnosti
    - Extrahovat informace z jiné dostupné databáze
    - Získat je kombinací již známých informací

Pokud budou data kombinovaná z více než jedné databáze, je nezbytné prověřit

- Stejnou úroveň podrobnosti všech dat (na úrovni ID záznamu)
- Unikátní klíč umožňujícího propojení dat z různých databází
- Časové období, ze kterého data pocházejí (propojení z časového hlediska)
    - Zde je také potřeba určit, s jak dlouhou historií je vhodné pracovat
        - Příliš krátké období nemusí odhalit možnou přítomnost trendu
        - Příliš dlouhé období může naopak souviset s technickou náročností zpracování dat – je nutné zvážit
            - Konzistenci všech podmínek, které mohou ovlivnit výsledek
                - Možnost případného očištění dat, např. o roky a měsíce, ve kterých došlo k náhlým změnám podmínek odchodu do důchodu

Po určení vhodných zdrojových tabulek (případně souborů) následuje výběr dat pro potřeby dalšího modelování za zohlednění některých zásadních kritérií, např.

- Maximální počet atributů, které se dají modelovací technikou zvládnout
- (Ne)důležitost některých atributů pro potřeby modelování

Závěrečným krokem této části analýzy datových vstupů a datové kvality je doplnění základního reportu sběru dat o následující informace ke každému použitému datovému zdroji

- Dostupnost zdroje
- Použitelné informace obsažené v tomto zdroji
- Frekvenci aktualizace údajů
- Formát dat
- Základní zobrazovanou jednotku
- Vlastní zdroj dat a kontaktní osoba (zejména důležité pro data, která nevlastní MPSV)

2.1.4 Doporučení

Protože v současné době neexistuje souhrnná databáze požadovaných informací a zároveň vstupní data pro následné modelování požadují vysokou podrobnost na individuální úrovni jedinců, je třeba využít několika separátních datových zdrojů. Propojení datových zdrojů na základě jednoznačných identifikátorů může vést k vzniku následujících problémů:

- Jednomu identifikátoru odpovídá více záznamů
- Dojde k propojení neodpovídajících si záznamů
- Nedojde k propojení odpovídajících si záznamů

Pokud dojde k výskytu některé z těchto možností, je v tomto kroku pro další práci s daty nezbytné:

- Správně problém identifikovat
- Popsat problém a zvážit další postup

2.1.5 Specifický příklad

Pro účely ukázky datové kvality budeme pracovat se třemi v současné době dostupnými individuálními databázemi

- INEP,
- STATMIN VZ,
- STATMIN ANOD.

Na těchto datových zdrojích budeme demonstrovat vzorové kroky v rámci fáze sběru dat směřující k jejich dalšímu zpracování v rámci následných kroků. Zaměříme se především na nahrání dat do systému, základní úpravy a přetypování, následně získáme první pohled na zpracovávaná data a odhalíme tak část problémů, které se mohou při zpracování zmíněných databází vyskytnout.

V následujícím textu jsme se rozhodli nevyužít databázi NEM, která byla ve studii proveditelnosti *Studie proveditelnosti – Implementace individuálních rozhodovacích procesů do dynamického mikrosimulačního modelu důchodového systému MPSV* (dále jen "Studie proveditelnosti") určena jako další potenciální zdroj nových informací. Důvodem je skutečnost, že ačkoliv databáze NEM obsahuje identifikační čísla osob, tyto čísla neodpovídají `ID` uvedeným v dříve zmíněných databázích (INEP, STATMIN VZ, STATMIN ANOD). Neexistence unikátního klíče, který by umožnil propojení, tak zamezuje v současné době využití informací z databáze NEM v propojení s individuálními daty z ostatních zdrojů.

Prvními kroky ve fázi sběru dat jsou následující

- Import zpracovávaných databází do prostředí SQL
- Převod načtených dat na správné datové typy uvedené v datových větách

Načtení databází do prostředí SQL jsme pro jednoduchost provedli ručně pomocí zabudované importní funkce. Tímto postupem jsme však narazili na některé nedostatky v nově vytvořených tabulkách. Takovými závadami jsou

- Nepřítomnost hlavičky v databázi STATMIN ANOD,
- Všechny hodnoty proměnných jsou v uvozovkách v databázích INEP a STATMIN VZ,
- Nevhodné datové typy proměnných ve všech databázích (všechna data byla načtena jako znaky).

Pomocí využití jiných importních metod (např. BULK LOAD metod), lze tomuto problému předejít, jedná se již však o implementační úlohu rozsahu přesahující zvolené ukázky.

V následujícím skriptu ukážeme na databázi STATMIN VZ, jak je možné odstranit přítomnost uvozovek ve všech polích jednotlivých proměnných odebráním prvních a posledních dvou znaků. Zároveň po odstranění přebytečných uvozovek v databázích STATMIN VZ i INEP přetypujeme ve všech datových zdrojích všechny proměnné na správné datové typy, neboť při manuálním importu byly tyto proměnné načtené jako řetězce

```
select cast(substring(ID_VZZAM_AN, 2, len(ID_VZZAM_AN)-2) as bigint) as ID_VZZAM_AN,
          cast(substring(ID_ELDP, 2, len(ID_ELDP)-2) as bigint) as ID_ELDP,
          cast(substring(ID, 2, len(ID)-2) as bigint) as ID,
          cast(substring(pohlavi, 2, len(pohlavi)-2) as int) as pohlavi,
          cast(substring(roknar, 2, len(roknar)-2) as int) as roknar,
          cast(substring(psc, 2, len(psc)-2) as int) as psc,
          cast(substring(kvc, 2, len(kvc)-2) as text) as kvc,
          convert(date, substring(OD, 2, len(OD)-2),4) as od,
          convert(date, substring(DO, 2, len(DO)-2),4) as do,
          cast(substring(dny, 2, len(dny)-2) as int) as dny,
          cast(substring(vdoba, 2, len(vdoba)-2) as int) as vdoba,
          cast(substring(odoba, 2, len(odoba)-2) as int) as odoba,
          cast(substring(vz, 2, len(vz)-2) as bigint) as vz,
          cast(substring(ID_ORG_AN, 2, len(ID_ORG_AN)-2) as bigint) as ID_ORG_AN,
          cast(substring(ixyear, 2, len(ixyear)-2) as int) as ixyear,
          cast(substring(typdokladu, 2, len(typdokladu)-2) as int) as typdokladu
into [VZ_upraveno]
from [VZ]
```

Jakmile máme k dispozici přetypované databáze na správné datové typy a jakmile jsou z hodnot všech proměnných odstraněné nepotřebné uvozovky, vypíšeme si několik prvních řádků zkoumaných tabulek. Už jen tento postup může vést k odhalení některých anomálií obsažených ve zpracovávaných datech.

```
select top 1000 * from [INEP] order by ID, rok
select top 1000 * from [ANOD] order by ID
select top 1000 * from [ANOD]
          where ID!=-1
          order by ID
select top 1000 * from [VZ] order by ID
GO
```

Pro další postup je užitečné vytvořit soubor, do kterého budou průběžně zapisovány odhalené problémy. Může se jednat o jednoduchou tabulku, která bude obsahovat minimálně následující sloupce

- Název postižené databáze,
- Stručný popis problému,
- Datum nálezu,
- Popis řešení,
- Průběžně aktualizovaný status,
- Datum poslední aktualizace.

Nálezy datové kvality					
Datum nálezu	Databáze	Popis problému	Návrh řešení	Status	Aktualizace
23.3.2015	STATMIN ANOD	Chybějící hlavička	Vložení hlavičky s názvy sloupců před první záznam	Zavřeno	23.3.2015
23.3.2015	STATMIN VZ, INEP	Všechny hodnoty jsou v uvozovkách	Odstranění uvozovek	Zavřeno	23.3.2015

23.3.2015	STATMIN ANOD, STATMIN VZ, INEP	Nevhodné datové typy	Přetypování hodnot ve všech databázích	Zavřeno	23.3.2015
…	…	…	…	…	…

Už při prvním pohledu na data jsme odhalili přítomnost několika anomálií, například

- v databázi STATMIN ANOD se vyskytují záznamy, které mají na pozici jednoznačného identifikátoru ve sloupci `ID` hodnotu -1
- hodnoty proměnných `datum_priznani` (databáze STATMIN ANOD) a `roknar` (databáze STATMIN VZ) se shodují, i když se jedná o různé informace
- databáze STATMIN VZ obsahuje jeden nebo více celých chybných řádků

V další části specifického příkladu jsme se zaměřili na propojení databází STATMIN ANOD a STATMIN VZ. Toto propojení je možné za pomoci jednoznačného identifikátoru, kterým je v případě obou databází proměnná `ID`. V prostředí SQL je možné tento proces realizovat použitím implementované funkce `left join`.

```
select A.*
      ,B.[datum_priznani]
      ,B.[pohlavi_anod]
      ,B.[vyse_primarniho]
      ,B.[typ_primarniho]
      ,B.[vyse_odvozeneho]
      ,B.[typ_odvozeneho]
into [VZ_ANOD_agregace]
from [VZ] A
          left join [ANOD] B on A.ID=B.ID
GO
```

Jakmile je propojení hotovo, je nutné zkontrolovat možné duplikace záznamů

- V databázi STATMIN ANOD je jednomu záznamu přiřazeno více řádků (první skript)
- V databázi STATMIN VZ je jednomu evidenčnímu listu přiřazeno více řádků (druhý skript)

```
select ID, count(*) as pocet_ID,
from [ANOD]
group by ID order by 1 desc

select ID_ELDP, count(*) as pocet_ELDP,
from [VZ]
group by ID_ELDP order by 1 desc
```

Pokud se v datech vyskytla některá z předchozích situací, je potřeba tuto informaci zapsat do souboru s odhalenými problémy a duplikované záznamy z databáze odstranit.

Nálezy datové kvality					
Datum nálezu	Databáze	Popis problému	Návrh řešení	Status	Aktualizace
23.3.2015	STATMIN ANOD	Chybějící hlavička	Vložení hlavičky s názvy sloupců před první záznam	Zavřeno	23.3.2015
23.3.2015	STATMIN VZ, INEP	Všechny hodnoty jsou v uvozovkách	Odstranění uvozovek	Zavřeno	23.3.2015
23.3.2015	STATMIN ANOD, STATMIN VZ, INEP	Nevhodné datové typy	Přetypování hodnot ve všech databázích	Zavřeno	23.3.2015
26.3.2015	STATMIN ANOD, STATMIN VZ, INEP	Duplicita záznamů	Odstranění duplicit	Nové	26.3.2015

Dále se můžeme podívat na propojení databází INEP a STATMIN VZ. Zde nastává komplikace, neboť obě databáze obsahují pro jednotlivá ID více řádků. Jednou z možností, jak se k tomuto problému postavit, je provést toto propojení po řadě pro jednotlivé konkrétní roky databáze INEP. Tento postup představuje relativně jednoduché řešení, které je zároveň přehlednější než jiné alternativy.

```
select *
into [INEP_cast]
from [INEP] where rok = zadany_rok
GO
```

I v případě tohoto propojení je nutné znovu zkontrolovat možné duplikace záznamů

- V databázi INEP je jednomu záznamu přiřazeno více řádků
- V databázi STATMIN VZ je jednomu evidenčnímu listu přiřazeno více řádků

Podobně jako v předchozím případě, vyskytla-li se ve zpracovávaných datech některá z předchozích situací, je potřeba opět tuto informaci zapsat do souboru s odhalenými problémy a duplikované záznamy z databáze odstranit.

Při procesu propojování těchto databází došlo k odhalení další anomálie, kterou je přítomnost „třetího" pohlaví v databázi STATMIN VZ. Tento problém mohl být způsoben skutečností, že databáze STATMIN VZ vzniká elektronizací přijatých evidenčních listů. K odstranění tohoto problému je vhodné kontaktovat majitele databáze a vyžádat si správné hodnoty odpovídající postiženému ID.
V rámci propojení jsme prověřili i shodu na následujících parametrech:

- rok narození – STATMIN ANOD (`rok_priznani`) a STATMIN VZ (`ROKNAR`),
- pohlaví – STATMIN ANOD (`MUŽ, ŽENA`) a STATMIN VZ (`1 = muž, 2 = žena`):

```
select * from VZ_ANOD_agregace where rok_priznani <> ROKNAR
select * from VZ_ANOD_agregace where (pohlavi = 1 and pohlavi_anod = 'ŽENA') or
(pohlavi = 2 and pohlavi_anod = 'MUŽ')
```

Tuto záležitost je potřeba zaznamenat do reportu sběru dat a řešit ji v dalších krocích analýzy datové kvality.

Při použití předchozího příkazu se vyskytují následující problémy

- Podezřelé hodnoty atributu `ROKNAR` v databázi STATMIN VZ
  - Výskyt hodnot menších než 1900 (např. rok 1886)
  - Výskyt budoucích hodnot (např. rok 2053)
    - Pravděpodobně způsobeno elektronizací evidenčních listů
    - Možné způsoby nápravy
      - Kontaktování vlastníka databáze
      - Doplnění hodnot z databáze INEP
      - Náhodné přiřazení hodnot na základě rozdělení

Nálezy datové kvality					
Datum nálezu	Databáze	Popis problému	Návrh řešení	Status	Aktualizace
23.3.2015	STATMIN ANOD	Chybějící hlavička	Vložení hlavičky s názvy sloupců před první záznam	Zavřeno	23.3.2015
23.3.2015	STATMIN VZ, INEP	Všechny hodnoty jsou v uvozovkách	Odstranění uvozovek	Zavřeno	23.3.2015
23.3.2015	STATMIN ANOD, STATMIN VZ, INEP	Nevhodné datové typy	Přetypování hodnot ve všech databázích	Zavřeno	23.3.2015

26.3.2015	STATMIN ANOD, STATMIN VZ, INEP	Duplicita záznamů	Odstranění duplicit	Otevřeno	26.3.2015
26.3.2015	STATMIN VZ	Podezřelé roky narození	Doplnění z INEP	Nové	26.3.2015

2.2 Popis dat

Pro další postup je nezbytné sesbíraná data pečlivě popsat. Náš popis rozdělíme z hlediska kvantitativního (objemová analýza) a kvalitativního (popis atributů).

2.2.1 Cíl

Cílem této fáze je:

- Prozkoumání vlastností nasbíraných dat
- Dokumentace prozkoumaných vlastností dat

2.2.2 Výstup

Výstupem této fáze je vytvoření reportu popisu dat. Tento report obsahuje:

- Detailní popis každého datového zdroje z hlediska
    - Kvantitativní části
        - Množství dat (počet tabulek, počet záznamů a polí v každé tabulce)
    - Kvalitativní části
        - Druhy hodnot (data mají různý formát, např. číslo, datum nebo logická hodnota)
- Návrh řešení
- Návrhy na zlepšení

2.2.3 Činnost

Prvním krokem popisu dat je podrobná dokumentace základních kroků, které jsou potřebné k jejich zisku

- Postup extrakce dat z dostupných zdrojů
- Popis přístupu k datovým zdrojům

Ke každému datovému zdroji, který byl získán v rámci sběru dat pro následné použití v mikrosimulačním modelu, je nezbytné pro další práci doplnit jeho detailní popis

- Seznam všech tabulek
- Seznam dalších použitých databázových objektů
- Vztahy mezi jednotlivými databázovými objekty

Pro každou z použitých tabulek je nutné popsat

- Aktuálnost obsažených informací
- Velikost (počty řádků a sloupců)
- Informace o prázdných polích (pro které atributy, počty)

Kromě kvantitativního hlediska je potřeba sebraná data popsat také z hlediska kvalitativního. Kvalitativní analýza zkoumá typy a hodnoty jednotlivých atributů. Pro všechny atributy obsažené v uvažovaných tabulkách je tedy nutné zajistit

- Typ atributu (číslo, znak, datum, logický operátor, …)

- Jednotky
- Význam
- Rozsah možných hodnot
- Korelace mezi jednotlivými atributy
- Popisné statistiky, např.
    - Pravděpodobnostní rozdělení
    - Rozptyl / směrodatnou odchylku
    - Šikmost, špičatost
    - Minimum, průměr, maximum
- Význam popisných statistik

2.2.4 Doporučení

Všechny klíčové vztahy musí být dostatečně dokumentovány. Jedná se o vzájemné vztahy

- Mezi uvažovanými datovými objekty
- Mezi jednotlivými atributy

Mezi jednotlivými datovými objekty mohou nastat následující vztahy

- Použité objekty vzájemně disjunktní
- Částečné překrytí objektů
    - Potřeba zjistit a dokumentovat počet překrytí
- Vznikla výběrem z jiných objektů
    - Potřeba identifikovat, o které objekty se jedná
    - Ověřit účel výběru
- Vznik sloučením několika jiných objektů
    - Potřeba identifikovat, o které objekty se jedná
    - Ověřit účel sloučení

Mezi atributy mohou nastat následující vztahy

- Duplicita atributů
    - Vícenásobný výskyt jednoho atributu by měl být zachycen už v základním reportu sběru dat
- Vznik sloučením více atributů
    - Nutné ověřit důvod sloučení

2.2.5 Specifický příklad

Pro další postup máme z předchozí části procesu posouzení datových vstupů a datové kvality k dispozici načtená a přetypovaná data ze všech používaných databází.

Pro účely specifického příkladu týkajícího se popisu dat vybereme databázi INEP, neboť na základě informací dostupných ze Studie proveditelnosti se jedná o klíčový datový zdroj, který by měl obsahovat většinu potřebných informací.

Z kvantitativního hlediska je nutné se při popisu datových zdrojů zaměřit na seznam všech atributů, jejich vzájemné vztahy (například součet pojištěné doby, nepojištěné doby a náhradní doby v pojištění v databázi INEP musí být roven číslu 365, resp. 366), použité jednotky, rozsahy atributů a jejich významy. Tyto informace jsou uvedeny v datových větách odpovídajících jednotlivým databázím.

V prvním kroku si tedy na základě dostupných příkladů datových vět vytvoříme přehledovou tabulku obsahující informace o všech proměnných, které jsou ve zkoumané databázi obsažené, která může mít například následující podobu.

Informace o datovém zdroji – databáze INEP					
Atribut	Příklad	Popis	Typ	Obsah	Poznámka
ID_OSOBY	1008637048	Jednoznačný identifikátor	Celé číslo	10-digit	-
ROK_NAROZENI	1973	Rok narození osoby	Celé číslo	4-digit	-
ROK	2004	Kalendářní rok záznamu	Celé číslo	4-digit	1 rok = 1 záznam
CIS_POHLAVI_ID	Z	Identifikátor pohlaví	Text	M-Z	-
DOBA_POJISTENA	128	Pojištěná doba v roce	Celé číslo	0-365/366	-
…	…	…	…	…	…

V druhém kroku vytvoříme pro každý datový zdroj tabulku, která bude obsahovat základní popisné informace, včetně:

- Aktuálnosti obsažených informací, např.
  - Poslední dostupná data
  - Datum vyplnění popisné tabulky
- Základních informace o velikosti, např.:
  - Počty řádků
  - Počty unikátních ID
- Počtu sloupců v tabulce

Základní informace o velikosti z databáze INEP je možné získat použitím následujícího skriptu:

```
select count(*) as pocet_radku,
       count(distinct ID) as pocet_ID
from [INEP]
```

Příklad vytvoření přehledové tabulky ukažme opět na databázi INEP, která je v současnosti rozdělena do několika dílčích souborů, z nichž jeden vybereme pro účely tohoto specifického příkladu. Taková tabulka může mít například následující podobu.

Základní informace – dílčí soubor databáze INEP	
Popis tabulky	INEP
Datum vyplnění	23. 3. 2015
Poslední dostupná data	2014
Počet řádků	13 212 158
Počet unikátních ID	499 991
Počet sloupců	30

Kromě kvantitativního hlediska popisu datových zdrojů nás zajímá také hledisko kvalitativní. Pro tento účel je dobré zkonstruovat základní popisné statistiky, ukažme si toto opět na příkladu databáze INEP, konkrétně se zaměřme na pojištěnou dobu v roce. Abychom mohli takové úpravy provádět, nejprve databázi provedeme agregaci příslušného datového zdroje.

```
IF object_id('tempdb.dbo.##DB_temp_INEP_uprava') is NOT NULL DROP TABLE
##DB_temp_INEP_uprava
select ID, min(rok) as init_year, max(rok) as end_year
          , count(distinct rok) as total_years
          , sum(doba_pojistena) as total_doba_pojistena
into ##DB_temp_INEP_uprava
from INEP
group by ID order by init_year desc
GO
```

Existují více možností, jak odpovídající popisné statistiky vytvořit - např. výpočet statistik pro populaci rozdělenou do skupin podle počtu průřezových let strávených v databázi (na základě veličiny rok). Takový postup je ukázán v následujícím skriptu.

```
select total_years
     , min(total_doba_pojistena) as min_doba_pojistena
     , avg(total_doba_pojistena) as avg_doba_pojistena
     , max(total_doba_pojistena) as max_doba_pojistena
from ##DB_temp_INEP_uprava
group by total_years
order by 1 asc
```

Popisné statistiky – dílčí soubor databáze INEP					
Atribut	Počet let	Minimum	Maximum	Průměr	Jednotky
DOBA_POJISTENA	…	…	…	…	…
	42	0	15 341	11 297	Dny
	43	0	15 706	11 735	Dny
	44	0	16 039	12 187	Dny
	…	…	…	…	…

Je nicméně zřejmé, že tyto statistiky nemají z hlediska popisu dat příliš velký význam. Z tohoto důvodu je lepší použít alternativní postup, kdy budeme zkoumat doby pojištění normalizované na úroveň jednoho roku pojištění (viz následující skript).

```
select min(total_doba_pojistena/total_years) as min_doba_pojistena_normalized
     , avg(total_doba_pojistena/total_years) as avg_doba_pojistena_normalized
     , max(total_doba_pojistena/total_years) as max_doba_pojistena_normalized
from ##DB_temp_INEP_uprava
order by 1 asc
```

Popisné statistiky – dílčí soubor databáze INEP				
Atribut	Minimum	Maximum	Průměr	Jednotky
DOBA_POJISTENA (normalizovaná)	0	366	188	Dny

Získaná statistika již přináší přidanou hodnotu z pohledu datové kvality, jelikož vidíme, že zkoumaný dílčí soubor databáze INEP neobsahuje odlehlá pozorování pojištěné doby.

Pro výpočet korelací mezi jednotlivými atributy, stejně jako jejich šikmostí a špičatostí, je nutné použít statistický software (doporučujeme např. volně dostupný program *R*) určený pro takové účely. Na vzorových datech jsme však tuto nutnost neidentifikovali. Pro ostatní datové zdroje mohou být další operace vhodné.

Jak je z příkladu vidět, je pro přípravu popisných statistik potřebné důsledně zvážit jejich význam, jak bylo demonstrováno na předchozích dvou přístupech k jejich výpočtu pro proměnnou představující pojištěnou dobu v roce. Zvolením vhodné statistiky si zjednodušíme práci v dalších krocích.

2.3 Průzkum dat

Tato dílčí část týkající se fáze datových vstupů a datové kvality slouží k prozkoumání dat za pomoci jejich vizualizace. Slouží k formulaci základních hypotéz a utvoření představy o transformaci dat, která bude následovat v další kapitole.

2.3.1 Cíl

Cílem této fáze je:

- Vizualizace dat
- Formulace prvotních hypotéz
- Objevení prvních nálezů v sesbíraných datech

2.3.2 Výstup

Výstupem této fáze je vytvoření reportu průzkumu dat. Tento report obsahuje:

- Vizualizace dat (grafy a přehledové tabulky)
    - Histogramy odchylek od očekávaných hodnot
    - Sledování vybraných kvantilů, například
        - Velmi vysoké hodnoty (překračující 99% kvantil)
        - Velmi nízké hodnoty (nedosahující ani 5% kvantilu)
    - Intervaly spolehlivosti odhadů
- Prvotní nálezy
- První hypotézy a jejich možný dopad na zbytek projektu
- Návrh řešení
- Návrhy na zlepšení

2.3.3 Činnost

Z hlediska následného modelování je třeba detailně prozkoumat vlastnosti všech atributů, které mají potenciální význam. Je důležité se zaměřit na následující vlastnosti

- Popisné statistiky
- Závislost mezi atributy
- Popisné statistiky podskupin
    - Marginální rozdělení, např.
        - Vyměřovacího základu
        - Počtu dní zaměstnání
    - Průměr, maximum, minimum

Pokud v datech očekáváme jakékoliv pravidelnosti, je důležité v reportu průzkumu dat shrnout následující informace

- Identifikace pravidelností
- Původ těchto regularit
- Metody jejich odhalení
    - Grafické metody

    - Statistické testy
- Výsledky ověřování
    - (Ne)potvrzení očekávaných pravidelností
    - Jakékoliv další důležité závěry
- Dopad na následnou transformaci a očišťování dat

2.3.4 Doporučení

Průzkum dat může pomoci odhalit chyby, které byly způsobené manuálním vkládáním dat. Podle našich informací je např. databáze STATMIN VZ vytvořena na základě elektronizovaných ELDP, které zaměstnavatel za své zaměstnance odevzdává v papírové podobě. V tomto bodě mohou vzniknout chyby, které se mohou při průzkumu dat projevit jako

- Odlehlé body na grafech
- Extrémní hodnoty v popisných tabulkách

2.3.5 Specifický příklad

Kvůli způsobu, jakým byla podle našich informací databáze STATMIN VZ vytvořena (tj. elektronizací papírových ELDP, které zaměstnavatel odevzdává za své zaměstnance), očekáváme právě v ní největší koncentraci chybných pozorování. Grafická analýza může pomoci odhalit

- Chyby v datech
- Odlehlá pozorování
- Přítomnost pravidelností v jednotlivých skupinách

V prvním kroku se podíváme nejprve na počty roků narození podle jednotlivých pohlaví.

```
select roknar, sum(case when pohlavi=1 then 1 else 0 end) as C_Muzi
             , sum(case when pohlavi=2 then 1 else 0 end) as C_Zeny
from [VZ]
group by roknar
order by roknar
```

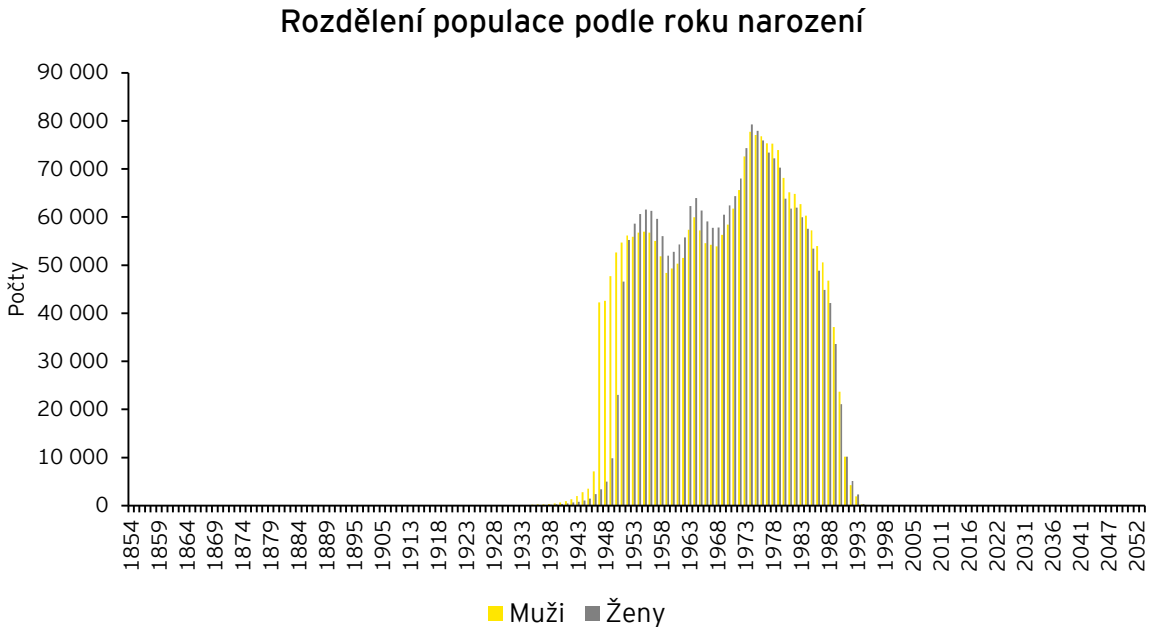
Po použití předchozího příkazu v prostředí SQL získáme tabulku s četnostmi jednotlivých roků narození postupně pro obě pohlaví, která jsou v databázi obsažena. Na první pohled je vidět, že v datech se objevují chyby

- roky narození z budoucnosti (např. 2053)
- roky narození z dávné minulosti (např. 1886 nebo 1854)
- rok narození není v datech vůbec uveden

Pro snadné odhalení odlehlých pozorování je dobré získané tabulky importovat do jiného programu, který je vhodnější pro grafickou úpravu dat. Pro snadnou manipulaci s daty doporučujeme například

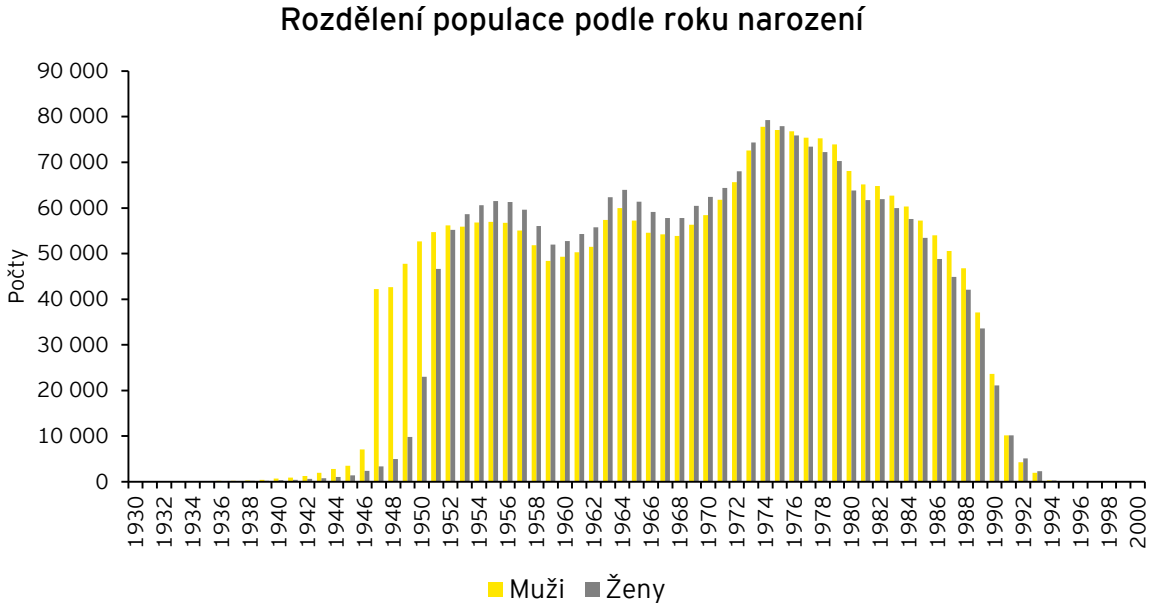
- Microsoft Excel
- R

Pro účely tohoto příkladu jsme pro jednoduchost použili MS Excel. V takovém softwaru už potom snadno sestrojíme histogram nebo jiný vhodný graf, který pomůže odhalit přítomnost odlehlých pozorování.



Obrázek 3: Histogram let narození v závislosti na pohlaví

Pokud odstraníme odlehlá pozorování, v tomto konkrétním případě zvolme např. vynechání hodnot menších než 1930 a větších než 2000, můžeme lépe pozorovat distribuci dat (viz následující obrázek).



Obrázek 4: Histogram let narození v závislosti na pohlaví po odstranění odlehlých hodnot

Pokud v datech odhalíme přítomnost chybných informací, je nutné pro další postup tyto nálezy zapsat do dříve vytvořeného souboru, který obsahuje všechny dosud objevené problémy datové kvality. Ukažme nyní možná řešení hodnot roků narození, které jsou buď starší než 1900 nebo novější než rok současný. V takových případech se nabízejí následující možnosti

- Kontaktování zdroje dat za účelem aktualizace chybných hodnot
- Posun do dvacátého století, například 1854 na 1954

- Porovnání s jiným rokem databáze
- Doplnění chybějících informací z jiné databáze (např. INEP)

Z nám dostupných informací považujeme poslední dvě možnosti za nejlepší, neboť při jejich použití nedochází ke ztrátě informací a na základě dostupných pozorování je možné odstranit výraznou část chybných nebo chybějících hodnot.

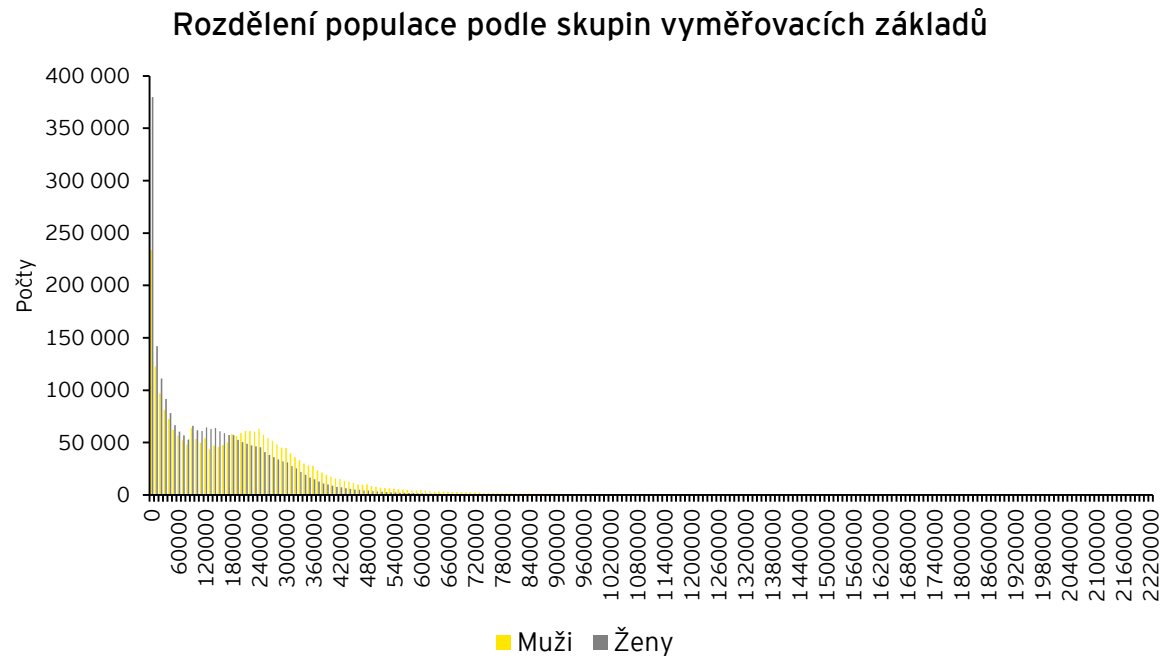
Následující skript ukazuje, jak je možné získat roky narození osob v databázi STATMIN VZ z databáze INEP pro osoby, jejichž rok původní rok narození je nižší než 1900 nebo není v databázi STATMIN VZ vůbec uveden:

```
select A.*,B.rok_narozeni from [VZ] A
          left join [INEP] B on A.ID=B.ID and B.ID is not null
where roknar<1900 and roknar!=0
```

Kromě roku narození se můžeme podívat na rozdělení vyměřovacích základů podle obou pohlaví. Následující skript je příkladem, jak takové rozdělení může být provedeno.

```
select vz-vz%10000 as vz_grouped
          , sum(case when pohlavi=1 then 1 else 0 end) as C_Muzi
          , sum(case when pohlavi=2 then 1 else 0 end) as C_Zeny
from [VZ_final]
group by vz-vz%10000
order by vz-vz%10000
```

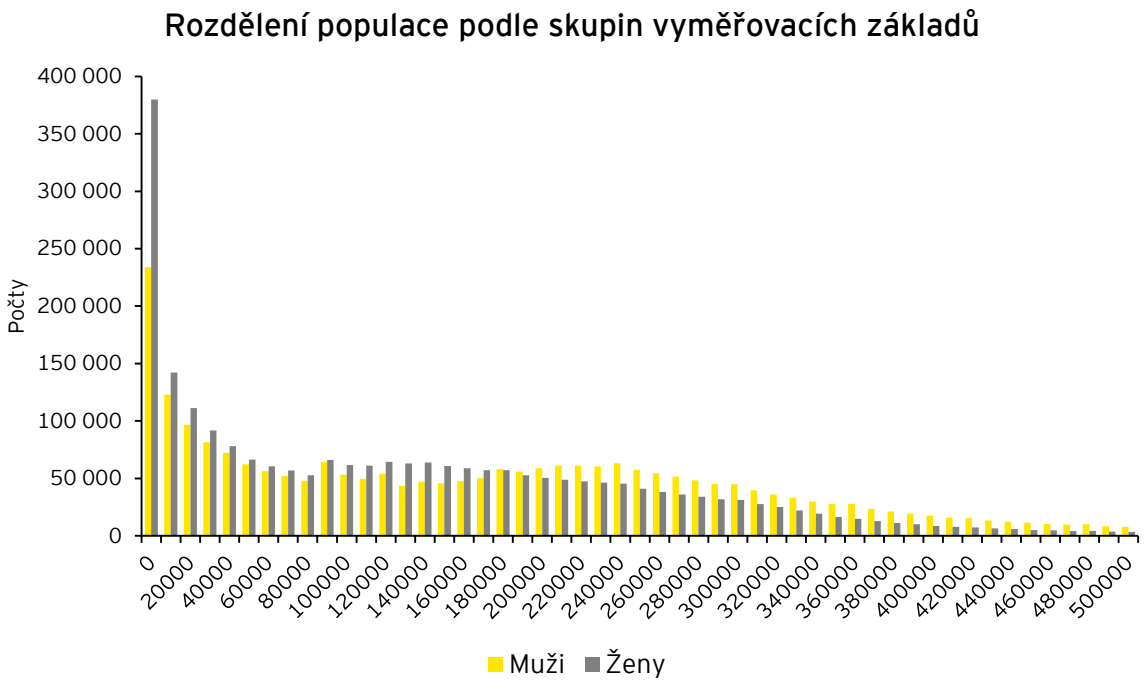
Při znázornění vyměřovacích základů je dobré spojit tuto veličinu do skupin po desetitisících korun, aby bylo možné její rozdělení lépe pozorovat (viz předchozí skript). Výsledky znázorníme na následujícím grafu.



Obrázek 5: Histogram vyměřovacích základů v závislosti na pohlaví

Výsledky na předchozím obrázku jsou velmi zkreslené kvůli odlehlým pozorováním ve skupinách s velmi vysokým vyměřovacím základem. Z tohoto důvodu sestrojíme tento graf ještě jednou, tentokrát ale pouze pro nižší skupiny (např. do 500 000 korun), aby lépe vyniklo rozdělení

populace.



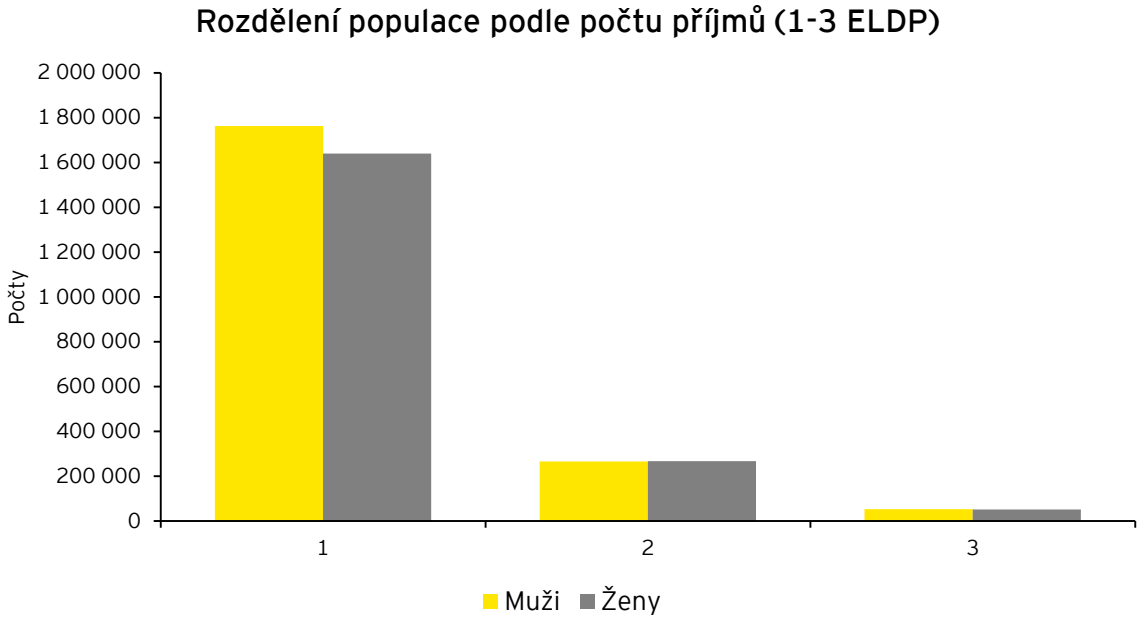
Obrázek 6: Histogram vyměřovacích základů v závislosti na pohlaví po odstranění odlehlých hodnot

Z hlediska datové kvality databáze STATMIN VZ je důležité se podívat také na rozdělení počtu příjmů podle např. pohlaví. Tyto počty jsou reprezentovány počtem různých evidenčních listů pro každé jednoznačné ID. Budeme tedy zkoumat, kolik je každému ID v databázi přiřazeno různých hodnot proměnné `ID_ELDP`, která reprezentuje právě identifikátor evidenčního listu, jak je ukázáno v následujícím skriptu:

```
IF object_id('tempdb.dbo.##DB_temp_VZ_ELDP') is NOT NULL DROP TABLE ##DB_temp_VZ_ELDP
select ID, pohlavi, count(distinct ID_ELDP) as C_ELDP
into ##DB_temp_VZ_ELDP
from [VZ]
group by ID, pohlavi
GO
```

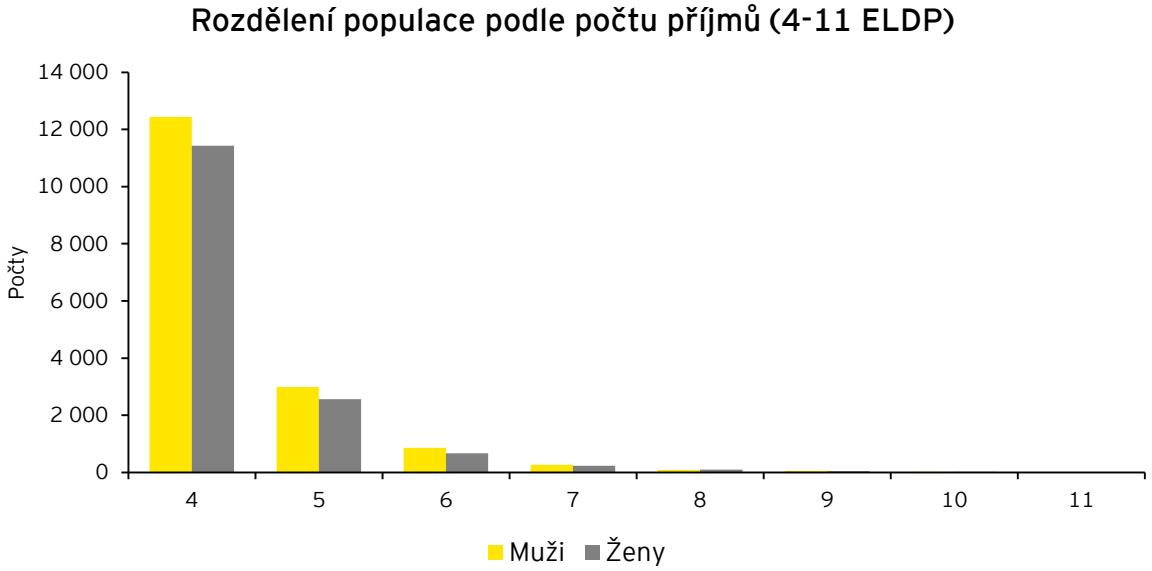
Po použití předchozího příkazu je vidět, že ve většině případů odpovídá jednomu ID právě jeden evidenční list. Vyskytují se však osoby, které měly během jednoho roku dokonce 20 a více zaměstnání. V takovém případě je nutné rozhodnout, jestli se jedná o chyby v datech nebo o odlehlá pozorování. Za účelem získání této informace je potřeba kontaktovat majitele databáze.

Při grafickém znázornění rozdělení populace podle počtu příjmů nastává situace, kdy většina pozorování spadá do skupin majících 1 až 3 evidenční listy a ostatní skupiny nejsou na společném grafu téměř vidět. Pro větší názornost tedy ukážeme celkové rozdělení na dvou separátních panelů.



Obrázek 7: Histogram počtu příjmů v závislosti na pohlaví pro tři největší skupiny

Z předchozího obrázku lze vidět, že valná většina populace měla během zkoumaného roku pouze jeden pracovní poměr. Ve vzorku se ale vyskytují také odlehlá pozorování (např. jedinci s více než jedenácti evidenčními listy). Jedná se pouze ale o ojedinělé případy, proto ukážeme pouze rozdělení pro skupiny se 4 až 11 evidenčními listy.



Obrázek 8: Histogram počtu příjmů v závislosti na pohlaví pro menší skupiny

2.4 Verifikace dat

Data jsou zřídka kdy perfektní a často obsahují chybějící hodnoty a mnohé další druhy nesrovnalostí, které znemožňují následnou analýzu. Z tohoto důvodu je nutné před samotným modelováním provést analýzu kvality dostupných dat, aby bylo možné provést potřebné kroky k nápravě problémů. Z tohoto důvodu je fáze verifikace dat klíčová, protože teprve během ní dochází

k hodnocení datové kvality. Na základě této fáze získáme ucelený přehled o tom, zda jsou data v pořádku, resp. zda je nutné nějaké případné problémy řešit. Možná řešení existují dvojího typu:

- Kontaktování vlastníka dat pro
    - Objasnění konkrétního problému
    - Zjednání nápravy přímo na úrovni datového zdroje.
- Samostatné vyřešení problému způsoby, které budou popsány v další fázi

2.4.1 Cíl

Cílem této fáze je:

- Prověření kvality dostupných dat

2.4.2 Výstup

Výstupem této fáze je vytvoření reportu kvality dat. Tento report obsahuje:

- Výsledky verifikace kvality dat
- Seznam objevených problémů
- Možná řešení těchto problémů
- Návrhy na zlepšení

2.4.3 Činnost

Při prověřování kvality je nutné zkontrolovat, jestli se v dostupných datech nevyskytují následující komplikace

- Chybějící hodnoty
    - Pohlaví
        - Kromě hodnot muž a žena se může vyskytovat „neurčené“ pohlaví
- Chyby v datech
    - Typografické chyby, které vznikly při zadávání dat
    - Možné především u dat z databáze STATMIN VZ
        - Vstupy databáze jsou elektronizované papírové ELDP
- Nesprávné jednotky
    - Částka v tisících korun místo v korunách
    - Doba v měsících místo ve dnech
- Nesrovnalosti v kódování
    - Vznik při použití nestandardních jednotek nejednotným značením
        - Pohlaví (M-muž-0, Ž-žena-1)
        - Typ primárního důchodu
        - Typ odvozeného důchodu

Nejčastějším problémem jsou chybějící hodnoty v datech. Pokud tato možnost nastane, je nezbytné pro další práci s daty udělat následující:

- Identifikovat chybějící atributy
- Identifikovat prázdná pole
- Zjistit význam chybějících dat
- Zjistit důvod nepřítomnosti
- Navrhnout možné řešení
    - Data jsou pro další postup zanedbatelná
        - Dokumentovat informace o chybějících datech
    - Data jsou pro další postup nezbytná

        - Hledat data v jiné databázi
        - Hledat atributy s podobným významem
        - Prověřit možnost získání informace kombinací známých atributů

Další důležité kontrolní body, které je třeba ověřit, než se s daty bude dále pracovat

- Přítomnost všech možných hodnot, kterých mohou data nabývat
- Přítomnost jednoznačného identifikačního klíče u každého záznamu
    - Pokud není k dispozici, je možné přiřadit údaje náhodně
        - Na základě pravděpodobnostního rozdělení
        - Například přiřazení
            - Vzdělání z databáze obyvatelstva ČSÚ
            - Ekonomický stav z Výběrového šetření pracovních sil ČSÚ
- Hodnoty v polích odpovídají významu atributů
- Pravopis a formát hodnot
    - Počáteční písmeno je někdy malé a někdy velké
    - Přítomnost diakritiky
- Věrohodnost hodnot
    - Všechna pole obsahují podobnou hodnotu
- Extrémní hodnoty v datech

Při použití více zdrojů mezi nimi mohou vyniknout nekonzistence a šum

- Kontrola konzistence mezi zdroji
- Kontrola šumu
    - Plán postupu při výskytu šumu
    - Odhalení druhu šumu
    - Detekce postižených atributů

2.4.4 Doporučení

Je dobré využít grafických nástrojů k odhalení nesrovnalostí v datech

- Histogramy
- Diagramy
- Grafy

2.4.5 Specifický příklad

Při verifikaci dostupných dat se zaměřme nejprve na ověření datové kvality databáze STATMIN ANOD, která v současné době obsahuje přibližně 5,7 milionu záznamů. V prvním kroku se podívejme na to, jestli se v databázi vyskytují duplicity jednoznačného identifikátoru ID:

```
select count(*), ID
from [ANOD]
group by ID ORDER BY 1 desc
```

Po spuštění předchozího příkazu vidíme, že se v databázi vyskytují ID se zvláštní hodnotou -1. Celkem se v datech objevuje 490 záznamů s touto hodnotou.
Podle popisu dat je v současné době databáze STATMIN ANOD dostupná za dva roky (2011 a 2012), proto by každé jednoznačné ID mělo mít přiřazeno dva záznamy. Při bližším zkoumání je však vidět, že se v datech vyskytují anomálie, například

- jednomu ID odpovídají 2 dvojice stejných důchodů

- jednomu ID odpovídají 4 různé důchody

Původ těchto pozorování je třeba řešit s vlastníkem databáze. Pokud by se jednalo o chybná data, existují dvě možnosti, jak dále postupovat:

- Tato pozorování z databáze vyloučit
    - Pokud by se však nejednalo o chybu, dojde ke ztrátě informací
- Náhodně jim přiřadit unikátní ID

Tento, stejně jako všechny další odhalené komplikace s datovou kvalitou, je pro další postup potřeba průběžně zapisovat do dříve vytvořené tabulky a udržovat ji stále aktualizovanou.

Nálezy datové kvality					
Datum nálezu	Databáze	Popis problému	Návrh řešení	Status	Aktualizace
23.3.2015	STATMIN ANOD	Chybějící hlavička	Vložení hlavičky s názvy sloupců před první záznam	Zavřeno	23.3.2015
23.3.2015	STATMIN VZ, INEP	Všechny hodnoty jsou v uvozovkách	Odstranění uvozovek	Zavřeno	23.3.2015
23.3.2015	STATMIN ANOD, STATMIN VZ, INEP	Nevhodné datové typy	Přetypování hodnot ve všech databázích	Zavřeno	23.3.2015
26.3.2015	STATMIN ANOD, STATMIN VZ, INEP	Duplicita záznamů	Odstranění duplicit	Otevřeno	26.3.2015
26.3.2015	STATMIN VZ	Podezřelé roky narození	Doplnění z INEP	Otevřeno	26.3.2015
30.3.2015	STATMIN ANOD	Více než dva záznamy u každého ID	Konzultace s vlastníkem databáze	Nové	30.3.2015

Jakmile je ošetřen jednoznačný identifikátor osob představovaný veličinou ID, je potřeba se podívat také na další proměnné, které by mohly představovat problém z hlediska datové kvality. Zaměřme se nyní na pohlaví, které by mělo nabývat hodnot 'MUŽ' a 'ŽENA'.

```
select count(*), pohlavi_anod
from [ANOD]
group by pohlavi_anod ORDER BY 1 desc
```

Při bližším pohledu na výsledky získané použitím předchozího příkazu vidíme, že v datech jsou větším podílem zastoupeny ženy a že databáze obsahuje v současnosti 378 hodnot s chybějícím údajem o pohlaví. Tento nález je nutné zapsat do reportu kvality dat a pro další postup se nabízí následující možnosti, jak s těmito pozorováními naložit

- Odstranění chybějících pozorování
    - Dojde ke ztrátě části informací pro budoucí použití
- Náhodné přiřazení pohlaví
    - Musí být zachováno pravděpodobnostní rozdělení pohlaví
- Doplnění chybějící informace
    - Propojením s jinou dostupnou databází (nejlépe INEP)
    - Kontaktováním majitele databáze

Nálezy datové kvality					
Datum nálezu	Databáze	Popis problému	Návrh řešení	Status	Aktualizace
23.3.2015	STATMIN ANOD	Chybějící hlavička	Vložení hlavičky s názvy sloupců před první záznam	Zavřeno	23.3.2015
23.3.2015	STATMIN VZ, INEP	Všechny hodnoty jsou v uvozovkách	Odstranění uvozovek	Zavřeno	23.3.2015
23.3.2015	STATMIN ANOD, STATMIN VZ,	Nevhodné datové typy	Přetypování hodnot ve všech databázích	Zavřeno	23.3.2015

	INEP				
26.3.2015	STATMIN ANOD, STATMIN VZ, INEP	Duplicita záznamů	Odstranění duplicit	Otevřeno	26.3.2015
26.3.2015	STATMIN VZ	Podezřelé roky narození	Doplnění z INEP	Otevřeno	26.3.2015
30.3.2015	STATMIN ANOD	Více než dva záznamy u každého ID	Konzultace s vlastníkem databáze	Nové	30.3.2015
30.3.2015	STATMIN ANOD	Chybějící pohlaví	Propojení s INEP	Nové	30.3.2015

Dále jsme se při zkoumání databáze STATMIN ANOD zaměřili také na informace o primárních a odvozených důchodech. Podívejme se nejprve na primární důchody:

```
select count(*), case when isnull(vyse_primarniho,0) = 0 then 'ERROR'
                      else 'OK' end
from [ANOD]
group by case when isnull(vyse_primarniho,0) = 0
              then 'ERROR' else 'OK' end
```

Po použití předchozího skriptu vidíme, že 86 pozorování má nulový primární důchod. Tuto skutečnost je sice dobré poznamenat, nemusí ale představovat problém z hlediska datové kvality, neboť tato pozorování mohou mít pouze odvozený důchod. Tuto teorii je dobré ověřit:

```
select count(*),
       case when isnull(vyse_primarniho,0) + isnull(vyse_odvozeneho,0) = 0 then
'ERROR' else 'OK' end
from [ANOD]
group by case when isnull(vyse_primarniho,0) + isnull(vyse_odvozeneho,0) = 0 then
'ERROR' else 'OK' end
```

Ukázalo se, že pouze jeden záznam v současnosti nemá přiřazen ani primární ani odvozený důchod. Je možné o tuto informaci požádat majitele databáze, protože se ale jedná jen o jedno pozorování, jeho vyloučení by nemělo mít velký vliv na celkový výsledek.

Nálezy datové kvality					
Datum nálezu	Databáze	Popis problému	Návrh řešení	Status	Aktualizace
23.3.2015	STATMIN ANOD	Chybějící hlavička	Vložení hlavičky s názvy sloupců před první záznam	Zavřeno	23.3.2015
23.3.2015	STATMIN VZ, INEP	Všechny hodnoty jsou v uvozovkách	Odstranění uvozovek	Zavřeno	23.3.2015
23.3.2015	STATMIN ANOD, STATMIN VZ, INEP	Nevhodné datové typy	Přetypování hodnot ve všech databázích	Zavřeno	23.3.2015
26.3.2015	STATMIN ANOD, STATMIN VZ, INEP	Duplicita záznamů	Odstranění duplicit	Otevřeno	26.3.2015
26.3.2015	STATMIN VZ	Podezřelé roky narození	Doplnění z INEP	Otevřeno	26.3.2015
30.3.2015	STATMIN ANOD	Více než dva záznamy u každého ID	Konzultace s vlastníkem databáze	Nové	30.3.2015
30.3.2015	STATMIN ANOD	Chybějící pohlaví	Propojení s INEP	Nové	30.3.2015
30.3.2015	STATMIN ANOD	Chybějící informace o důchodu u jednoho ID	Odstranění záznamu	Nové	30.3.2015

Další problém může nastat, pokud by vyplácený důchod neměl vyplněný typ. To je možné ověřit například následujícím způsobem (příklad uvedený platí pro odvozený důchod, pro primární je postup analogický)

```
select count(*), case when isnull(vyse_odvozeneho,0) > 0 and typ_odvozeneho = '' then
'ERROR' else 'OK' end
from [ANOD]
group by case when isnull(vyse_odvozeneho,0) > 0 and typ_odvozeneho = '' then 'ERROR'
else 'OK' end
```

Podobnou verifikaci datové kvality jako v případě databáze STATMIN ANOD je nutné provést také pro ostatní dostupné databáze. Zaměřme se nyní na analýzu STATMIN VZ. Prověření duplicity jednoznačného identifikátoru ID provedeme stejným způsobem jako v předchozím případě, proto tento postup nebudeme znovu popisovat. Komplikace nastává při analýze pohlaví.

```
select count(*), pohlavi
from [VZ]
group by POHLAVI ORDER BY 1 desc
```

Předchozí skript odhalil, že proměnná `pohlavi` nabývá kromě hodnot 1 (představující muže) a 2 (představující ženu) také v 1617 případech hodnotu 0. Tento nález je potřeba zapsat do reportu verifikace datové kvality a ověřit možnost získání správné informace, například

- Kontrolou výsledků s jiným rokem databáze STATMIN VZ
- Náhodné přiřazení hodnoty
- Kontrolou výsledků s databází INEP pro jeden konkrétní rok, např. následujícím způsobem:

```
select *
from [VZ] A left join [INEP] B on A.ID=B.ID and B.rok = 2009
where A.pohlavi = 0 and B.ID is not null
```

Nálezy datové kvality

Datum nálezu	Databáze	Popis problému	Návrh řešení	Status	Aktualizace
23.3.2015	STATMIN ANOD	Chybějící hlavička	Vložení hlavičky s názvy sloupců před první záznam	Zavřeno	23.3.2015
23.3.2015	STATMIN VZ, INEP	Všechny hodnoty jsou v uvozovkách	Odstranění uvozovek	Zavřeno	23.3.2015
23.3.2015	STATMIN ANOD, STATMIN VZ, INEP	Nevhodné datové typy	Přetypování hodnot ve všech databázích	Zavřeno	23.3.2015
26.3.2015	STATMIN ANOD, STATMIN VZ, INEP	Duplicita záznamů	Odstranění duplicit	Otevřeno	26.3.2015
26.3.2015	STATMIN VZ	Podezřelé roky narození	Doplnění z INEP	Otevřeno	26.3.2015
30.3.2015	STATMIN ANOD	Více než dva záznamy u každého ID	Konzultace s vlastníkem databáze	Nové	30.3.2015
30.3.2015	STATMIN ANOD	Chybějící pohlaví	Propojení s INEP	Nové	30.3.2015
30.3.2015	STATMIN ANOD	Chybějící informace o důchodu u jednoho ID	Odstranění záznamu	Nové	30.3.2015
30.3.2015	STATMIN VZ	Přítomnost „třetího“ pohlaví	Náhodné přiřazení	Nové	30.3.2015

Zaměřme se nyní na odpracovanou dobu jednotlivých modelpointů. Pro tento účel je nejprve nutné ověřit, jestli je pro všechny řádky databáze doplněna informace o době nástupu do zaměstnání (proměnná `OD`) a době výstupu ze zaměstnání (proměnná `DO`), například následujícím způsobem

```
select count(*), case when OD is NULL then 'ERROR' else 'OK' end
from [VZ] group by case when OD is NULL then 'ERROR' else 'OK' end

select count(*), case when DO is NULL then 'ERROR' else 'OK' end
from [VZ] group by case when DO is NULL then 'ERROR' else 'OK' end
```

Pokud takovým způsobem odhalíme, že databáze obsahuje řádky, které nemají uvedený začátek a konec zaměstnání, je nutné takové záznamy v datech najít a buď je pro další postup z databáze odstranit, nebo kontaktovat majitele a chybějící informaci doplnit.

```
select *
from [VZ] where OD IS NULL or DO IS NULL
```

Nálezy datové kvality					
Datum nálezu	Databáze	Popis problému	Návrh řešení	Status	Aktualizace
23.3.2015	STATMIN ANOD	Chybějící hlavička	Vložení hlavičky s názvy sloupců před první záznam	Zavřeno	23.3.2015
23.3.2015	STATMIN VZ, INEP	Všechny hodnoty jsou v uvozovkách	Odstranění uvozovek	Zavřeno	23.3.2015
23.3.2015	STATMIN ANOD, STATMIN VZ, INEP	Nevhodné datové typy	Přetypování hodnot ve všech databázích	Zavřeno	23.3.2015
26.3.2015	STATMIN ANOD, STATMIN VZ, INEP	Duplicita záznamů	Odstranění duplicit	Otevřeno	26.3.2015
26.3.2015	STATMIN VZ	Podezřelé roky narození	Doplnění z INEP	Otevřeno	26.3.2015
30.3.2015	STATMIN ANOD	Více než dva záznamy u každého ID	Konzultace s vlastníkem databáze	Nové	30.3.2015
30.3.2015	STATMIN ANOD	Chybějící pohlaví	Propojení s INEP	Nové	30.3.2015
30.3.2015	STATMIN ANOD	Chybějící informace o důchodu u jednoho ID	Odstranění záznamu	Nové	30.3.2015
30.3.2015	STATMIN VZ	Přítomnost „třetího“ pohlaví	Náhodné přiřazení	Nové	30.3.2015
30.3.2015	STATMIN VZ	Některé záznamy nemají počátek/konec zaměstnání	Konzultace s vlastníkem databáze	Nové	30.3.2015

Posledním objektem, jehož datovou kvalitu budeme v této části verifikovat, je databáze INEP. Prvním krokem bude opět kontrola duplicit u jednoznačného identifikátoru ID. Už v této části narazíme na chybu, kterou je nejednoznačnost identifikačního čísla osoby. Tato skutečnost může být způsobena dvěma hlavními důvody.
Prvním důvodem vedoucím k duplicitě identifikátorů je možnost, že může dojít ke zdvojení rodných čísel. Tedy v datech může nastat situace, kdy dojde k záměně člověka narozeného po roce 2000 s jedincem, který se narodil před rokem 1954. Rodná čísla sice nejsou obsažena v dostupných databázích, můžeme se však pokusit tento problém odstranit použitím roku narození a získat tak alespoň nějakou užitečnou informaci.

```
select count(*), ID, rok_narozeni
from [INEP]
group by ID, rok_narozeni order by 1 desc
```

Takový postup zamezí problémům souvisejícím s duplicitou rodných čísel (viz výše uvedený skript).
Druhým důvodem duplicity ID je skutečnost, že se jedna osoba skutečně vyskytuje v daném průřezovém roce více než jednou.

```
select count(*), ID, rok_narozeni, rok
from [INEP]
group by ID, rok_narozeni, rok order by 1 desc rok desc
```

Z tohoto důvodu je dobré při analýze dat nespoléhat pouze na ID a rok narození, ale použít unikátní kombinaci ID, roku narození a průřezového roku, která zamezí výskytu tohoto problému (viz výše uvedený skript).

Nálezy datové kvality					
Datum nálezu	Databáze	Popis problému	Návrh řešení	Status	Aktualizace
23.3.2015	STATMIN ANOD	Chybějící hlavička	Vložení hlavičky s názvy sloupců před první záznam	Zavřeno	23.3.2015
23.3.2015	STATMIN VZ, INEP	Všechny hodnoty jsou v uvozovkách	Odstranění uvozovek	Zavřeno	23.3.2015
23.3.2015	STATMIN ANOD, STATMIN VZ, INEP	Nevhodné datové typy	Přetypování hodnot ve všech databázích	Zavřeno	23.3.2015
26.3.2015	STATMIN ANOD, STATMIN VZ, INEP	Duplicita záznamů	Odstranění duplicit	Otevřeno	26.3.2015
26.3.2015	STATMIN VZ	Podezřelé roky narození	Doplnění z INEP	Otevřeno	26.3.2015
30.3.2015	STATMIN ANOD	Více než dva záznamy u každého ID	Konzultace s vlastníkem databáze	Nové	30.3.2015
30.3.2015	STATMIN ANOD	Chybějící pohlaví	Propojení s INEP	Nové	30.3.2015
30.3.2015	STATMIN ANOD	Chybějící informace o důchodu u jednoho ID	Odstranění záznamu	Nové	30.3.2015
30.3.2015	STATMIN VZ	Přítomnost „třetího“ pohlaví	Náhodné přiřazení	Nové	30.3.2015
30.3.2015	STATMIN VZ	Některé záznamy nemají počátek/konec zaměstnání	Konzultace s vlastníkem databáze	Nové	30.3.2015
30.3.2015	INEP	Duplicita záznamů	Použití jednoznačné kombinace	Nové	30.3.2015

Jakmile je ověřeno, že identifikátory ID jsou skutečně jednoznačné, je dobré zaměřit se na dobu, kterou každý člověk strávil během jednotlivých let v pojištění. Tato doba je označena jako proměnná `doba_pojistena`. Podívejme se nejprve, jestli data z tohoto hlediska obsahují nějaká odlehlá pozorování. Taková situace může nastat, pokud

- Pojištěná doba během roku je záporná
- Pojištěná doba během roku přesáhne 365/366 dní

```
select *
from [INEP] where doba_pojistena<0 or doba_pojistena>366
```

Nálezy datové kvality					
Datum nálezu	Databáze	Popis problému	Návrh řešení	Status	Aktualizace
23.3.2015	STATMIN ANOD	Chybějící hlavička	Vložení hlavičky s názvy sloupců před první záznam	Zavřeno	23.3.2015
23.3.2015	STATMIN VZ, INEP	Všechny hodnoty jsou v uvozovkách	Odstranění uvozovek	Zavřeno	23.3.2015
23.3.2015	STATMIN ANOD, STATMIN VZ, INEP	Nevhodné datové typy	Přetypování hodnot ve všech databázích	Zavřeno	23.3.2015
26.3.2015	STATMIN ANOD, STATMIN VZ, INEP	Duplicita záznamů	Odstranění duplicit	Otevřeno	26.3.2015
26.3.2015	STATMIN VZ	Podezřelé roky narození	Doplnění z INEP	Otevřeno	26.3.2015
30.3.2015	STATMIN ANOD	Více než dva záznamy u každého ID	Konzultace s vlastníkem databáze	Nové	30.3.2015
30.3.2015	STATMIN ANOD	Chybějící pohlaví	Propojení s INEP	Nové	30.3.2015
30.3.2015	STATMIN ANOD	Chybějící informace o důchodu u jednoho ID	Odstranění záznamu	Nové	30.3.2015
30.3.2015	STATMIN VZ	Přítomnost „třetího“ pohlaví	Náhodné přiřazení	Nové	30.3.2015
30.3.2015	STATMIN VZ	Některé záznamy nemají počátek/konec zaměstnání	Konzultace s vlastníkem databáze	Nové	30.3.2015

30.3.2015	INEP	Duplicita záznamů	Použití jednoznačné kombinace	Nové	30.3.2015
30.3.2015	INEP	Odlehlá pozorování doby pojištěné	Konzultace s vlastníkem databáze	Nové	30.3.2015

Další oblastí, kde je nutné provést verifikaci datové kvality, je součet dob přes všechny stavy, ve kterých se jedinec během roku mohl nacházet. Na základě popisu dat předpokládáme, že každý rok každého jedince je možné rozdělit na následující úseky

- Doba pojištěná – pojištěná doba během roku
- Doba nepojištěná
- Náhradní doba pojištění – může být způsobena různými důvody, například:
    - Péče o dítě
    - Nezaměstnanost
    - Studium
    - Péče o závislou osobu
    - Vojenská služba
    - Ostatní - žádná z vyjmenovaných možností
- Vyloučená doba
- Jiná doba - doba neodpovídající žádnému z předchozích případů

Dále podle popisu dat předpokládáme, že součet přes všechny tyto úseky by měl ve výsledku pokrýt celý rok.

```
select *
from [INEP]
where isnull(doba_pojistena,0) + isnull(ndp_pece_o_dite,0) +
isnull(ndp_nezamestnanost,0) + isnull(ndp_studium,0) + isnull(ndp_pece,0) +
isnull(ndp_ostatni,0) + isnull(doba_nepojistena,0) +  isnull(doba_jine,0) +
isnull(ndp_vojenska_sluzba,0) not in (365,366)
```

Předchozí skript je příkladem, jak ze zkoumané databáze získat všechny záznamy, pro které neplatí, že součet přes všechny uvažované úseky pokryje celý rok, tedy že součet všech dob nebude roven ani číslu 365 ani 366. Pokud odhalíme nějaká odlehlá pozorování, která tuto podmínku porušují, lze pro další postup kontaktovat majitele databáze a doplnit chybějící informace.

Nálezy datové kvality					
Datum nálezu	Databáze	Popis problému	Návrh řešení	Status	Aktualizace
23.3.2015	STATMIN ANOD	Chybějící hlavička	Vložení hlavičky s názvy sloupců před první záznam	Zavřeno	23.3.2015
23.3.2015	STATMIN VZ, INEP	Všechny hodnoty jsou v uvozovkách	Odstranění uvozovek	Zavřeno	23.3.2015
23.3.2015	STATMIN ANOD, STATMIN VZ, INEP	Nevhodné datové typy	Přetypování hodnot ve všech databázích	Zavřeno	23.3.2015
26.3.2015	STATMIN ANOD, STATMIN VZ, INEP	Duplicita záznamů	Odstranění duplicit	Otevřeno	26.3.2015
26.3.2015	STATMIN VZ	Podezřelé roky narození	Doplnění z INEP	Otevřeno	26.3.2015
30.3.2015	STATMIN ANOD	Více než dva záznamy u každého ID	Konzultace s vlastníkem databáze	Nové	30.3.2015
30.3.2015	STATMIN ANOD	Chybějící pohlaví	Propojení s INEP	Nové	30.3.2015
30.3.2015	STATMIN ANOD	Chybějící informace o důchodu u jednoho ID	Odstranění záznamu	Nové	30.3.2015
30.3.2015	STATMIN VZ	Přítomnost „třetího“ pohlaví	Náhodné přiřazení	Nové	30.3.2015
30.3.2015	STATMIN VZ	Některé záznamy nemají počátek/konec	Konzultace s vlastníkem databáze	Nové	30.3.2015

		zaměstnání			
30.3.2015	INEP	Duplicita záznamů	Použití jednoznačné kombinace	Nové	30.3.2015
30.3.2015	INEP	Odlehlá pozorování doby pojištěné	Konzultace s vlastníkem databáze	Nové	30.3.2015
30.3.2015	INEP	Součet přes všechny úseky nepokryje přesně celý rok	Konzultace s vlastníkem databáze	Nové	30.3.2015

Z hlediska verifikace datové kvality se můžeme podívat, jestli vyměřovací základy uvedené v databázi INEP odpovídají vyměřovacím základům uvedeným v dříve zpracované databázi STATMIN VZ. Jelikož databáze STATMIN VZ obsahuje více řádků odpovídajících jednotlivým osobám, je nutné nejprve vyměřovací základy agregovat přes všechna ID a výsledek pro snazší manipulaci uložit do dočasné tabulky (viz následující skript).

```
IF object_id('tempdb.dbo.##DB_temp_VZ_aggr') is NOT NULL DROP TABLE ##DB_temp_VZ_aggr
select ID, sum(isnull(vz,0)) as vz
into ##DB_temp_VZ_aggr
from [VZ] group by ID
GO
```

V dalším kroku je potřeba obě databáze sloučit pomocí funkce `join`, aby bylo možné získat potřebné informace a vzájemně je porovnat. Protože databáze INEP obsahuje data po jednotlivých průřezových letech, je nutné pevně jeden takový rok zvolit (následující ukázka pro rok 2009). Výsledky uložme opět do temporární tabulky, abychom se k výsledkům mohli v budoucnosti vrátit.

```
IF object_id('tempdb.dbo.##DB_temp_VZ_INEP_rozdily') is NOT NULL DROP TABLE
##DB_temp_VZ_INEP_rozdily
select coalesce(A.ID, B.ID) as ID, A.vymerovaci_zaklad, B.vz
from [INEP] A full join ##DB_temp_VZ_aggr B on A.ID=B.ID and B.ID is not null
where A.ID is not null and A.rok = 2009
GO
```

Při propojení s databází INEP jsme však narazili na problém, neboť databáze INEP obsahuje záznamy i pro osoby, které jsou již po smrti. Tato skutečnost je komplikací pro většinu následných výpočtů, které se budou s databází provádět, proto se k tomuto nálezu dále vrátíme v kapitole 3, která se blíže zabývá datovou přípravou a transformací, a zesnulé osoby z prováděných výpočtů odstraníme. Pro účely specifického příkladu v části verifikace datové kvality se pouze podívejme, kterých ID se tento problém týká použitím následujícího skriptu

```
select *
from [INEP] where (rok_umrti = 0 or rok_umrti < zadany_rok)
```

Nálezy datové kvality					
Datum nálezu	Databáze	Popis problému	Návrh řešení	Status	Aktualizace
23.3.2015	STATMIN ANOD	Chybějící hlavička	Vložení hlavičky s názvy sloupců před první záznam	Zavřeno	23.3.2015
23.3.2015	STATMIN VZ, INEP	Všechny hodnoty jsou v uvozovkách	Odstranění uvozovek	Zavřeno	23.3.2015
23.3.2015	STATMIN ANOD, STATMIN VZ, INEP	Nevhodné datové typy	Přetypování hodnot ve všech databázích	Zavřeno	23.3.2015
26.3.2015	STATMIN ANOD, STATMIN VZ, INEP	Duplicita záznamů	Odstranění duplicit	Otevřeno	26.3.2015
26.3.2015	STATMIN VZ	Podezřelé roky narození	Doplnění z INEP	Otevřeno	26.3.2015

30.3.2015	STATMIN ANOD	Více než dva záznamy u každého ID	Konzultace s vlastníkem databáze	Nové	30.3.2015
30.3.2015	STATMIN ANOD	Chybějící pohlaví	Propojení s INEP	Nové	30.3.2015
30.3.2015	STATMIN ANOD	Chybějící informace o důchodu u jednoho ID	Odstranění záznamu	Nové	30.3.2015
30.3.2015	STATMIN VZ	Přítomnost „třetího“ pohlaví	Náhodné přiřazení	Nové	30.3.2015
30.3.2015	STATMIN VZ	Některé záznamy nemají počátek/konec zaměstnání	Konzultace s vlastníkem databáze	Nové	30.3.2015
30.3.2015	INEP	Duplicita záznamů	Použití jednoznačné kombinace	Nové	30.3.2015
30.3.2015	INEP	Odlehlá pozorování doby pojištěné	Konzultace s vlastníkem databáze	Nové	30.3.2015
30.3.2015	INEP	Součet přes všechny úseky nepokryje přesně celý rok	Konzultace s vlastníkem databáze	Nové	30.3.2015
30.3.2015	INEP	Přítomnost záznamů pro zesnulé osoby	Odstranění z databáze	Nové	30.3.2015

Při dalším postupu nás budou zajímat záznamy, pro které hodnoty vyměřovacích základů z databáze INEP neodpovídají vyměřovacím základům z databáze STATMIN VZ. Za tímto účelem v dočasné tabulce z předchozího kroku spočítáme rozdíly vyměřovacích základů a vybereme z ní ID, pro která jsou tyto rozdíly nenulové.

```
select ID, vymerovaci_zaklad, vz, isnull(vymerovaci_zaklad,0)-isnull(vz,0) as rozdil
from ##DB_temp_VZ_INEP_rozdily
where isnull(vymerovaci_zaklad,0)-isnull(vz,0)<>0
order by 2 desc
```

Případné nenulové hodnoty rozdílů mohou nastat dvěma způsoby

- V jedné z databází úplně chybí informace o vyměřovacím základu
  - Řešení: doplněné údaje z druhé databáze
- V obou databázích jsou různé informace o vyměřovacím základu
  - Řešení: kontaktování majitelů databází za účelem získání správné hodnoty

Pouhé porovnání rozdílů však neodhalí situaci, kdy informace o výši vyměřovacího základu pro dané ID chybí v obou databázích. Z tohoto důvodu je nutné takový případ ověřit zvlášť, a pokud nastane, kontaktovat majitele databází za účelem doplnění informací.

Nálezy datové kvality					
Datum nálezu	Databáze	Popis problému	Návrh řešení	Status	Aktualizace
23.3.2015	STATMIN ANOD	Chybějící hlavička	Vložení hlavičky s názvy sloupců před první záznam	Zavřeno	23.3.2015
23.3.2015	STATMIN VZ, INEP	Všechny hodnoty jsou v uvozovkách	Odstranění uvozovek	Zavřeno	23.3.2015
23.3.2015	STATMIN ANOD, STATMIN VZ, INEP	Nevhodné datové typy	Přetypování hodnot ve všech databázích	Zavřeno	23.3.2015
26.3.2015	STATMIN ANOD, STATMIN VZ, INEP	Duplicita záznamů	Odstranění duplicit	Otevřeno	26.3.2015
26.3.2015	STATMIN VZ	Podezřelé roky narození	Doplnění z INEP	Otevřeno	26.3.2015
30.3.2015	STATMIN ANOD	Více než dva záznamy u každého ID	Konzultace s vlastníkem databáze	Nové	30.3.2015
30.3.2015	STATMIN ANOD	Chybějící pohlaví	Propojení s INEP	Nové	30.3.2015
30.3.2015	STATMIN ANOD	Chybějící informace o důchodu u jednoho ID	Odstranění záznamu	Nové	30.3.2015
30.3.2015	STATMIN VZ	Přítomnost „třetího“ pohlaví	Náhodné přiřazení	Nové	30.3.2015

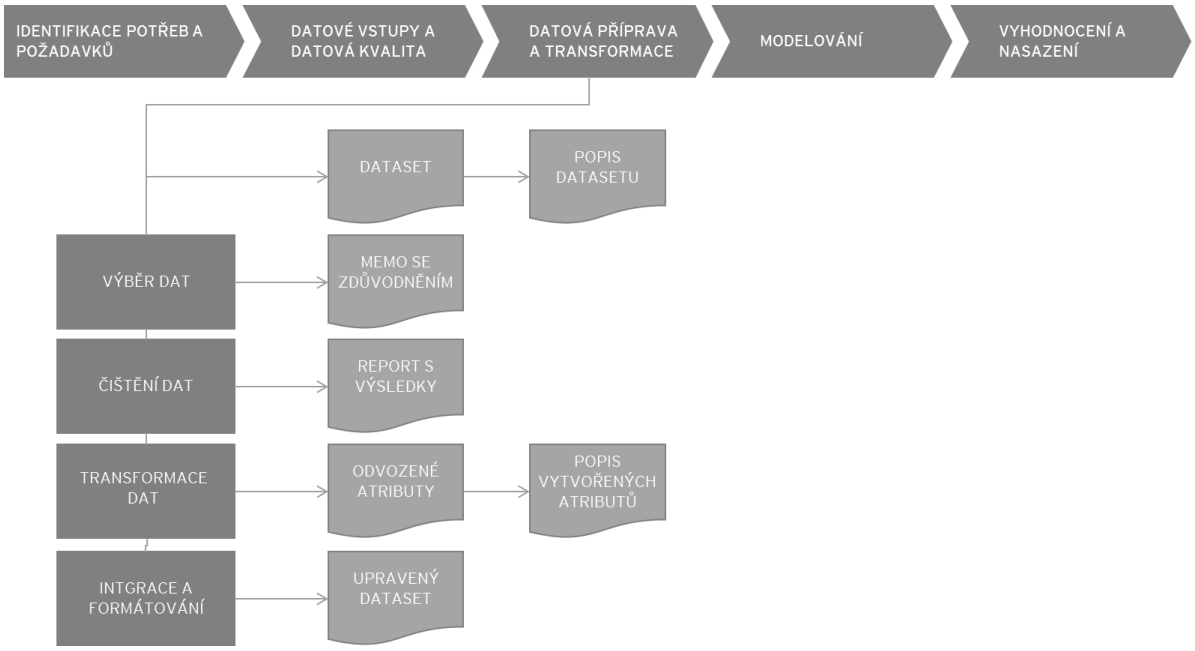
30.3.2015	STATMIN VZ	Některé záznamy nemají počátek/konec zaměstnání	Konzultace s vlastníkem databáze	Nové	30.3.2015
30.3.2015	INEP	Duplicita záznamů	Použití jednoznačné kombinace	Nové	30.3.2015
30.3.2015	INEP	Odlehlá pozorování doby pojištěné	Konzultace s vlastníkem databáze	Nové	30.3.2015
30.3.2015	INEP	Součet přes všechny úseky nepokryje přesně celý rok	Konzultace s vlastníkem databáze	Nové	30.3.2015
30.3.2015	INEP	Přítomnost záznamů pro zesnulé osoby	Odstranění z databáze	Nové	30.3.2015
30.3.2015	STATMIN VZ, INEP	Nekonzistence vyměřovacích základů	Konzultace s vlastníky obou databází	Nové	30.3.2015

3. Datová příprava a transformace

Datová příprava a transformace shrnuje všechny aktivity, které jsou potřeba k vytvoření finálního datasetu, který bude následně použit k modelování. Jedná se o jednu z nejdůležitějších a časově nejnáročnějších částí samotného data-miningu. Datová příprava často zahrnuje například

- Sloučení datových souborů a záznamů
- Výběr podmnožin dat
- Odvození nových atributů
- Odstranění nebo nahrazení chybějících hodnot

Následující obrázek znázorňuje jednotlivé fáze procesu datové přípravy a transformace dat. Základním krokem tohoto postupu je detailní popis datasetu, se kterým se bude následně pracovat. Proces datové přípravy se skládá z výběru dat (vedoucímu k vytvoření výsledného mema) a čištění dat (vedoucímu k reportu). Následuje samotná transformace dat, která poskytne odvozené atributy (včetně popisu). Posledním krokem je integrace dat a konečné formátování, jehož výsledkem je finální dataset vhodný k samotnému modelování.



Obrázek 9: Dílčí úlohy a výstupy týkající se datové přípravy a transformace

3.1 Výběr dat

Účelem této kapitoly je výběr dat, která budou finálně použita pro vstup do mikrosimulačního modelu. Jejím výchozím bodem je rešerše datových zdrojů. Nicméně v této části můžeme zohlednit skutečnost, že některá data jsou obsažena ve více zdrojích. Díky předchozímu průzkumu máme informaci o tom, které z nich jsou lepší (například úplnější), proto můžeme rozhodnout, že pro tento atribut použijeme pouze jeden konkrétní zdroj.

3.1.1 Cíl

Cílem této fáze je:

- Výběr dat vhodných k další analýze

3.1.2 Výstup

Výstupem této fáze je vytvoření mema obsahujícího:
- Seznam dat, která budou použita nebo vyloučena při další analýze
- Zdůvodnění použití nebo vyloučení dat
- Návrh řešení
- Návrhy na zlepšení

3.1.3 Činnost

V této fázi je ale třeba důkladně zvážit
- Závislost správnosti výsledku na kvalitě daného datasetu nebo atributu
    - Kvalita atributu může ovlivnit výsledek
        - Opětovné provedení posouzení kvality dat
        - Navržení postupu zlepšení kvality
    - Výsledek nezávisí na kvalitě atributu
        - Není třeba se kvalitou atributu dále zabývat

Poznámka: Pokud kvalita atributu může ovlivnit výsledek modelování, je nutné tuto skutečnost brát do úvahy v dalších krocích.

- Významnost atributu
- Omezení na použití jednotlivých polí
- Možnost dodatečného sběru dat
    - Pokud v této fázi zjistíme, že nemáme k dispozici všechna potřebná data, je třeba zvážit jejich doplnění
        - Z jiné databáze
        - Z externích zdrojů
        - Použít jiné metody k získání dat
            - Interpolace dat
            - Extrapolace dat
            - Zpětné použití modelu k odvození chybějících informací

Pokud uvažujeme o vyloučení některých dat, je klíčové vědět, jestli tyto data
- Můžeme v budoucnosti potřebovat
    - Nutná záloha těchto dat
        - Záloha na vhodné medium (CD, DVD, flash disk)
        - Virtuální záloha
        - Kombinace těchto možností
- Nebudou potřebná
    - Nutná dokumentace vyloučených dat

Při výběru se můžeme zaměřit pouze na některé podmnožiny, přitom je třeba dokumentovat
- Identifikace podmnožiny
- Zdůvodnění rozhodnutí k zúžení výběru
- Otestování významnosti podmnožiny
- Možnost modifikace podmnožiny
    - Malá změna podmnožiny může vést ke změně významnosti skupiny, např.

- Změna věkové skupiny (např. místo rozmezí 20-28 uvažovat 21-29) při zachování ostatních hodnot
- Změna skupiny primárního důchodu (např. místo skupiny do 6 000 Kč uvažovat skupinu do 8 000 Kč) při zachování ostatních hodnot

3.1.4 Doporučení

Některá data jsou obsažena ve více datových souborech. Pro další postup je dobré vybrat ta, která jsou nejúplnější či nejkvalitnější na úrovni zdroje.

3.1.5 Specifický příklad

Pro účely specifického příkladu této podkapitoly datové přípravy se zaměříme na výběr dat pro přípravu matice pravděpodobností přechodu ze stavu zaměstnanosti do neaktivity. Důvodem této volby je skutečnost, že potřebná data budeme vybírat ze všech tří v současnosti dostupných individuálních databází.

Protože budeme chtít, aby byly výpočty snadno replikovatelné pro různé roky, uložíme si na začátku zvolený rok do temporární tabulky, ze které budeme potřebné údaje v budoucnosti snadno brát jako lokální proměnné. Tento postup je ukázán v následujícím kódu, ve kterém mají lokální proměnné následující význam:

- `@RRRR` – zvolený rok (např. 2012),
- `@RRRRMM` – zvolený rok následovaný měsícem (např. 201209 pro září 2012),
- `@sRRRRPrev` – rok předcházející zvolenému (např. 2011 pro zvolený 2012).

```
if object_id('tempdb.dbo.##DB_tempDate') is NOT NULL drop table ##DB_tempDate
declare @RRRR int
set @RRRR = zvoleny_rok
select @RRRR as RRRR into ##DB_tempDate
GO

declare @RRRR int
declare @RRRRMM int
declare @sRRRRPrev int


select top 1 @RRRR= RRRR from ##DB_tempDate
select top 1 @RRRRMM = @RRRR * 100 + 1
select top 1 @sRRRRPrev = convert(char(4), dateadd(m,-12,convert(datetime,
convert(char(8), @RRRRMM*100+1, 112)) ),112)
```

Pro vytvoření matice pro jeden zvolený rok budeme potřebovat data z

- INEP pro předchozí průřezový rok `@sRRRRPrev`
- STATMIN ANOD pro předchozí rok `@sRRRRPrev`
- STATMIN ANOD pro současný rok `@RRRR`
- STATMIN VZ pro současný rok `@RRRR`

V prvním kroku je potřeba zajistit, abychom měli data dostupná po jednotlivých letech. Databáze STATMIN ANOD a STATMIN VZ jsou vytvářené postupně každý rok, není tedy třeba je dále dělit. Komplikace nastává s databází INEP, pro kterou takové rozdělení neexistuje, a veškerá data jsou uložena v jednom souboru obsahujícím přes 207 milionů záznamů. Je tedy nutné ji rozdělit po průřezových letech, aby bylo možné její propojení s prvními dvěma datovými zdroji. Následující skript ukazuje příklad takového výběru a jeho uložení do dočasné tabulky:

```
if object_id('tempdb.dbo.##DB_temp_INEP_cast') is not null drop table
##DB_temp_INEP_cast
select ID_OSOBY, CIS_POHLAVI, ROK_UMRTI, ROK, SKR_ZAMESTNANEC
into ##DB_temp_INEP_cast
from [INEP] where rok = @RRRR
GO
```

V dalším postupu budeme vycházet z databáze INEP, protože podle popisu obsahuje nejúplnější informace ze všech tří v současnosti dostupných individuálních datových zdrojů. Pomocí předchozího skriptu vybereme data za rok předcházející zkoumanému roku, abychom zjistili pro všechny záznamy stav ke konci roku. Budeme rozlišovat dva hlavní stavy

- zaměstnaný,
- neaktivní.

Protože budeme chtít získat rozdělení populace nejen podle pohlaví a věku, ale také podle stupně invalidity, doplníme do vybrané části databáze INEP tyto informace z databáze STATMIN ANOD z předchozího roku. Této části se budeme více věnovat v kapitole o integraci dat.

```
IF object_id('tempdb.dbo.##DB_temp_spojeni_cast') is NOT NULL DROP TABLE
##DB_temp_spojeni_cast
select A.*
      ,B.[rok_priznani]
      ,B.[pohlavi_anod]
      ,B.[typ_primarniho]
      ,B.[typ_odvozeneho]
into ##DB_temp_spojeni_cast
from [##DB_temp_INEP_cast] A
          left join [ANOD_predchozi] B on A.ID=B.ID
GO
```

Propojení provedeme pomocí funkce `left join`. Takto doplníme informaci o důchodech z databáze STATMIN ANOD k databázi INEP, neboť první datový zdroj obsahuje informace pouze o důchodcích, zatímco druhý o celé populaci. Podobným způsobem doplníme informace o stavu invalidity jedinců ke konci aktuálního roku opět z databáze STATMIN ANOD. Před propojením je nutné přejmenovat sloupce této databáze, např. následujícím způsobem

```
IF object_id('tempdb.dbo.##DB_temp_ANOD_soucasny') is NOT NULL DROP TABLE
##DB_temp_ANOD_soucasny
select ID  ,typ_primarniho as typ_primarniho_konec
           ,typ_odvozeneho as typ_odvozeneho_konec
into ##DB_temp_ANOD_soucasny
from [ANOD_soucasny]
```

Důvodem, který stojí za nutností přejmenování sloupců, je skutečnost, že slučujeme všechny informace do jedné tabulky, kde nesmí být dva sloupce se stejným jménem. V dalším kroku je možné doplnit zbytek potřebných informací k předchozímu částečnému propojení

```
IF object_id('tempdb.dbo.##DB_temp_spojeni') is NOT NULL DROP TABLE ##DB_temp_spojeni
select A.*
      ,B.[typ_primarniho_konec]
      ,B.[typ_odvozeneho_konec]
into ##DB_temp_spojeni
from [##DB_temp_spojeni_cast] A
          left join ##DB_temp_ANOD_soucasny B on A.ID=B.ID
GO
```

Dalším krokem při výběru dat k vytvoření matice přechodu je získání potřebných informací z databáze STATMIN VZ. Za pomoci proměnné `ixyear` načteme nezbytná data za současný rok (`@RRRR`) do dočasné tabulky pro snazší manipulaci v dalších částech.

```
if object_id('tempdb.dbo.##DB_temp_VZ') is NOT NULL drop table ##DB_temp_VZ
select ID,OD,DO
into ##DB_temp_VZ
from [VZ] where ixyear = @RRRR
GO
```

3.2 Čištění dat

Report kvality dat připravený v kapitole Datové vstupy a datová kvalita v části Verifikace dat je základem při čištění dat, neboť obsahuje detaily o problémech, které se v datech vyskytují.

3.2.1 Cíl

Cílem této fáze je

- Zvýšení kvality dat na úroveň požadovanou modelem například použitím následujících metod
    - Výběr očištěných podmnožin dat
    - Vložení vhodných standardních hodnot
    - Odhad chybějících hodnot náhodným přiřazením na základě pravděpodobnostních rozdělení

3.2.2 Výstup

Výstupem této fáze je vytvoření reportu s výsledky. Tento report obsahuje

- Popis postupů, které byly použity za účelem odstranění problémů odhalených při verifikaci dat
- Popis rozhodnutí, které vedly k těmto postupům
- Návrh řešení
- Návrhy na zlepšení

Pokud nedošlo k odstranění všech problémů, report musí obsahovat

- Seznam a popis přetrvávajících problémů
- Jejich možný dopad na výsledky modelování

3.2.3 Činnost

Pokud byla při verifikaci kvality dat odhalena přítomnost šumu, existuje několik možností, jak s ním naložit

- Oprava
    - Identifikace technik použitých k opravě
    - Zdůvodnění použitého postupu
    - Dokumentace úspěšnosti
- Odstranění
    - Identifikace technik použitých k odstranění šumu
    - Zdůvodnění použitého postupu
    - Dokumentace úspěšnosti

    - Navržení postupu při nemožnosti odstranění šumu
- Ignorace
    - Zdůvodnění neodstranění šumu
    - Dokumentace možného dopadu na výsledky modelování

Při verifikaci datové kvality mohla být odhalena řada problémů. Tyto problémy musí být dostatečně dokumentovány a pokud možno opraveny.

- Chybějící hodnoty
    - Možná řešení:
        - Vynechání řádků
            - Dokumentace vynechaných informací
            - Důležitost těchto informací při modelování
            - Možný dopad na výsledek modelování
        - Vynechání atributů
            - Dokumentace vynechaných atributů
            - Zdůvodnění vynechání atributů
            - Možný dopad na výsledek modelování
        - Doplnění chybějících údajů
            - Vhodné standardní hodnoty
            - Odhady chybějících hodnot
- Chyby v datech
    - Možná řešení:
        - Vynechání atributů
            - Dokumentace vynechaných atributů
            - Zdůvodnění vynechání atributů
            - Možný dopad na výsledek modelování
        - Manuální oprava
            - Dokumentace všech manuálních zásahů do dat
            - Důležitá následná kontrola všech oprav
- Nesprávné jednotky
    - Možná řešení:
        - Manuální oprava
            - Dokumentace všech manuálních zásahů do dat
            - Důležitá následná kontrola všech oprav
- Nesrovnalosti v kódování
    - Možná řešení:
        - Jednotné kódování ve všech zdrojích
            - Manuální oprava (nízký počet polí)
                - Dokumentace všech manuálních zásahů do dat
                - Důležitá následná kontrola všech oprav
            - Systémová oprava samotného vytváření databáze
                - Dokumentace všech systémových úprav

3.2.4 Doporučení

V databázi mohou existovat pole, která nejsou pro následné modelování významná. Tím pádem i šum, který tato pole mohou obsahovat, nemá významný vliv na výsledek modelování. Je ale důležité tyto informace dokumentovat, neboť v budoucnu se významnost ignorovaného šumu může změnit.

3.2.5 Specifický příklad

Jako příklad čištění dat zvolíme databázi STATMIN VZ, ve které jsme v dřívějších fázích kontroly kvality dat odhalili, že v proměnné pohlaví se kromě hodnot 1 (muž) a 2 (žena) vyskytuje také v několika případech hodnota nula (dokonce 1617 pozorování) nebo jiná chyba. Ukažme nyní dva postupy, jak s těmito hodnotami naložit

- Odstranění z následujících analýz
- Náhodné přiřazení hodnoty pohlaví

```
IF object_id('tempdb.dbo.##DB_temp_ukazka_pohlavi') is NOT NULL DROP TABLE
##DB_temp_ukazka_pohlavi
select ID, ixyear, pohlavi,
          case when pohlavi=1 THEN '0'
               when pohlavi=2 THEN '1'
               else 'ERROR' end as pohlavi1,
          case when pohlavi=1 THEN '0'
               when pohlavi=2 THEN '1'
               else case when RAND(ID)<= hranice then '0' else '1' end end as pohlavi2
into ##DB_temp_ukazka_pohlavi
from [VZ]
GO
```

Do proměnné `pohlavi1` jsme vložili variantu, kdy při jiné hodnotě než 1 nebo 2 bude vrácena chybová hodnota a tak se snadno v dalším postupu oddělí problémová pozorování. Naopak do proměnné `pohlavi2` je náhodně přiřazena hodnota pohlaví na základě ID. V této části je nutné určit vhodnou hodnotu omezující proměnné `hranice` tak, aby bylo zachováno současné rozdělení pohlaví mezi muži a ženami.
Zároveň je v obou variantách proměnná `pohlavi` přetypována tak, aby mužům byla přiřazena hodnota 0 a ženám hodnota 1. Takové přetypování je vhodné provést i v ostatních použitých databázích, aby byla zaručena konzistence této proměnné napříč všemi datovými zdroji.

Nálezy datové kvality					
Datum nálezu	Databáze	Popis problému	Návrh řešení	Status	Aktualizace
23.3.2015	STATMIN ANOD	Chybějící hlavička	Vložení hlavičky s názvy sloupců před první záznam	Zavřeno	23.3.2015
23.3.2015	STATMIN VZ, INEP	Všechny hodnoty jsou v uvozovkách	Odstranění uvozovek	Zavřeno	23.3.2015
23.3.2015	STATMIN ANOD, STATMIN VZ, INEP	Nevhodné datové typy	Přetypování hodnot ve všech databázích	Zavřeno	23.3.2015
26.3.2015	STATMIN ANOD, STATMIN VZ, INEP	Duplicita záznamů	Odstranění duplicit	Otevřeno	26.3.2015
26.3.2015	STATMIN VZ	Podezřelé roky narození	Doplnění z INEP	Otevřeno	26.3.2015
30.3.2015	STATMIN ANOD	Více než dva záznamy u každého ID	Konzultace s vlastníkem databáze	Otevřeno	30.3.2015
30.3.2015	STATMIN ANOD	Chybějící pohlaví	Propojení s INEP	Otevřeno	30.3.2015
30.3.2015	STATMIN ANOD	Chybějící informace o důchodu u jednoho ID	Odstranění záznamu	Otevřeno	30.3.2015
30.3.2015	STATMIN VZ	Přítomnost „třetího“ pohlaví	Náhodné přiřazení	Zavřeno	5.4.2015
30.3.2015	STATMIN VZ	Některé záznamy nemají počátek/konec zaměstnání	Konzultace s vlastníkem databáze	Otevřeno	30.3.2015
30.3.2015	INEP	Duplicita záznamů	Použití jednoznačné kombinace	Otevřeno	30.3.2015
30.3.2015	INEP	Odlehlá pozorování doby pojištěné	Konzultace s vlastníkem databáze	Otevřeno	30.3.2015

30.3.2015	INEP	Součet přes všechny úseky nepokryje přesně celý rok	Konzultace s vlastníkem databáze	Otevřeno	30.3.2015
30.3.2015	INEP	Přítomnost záznamů pro zesnulé osoby	Odstranění z databáze	Otevřeno	30.3.2015
30.3.2015	STATMIN VZ, INEP	Nekonzistence vyměřovacích základů	Konzultace s vlastníky obou databází	Otevřeno	30.3.2015

Dalším příkladem čištění dat bude analýza jednotlivých roků narození uvedených v databázi STATMIN VZ. V dřívější části jsme při kontrole datové kvality této databáze narazili na výskyt několika chyb. Počty takových pozorování jsou v současnosti následující

- 1617 pozorování má nulový věk narození (stejně jako nulových pohlaví)
- 1931 pozorování má jinou chybu v roce narození

Navíc jsme při analýze této databáze narazili na skutečnost, že se v ní vyskytují případy, kdy rok narození je nižší než 1900, nebo vyšší než rok současný.
Protože budeme chybné roky narození doplňovat z databáze INEP, je nejprve nutné roky narození v ní obsažené agregovat přes jednotlivá ID.

```
IF object_id('tempdb.dbo.##DB_temp_INEP_aggr_roknar') is NOT NULL DROP TABLE
##DB_temp_INEP_aggr_roknar
select ID, min(isnull(rok_narozeni,0)) as min_roknar
         , max(isnull(rok_narozeni,0)) as max_roknar
into ##DB_temp_INEP_aggr_roknar
from [INEP] group by ID
GO
```

Ukažme nyní, jak je možné doplnit roky narození z databáze INEP do STATMIN VZ pomocí funkce `left join` přes jednoznačný identifikátor ID pro odlehlá pozorování, pro která platí, že rok narození je buď menší než 1900 nebo větší než rok současný.

```
IF object_id('tempdb.dbo.##DB_temp_ukazka_roknar') is NOT NULL DROP TABLE
##DB_temp_ukazka_roknar
select A.ID, A.roknar
           , B.min_roknar as roknar_INEP
           ,coalesce(B.min_roknar,A.roknar) as roknar_upr
##DB_temp_ukazka_roknar
from [VZ] A left join ##DB_temp_INEP_aggr_roknar B on A.ID=B.ID
where (roknar < 1900 or roknar > 2015)
GO
```

Aplikací předchozího skriptu dosáhneme toho, že při použití pouze jednoho průřezového roku databáze INEP došlo k doplnění informací ve více než 100 pozorováních z původních přibližně 3500.

Nálezy datové kvality					
Datum nálezu	Databáze	Popis problému	Návrh řešení	Status	Aktualizace
23.3.2015	STATMIN ANOD	Chybějící hlavička	Vložení hlavičky s názvy sloupců před první záznam	Zavřeno	23.3.2015
23.3.2015	STATMIN VZ, INEP	Všechny hodnoty jsou v uvozovkách	Odstranění uvozovek	Zavřeno	23.3.2015
23.3.2015	STATMIN ANOD, STATMIN VZ, INEP	Nevhodné datové typy	Přetypování hodnot ve všech databázích	Zavřeno	23.3.2015
26.3.2015	STATMIN ANOD, STATMIN VZ, INEP	Duplicita záznamů	Odstranění duplicit	Otevřeno	26.3.2015
26.3.2015	STATMIN VZ	Podezřelé roky narození	Doplnění z INEP	Zavřeno	5.4.2015

30.3.2015	STATMIN ANOD	Více než dva záznamy u každého ID	Konzultace s vlastníkem databáze	Otevřeno	30.3.2015
30.3.2015	STATMIN ANOD	Chybějící pohlaví	Propojení s INEP	Otevřeno	30.3.2015
30.3.2015	STATMIN ANOD	Chybějící informace o důchodu u jednoho ID	Odstranění záznamu	Otevřeno	30.3.2015
30.3.2015	STATMIN VZ	Přítomnost „třetího“ pohlaví	Náhodné přiřazení	Zavřeno	5.4.2015
30.3.2015	STATMIN VZ	Některé záznamy nemají počátek/konec zaměstnání	Konzultace s vlastníkem databáze	Otevřeno	30.3.2015
30.3.2015	INEP	Duplicita záznamů	Použití jednoznačné kombinace	Otevřeno	30.3.2015
30.3.2015	INEP	Odlehlá pozorování doby pojištěné	Konzultace s vlastníkem databáze	Otevřeno	30.3.2015
30.3.2015	INEP	Součet přes všechny úseky nepokryje přesně celý rok	Konzultace s vlastníkem databáze	Otevřeno	30.3.2015
30.3.2015	INEP	Přítomnost záznamů pro zesnulé osoby	Odstranění z databáze	Otevřeno	30.3.2015
30.3.2015	STATMIN VZ, INEP	Nekonzistence vyměřovacích základů	Konzultace s vlastníky obou databází	Otevřeno	30.3.2015

Posledním příkladem, na který se zaměříme při čištění dat, je v databázi INEP ošetření přítomnosti osob, které jsou ve zvoleném průřezovém roce po smrti. V následující části budeme vycházet z dočasné tabulky `##DB_temp_INEP_cast`, ve které je uložena část databáze INEP pro vybraný rok. Tento krok je pro další postup nezbytný a je možné jej provést např. následujícím příkazem

```
if object_id('tempdb.dbo.##DB_temp_INEP_ocisteno') is not null drop table
##DB_temp_INEP_ocisteno
select *
into ##DB_temp_INEP_ocisteno
from ##DB_temp_INEP_cast where (rok_umrti = 0 or rok_umrti < @RRRR)
GO
```

Předchozí skript ukazuje, jak je možné takovou operaci provést pomocí dynamického přiřazování. Je tedy možné celý tento proces aplikovat pro libovolný rok podle potřeby uživatele. Jakmile je tento krok hotový, je dobré aktualizovat tabulku obsahující nálezy datové kvality.

Nálezy datové kvality					
Datum nálezu	Databáze	Popis problému	Návrh řešení	Status	Aktualizace
23.3.2015	STATMIN ANOD	Chybějící hlavička	Vložení hlavičky s názvy sloupců před první záznam	Zavřeno	23.3.2015
23.3.2015	STATMIN VZ, INEP	Všechny hodnoty jsou v uvozovkách	Odstranění uvozovek	Zavřeno	23.3.2015
23.3.2015	STATMIN ANOD, STATMIN VZ, INEP	Nevhodné datové typy	Přetypování hodnot ve všech databázích	Zavřeno	23.3.2015
26.3.2015	STATMIN ANOD, STATMIN VZ, INEP	Duplicita záznamů	Odstranění duplicit	Otevřeno	26.3.2015
26.3.2015	STATMIN VZ	Podezřelé roky narození	Doplnění z INEP	Zavřeno	5.4.2015
30.3.2015	STATMIN ANOD	Více než dva záznamy u každého ID	Konzultace s vlastníkem databáze	Otevřeno	30.3.2015
30.3.2015	STATMIN ANOD	Chybějící pohlaví	Propojení s INEP	Otevřeno	30.3.2015
30.3.2015	STATMIN ANOD	Chybějící informace o důchodu u jednoho ID	Odstranění záznamu	Otevřeno	30.3.2015
30.3.2015	STATMIN VZ	Přítomnost „třetího“ pohlaví	Náhodné přiřazení	Zavřeno	5.4.2015
30.3.2015	STATMIN VZ	Některé záznamy nemají počátek/konec zaměstnání	Konzultace s vlastníkem databáze	Otevřeno	30.3.2015

30.3.2015	INEP	Duplicita záznamů	Použití jednoznačné kombinace	Otevřeno	30.3.2015
30.3.2015	INEP	Odlehlá pozorování doby pojištěné	Konzultace s vlastníkem databáze	Otevřeno	30.3.2015
30.3.2015	INEP	Součet přes všechny úseky nepokryje přesně celý rok	Konzultace s vlastníkem databáze	Otevřeno	30.3.2015
30.3.2015	INEP	Přítomnost záznamů pro zesnulé osoby	Odstranění z databáze	Zavřeno	5.4.2015
30.3.2015	STATMIN VZ, INEP	Nekonzistence vyměřovacích základů	Konzultace s vlastníky obou databází	Otevřeno	30.3.2015

3.3 Transformace dat

V mnoha případech je pro další postup nutná transformace používaných dat, zejména doplnění chybějících dat na základě odvození. Pro transformaci dat existují dva základní přístupy

- Odvození atributů
- Generování záznamů

3.3.1 Cíl

Cílem této fáze je

- Transformace hodnot existujících atributů
- Vytvoření odvozených atributů
- Doplnění nových záznamů

3.3.2 Výstup

Výstupem této fáze jsou

- Odvozené atributy
- Generované záznamy
- Návrh řešení
- Návrhy na zlepšení

3.3.3 Činnost

Pokud je pro konstrukci nových dat k dispozici nějaký mechanismus nebo nástroj, je potřeba

- Zajistit dostupnost tohoto nástroje
- Rozhodnout, jestli je toho použití vůbec vhodné z hlediska
    - Opakovatelnosti
    - Náročnosti
    - Přesnosti
- Přehodnotit výběr dat
    - Některá dříve vynechaná data mohou být nyní zahrnuta a naopak

Při konstrukci modelu není nutné se omezovat pouze na sesbíraná data z dostupných databází. Je třeba uvažovat také odvozené atributy, které lze zkonstruovat z jednoho nebo více existujících atributů. Důvodem jejich vytvoření je

- Znalosti základních souvislostí
- Datová omezení modelovacího algoritmu
- Naznačení výsledku modelování, že některé skutečnosti nebyly dostatečně pokryty

Možné zpracování existujících atributů

- Normalizace
- Dodání nových informací
    - Přidání vah atributům
    - Vážená normalizace
- Konstrukce chybějících atributů, například
    - Odvození na základě rešerží
    - Průměrování známých hodnot
    - Agregace známých hodnot
- Převod znaků na číselné hodnoty

Kromě transformace existujících atributů je možné generovat úplně nové záznamy. Například pokud máme k dispozici segmentaci dat, je možné generovat pro každou skupinu typické členy. V takovém případě je potřeba

- Identifikovat segmentaci
- Zvolit postup konstrukce záznamu pro jednotlivé segmenty
- Zdůvodnit zvolený postup

3.3.4 Doporučení

Transformace některých atributů je nezbytná pro vstup do modelu. Jedná se především o převod znakových polí na číselný tvar.

- Dichotomní atributy (například pohlaví z muž/žena na 0/1)
- Kategoriální atributy (typ primárního nebo odvozeného důchodu)

Neodvozujte atributy jen z důvodu snížení počtu atributů, které vstupují do modelu. Důležité je rozhodnout, jakým způsobem odvození zjednoduší proces modelování.

3.3.5 Specifický příklad

Ve specifickém příkladu týkajícího se transformace dat za účelem tvorby přechodové matice navážeme na vybraná data z předchozí fáze. V první části vyjdeme z dočasné tabulky `##DB_temp_spojeni`, která obsahuje pro všechny jedince z databáze INEP také informace o tom, jestli byli v předchozím roce

- zdraví,
- invalidní – prvního, druhého nebo třetího stupně.

V tomto případě za zdravé osoby považujeme všechny, které neměly přiřazený důchod v databázi STATMIN ANOD. Zároveň tabulka obsahuje pro všechny záznamy podobnou informaci ke konci zkoumaného roku, díky níž budeme zkoumat, jak se změnil u každého jednotlivce během roku jeho zdravotní stav. Jak je vidět v následujícím skriptu, budeme rozlišovat následující skupiny

- zdraví (`'zdravy'`)
- invalidní prvního stupně (`'prvni_stupen'`)
- invalidní druhého stupně (`'druhy_stupen'`)
- invalidní třetího stupně (`'treti_stupen'`)

Pro každou z těchto skupin budeme dále rozlišovat dva případy a to pokud

- jedinec setrval během roku v daném stavu (`'staly'`)
- jedinec se během roku přesunul do jiného stavu (`'zmena'`)

```
IF object_id('tempdb.dbo.##DB_temp_marginalni') is NOT NULL DROP TABLE
##DB_temp_marginalni
select DISTINCT ID, rok_narozeni, cis_pohlavi_id, skr_zamestnanec, skr_osvc,
typ_primarniho, typ_primarniho_konec
```

```
,case when cast(typ_primarniho as varchar(max)) = 'IP' and cast(typ_primarniho_konec
as varchar(max)) = 'IP' then 'prvni_stupen_staly'
     when cast(typ_primarniho as varchar(max)) = 'IP' and cast(typ_primarniho_konec as
varchar(max)) != 'IP' then 'prvni_stupen_zmena'
     when cast(typ_primarniho as varchar(max)) = 'ID' and cast(typ_primarniho_konec as
varchar(max)) = 'ID' then 'druhy_stupen_staly'
     when cast(typ_primarniho as varchar(max)) = 'ID' and cast(typ_primarniho_konec as
varchar(max)) != 'ID' then 'druhy_stupen_zmena'
     when cast(typ_primarniho as varchar(max)) = 'IT' and cast(typ_primarniho_konec as
varchar(max)) = 'IT' then 'treti_stupen_staly'
     when cast(typ_primarniho as varchar(max)) = 'IT' and cast(typ_primarniho_konec as
varchar(max)) != 'IT' then 'treti_stupen_zmena'
     when cast(typ_primarniho as varchar(max)) != 'IP' and cast(typ_primarniho as
varchar(max)) != 'ID' and cast(typ_primarniho as varchar(max)) != 'IT'
           and (cast(typ_primarniho_konec as varchar(max)) = 'IP' or
cast(typ_primarniho_konec as varchar(max)) = 'ID' or cast(typ_primarniho_konec as
varchar(max)) = 'IT') then 'zdravy_zmena'
                    else 'zdravy_staly' end as duchod_zmeny
into ##DB_temp_marginalni
from ##DB_temp_spojeni
GO
```

V další části ukážeme, jak vytvořit pro každé ID nepřekrývající se intervaly, které budou indikovat, že jedinec byl v jimi pokrývající době bez pracovního poměru a tedy neaktivní na trhu práce. Vyjdeme opět z podkapitoly výběru dat, tentokrát využijeme dočasnou tabulku ##DB_temp_VZ, obsahující data z databáze STATMIN VZ z daného roku. V prvním kroku vytvoříme na základě informací z databáze STATMIN VZ nejdelší nepřekrývající se intervaly zaměstnanosti pro jednotlivá ID. Začneme ošetřením intervalů na začátku roku, například pomocí následujícího skriptu.

```
if object_id('tempdb.dbo.##DB_temp_zacatekRoku') is NOT NULL drop table
##DB_temp_zacatekRoku
select ID, YEAR(OD) as ROK, MIN(OD) as MIN_OD
into ##DB_temp_zacatekRoku
from ##DB_temp_VZ
group by ID, YEAR(OD)
having MIN(OD) > (cast(YEAR(OD) as varchar(4)) + '-01-01')
```

Podobným způsobem se zaměříme také na situaci týkající se konce roku.

```
if object_id('tempdb.dbo.##DB_temp_konecRoku') is NOT NULL drop table
##DB_temp_konecRoku
select ID, YEAR(DO) as ROK, MAX(DO) as MAX_DO
into ##DB_temp_konecRoku
from ##DB_temp_VZ
group by ID, YEAR(DO)
having MAX(DO) < (cast(YEAR(DO) as varchar(4)) + '-12-31')
```

Jakmile máme ošetřené případy na začátku a konci roku, je možné použitím následujícího skriptu napočítat pro každou osobu nejdelší nepřekrývající se intervaly, ve kterých byla během zkoumaného roku zaměstnána. Výsledky uložíme do dočasné tabulky, což umožní jejich snadné použití pro další operace se získanými transformovanými veličinami.

```
if object_id('tempdb.dbo.##DB_temp_spojeniIntervalu') is NOT NULL drop table
##DB_temp_spojeniIntervalu
SELECT ROW_NUMBER() OVER(ORDER BY s1.ID) AS DAT_ID,
       s1.ID,
       s1.OD,
       MIN(t1.DO) AS DO
into ##DB_temp_spojeniIntervalu
FROM ##DB_temp_VZ s1
```

```
INNER JOIN ##DB_temp_VZ t1 ON (s1.ID = t1.ID AND s1.OD <= t1.DO)
  AND NOT EXISTS(SELECT * FROM ##DB_temp_VZ t2 WHERE t1.ID = t2.ID AND t1.DO >= t2.OD
                                                                    AND t1.DO < t2.DO)
WHERE NOT EXISTS(SELECT * FROM ##DB_temp_VZ s2 WHERE s1.ID = s2.ID AND s1.OD > s2.OD
                                                                    AND s1.OD <= s2.DO)
and s1.ID = 1001457950
GROUP BY s1.ID, s1.OD ORDER BY s1.ID, s1.OD
```

V dalším kroku odvodíme z již známých informací pro každé ID, zda se během roku vyskytly nějaké mezery v zaměstnanosti (výsledky uložíme do dočasné tabulky ##DB_temp_mezery, na kterou se budeme v následujících částech odvolávat). Celý tento proces je možné provést např. pomocí následujícího skriptu

```
if object_id('tempdb.dbo.##DB_temp_mezery') is NOT NULL drop table ##DB_temp_mezery
;WITH
employmentData as (SELECT ROW_NUMBER() OVER(ORDER BY s1.ID) AS DAT_ID,
                  s1.ID,
                  s1.OD,
                  MIN(t1.DO) AS DO,
                  datediff(dd, s1.OD, MIN(t1.DO)) as DNI
           FROM ##DB_temp_VZ s1
           INNER JOIN ##DB_temp_VZ t1 ON (s1.ID = t1.ID AND s1.OD <=
dateadd(dd,1,t1.DO))
             AND NOT EXISTS(SELECT * FROM ##DB_temp_VZ t2
                            WHERE t1.ID = t2.ID
                                    AND dateadd(dd,1,t1.DO) >= t2.OD
                                          AND dateadd(dd,1,t1.DO) < t2.DO)
           WHERE NOT EXISTS(SELECT * FROM ##DB_temp_VZ s2
                                  WHERE s1.ID = s2.ID
                                          AND s1.OD > s2.OD
                                                AND s1.OD <= dateadd(dd,1,s2.DO))
           GROUP BY s1.ID, s1.OD
           HAVING datediff(dd, s1.OD, MIN(t1.DO)) < 365
),
yearBeginning AS (SELECT ID, MIN(OD) as MIN_OD
           FROM employmentData
           GROUP BY ID
           HAVING MIN(OD) > (CAST(YEAR(MIN(OD)) AS VARCHAR(4)) + '-01-01')
),
yearEnd AS (SELECT ID, MAX(DO) as MAX_DO
           FROM employmentData
           GROUP BY ID
           HAVING MAX(DO) < (CAST(YEAR(MAX(DO)) AS VARCHAR(4)) + '-12-31')
)

SELECT T1.ID, DATEADD(dd, 1, T2.DO) AS GAP_OD, DATEADD(dd, -1, T1.OD) AS GAP_DO
into ##DB_temp_mezery
FROM
    (   SELECT DISTINCT ID, OD, ROW_NUMBER() OVER (ORDER BY ID) RN
        FROM employmentData T1
        WHERE
            NOT EXISTS (
                SELECT *
                FROM employmentData T2
                WHERE T1.ID = T2.ID AND T1.OD > T2.OD AND T1.OD < T2.DO
            )
        ) T1
    JOIN (SELECT DISTINCT ID, DO, ROW_NUMBER() OVER (ORDER BY ID) RN
        FROM employmentData T1
        WHERE
            NOT EXISTS (
                SELECT *
```

```
                FROM employmentData T2
                WHERE T1.ID = T2.ID AND T1.DO > T2.OD AND T1.DO < T2.DO
            )
    ) T2
    ON T1.ID = T2.ID AND T1.RN - 1 = T2.RN
WHERE
    T2.DO < T1.OD
UNION
SELECT ID,
       CAST(CAST(YEAR(MIN_OD) AS VARCHAR(4)) + '-01-01' AS DATE) AS GAP_OD,
       DATEADD(dd, -1, MIN_OD) AS GAP_DO
FROM yearBeginning
UNION
SELECT ID,
       DATEADD(dd, 1, MAX_DO) AS GAP_OD,
       CAST(CAST(YEAR(MAX_DO) AS VARCHAR(4)) + '-12-31' AS DATE) AS GAP_DO
FROM yearEnd
;
```

Posledním příkladem transformace dat, na který se zaměříme, je odvození dvou nových skupin proměnných. V následující části budeme vycházet z dočasné tabulky `##DB_temp_marginalni`, ve které je uložena část databáze INEP pro vybraný rok doplněna o informace o změnách zdravotních stavů. Tuto tabulku je potřeba doplnit o informace o mezerách v zaměstnanosti, což bude blíže ukázáno v podkapitole o integraci dat.

Po přidání všech potřebných informací můžeme vytvořit dvě nové skupiny proměnných odvozených od známých parametrů

- `nezam_1`, ..., `nezam_12` – indikátor stavu, kdy byla osoba bez zaměstnání celý měsíc
    - speciálně `nezam_0` – indikátor stavu nezaměstnanosti ke konci předchozího roku
- `vstup_1`, ..., `vstup_12` – indikátor stavu, kdy osoba daný měsíc vystoupila ze zaměstnanosti

```
if object_id('tempdb.dbo.##DB_temp_mesice') is NOT NULL drop table ##DB_temp_mesice
select ID
,sign(sum(case when skr_zamestnanec = 1 or skr_osvc = 1 then 0 else 1 end)) as nezam_0
,sign(sum(case when month(MEZERA_OD) <= 1 and month(MEZERA_DO) > 1 then 1 else 0 end))
as nezam_1
,sign(sum(case when month(MEZERA_OD) <= 2 and month(MEZERA_DO) > 2 then 1 else 0 end))
as nezam_2
,sign(sum(case when month(MEZERA_OD) <= 3 and month(MEZERA_DO) > 3 then 1 else 0 end))
as nezam_3
,sign(sum(case when month(MEZERA_OD) <= 4 and month(MEZERA_DO) > 4 then 1 else 0 end))
as nezam_4
,sign(sum(case when month(MEZERA_OD) <= 5 and month(MEZERA_DO) > 5 then 1 else 0 end))
as nezam_5
,sign(sum(case when month(MEZERA_OD) <= 6 and month(MEZERA_DO) > 6 then 1 else 0 end))
as nezam_6
,sign(sum(case when month(MEZERA_OD) <= 7 and month(MEZERA_DO) > 7 then 1 else 0 end))
as nezam_7
,sign(sum(case when month(MEZERA_OD) <= 8 and month(MEZERA_DO) > 8 then 1 else 0 end))
as nezam_8
,sign(sum(case when month(MEZERA_OD) <= 9 and month(MEZERA_DO) > 9 then 1 else 0 end))
as nezam_9
,sign(sum(case when month(MEZERA_OD) <= 10 and month(MEZERA_DO) > 10 then 1 else 0
end)) as nezam_10
,sign(sum(case when month(MEZERA_OD) <= 11 and month(MEZERA_DO) > 11 then 1 else 0
end)) as nezam_11
,sign(sum(case when month(MEZERA_OD) <= 12 and (month(MEZERA_DO) = 12 and
day(MEZERA_DO) = 31) then 1 else 0 end)) as nezam_12

,sum(case when month(MEZERA_OD) = 1 AND (skr_zamestnanec >= 1) then 1 else 0 end) as
vstup_1
```

```
,sum(case when month(MEZERA_OD) = 2 then 1 else 0 end) as vstup_2
,sum(case when month(MEZERA_OD) = 3 then 1 else 0 end) as vstup_3
,sum(case when month(MEZERA_OD) = 4 then 1 else 0 end) as vstup_4
,sum(case when month(MEZERA_OD) = 5 then 1 else 0 end) as vstup_5
,sum(case when month(MEZERA_OD) = 6 then 1 else 0 end) as vstup_6
,sum(case when month(MEZERA_OD) = 7 then 1 else 0 end) as vstup_7
,sum(case when month(MEZERA_OD) = 8 then 1 else 0 end) as vstup_8
,sum(case when month(MEZERA_OD) = 9 then 1 else 0 end) as vstup_9
,sum(case when month(MEZERA_OD) = 10 then 1 else 0 end) as vstup_10
,sum(case when month(MEZERA_OD) = 11 then 1 else 0 end) as vstup_11
,sum(case when month(MEZERA_OD) = 12 then 1 else 0 end) as vstup_12
into ##DB_temp_mesice
from ##DB_temp_doplneniMezer group by ID
GO
```

3.4 Integrace a formátování

V praxi je častá situace, kdy zkoumaná data nepochází pouze z jednoho zdroje. Pokud existuje jednoznačný identifikační klíč, je možné data spojit. Posledním krokem před vložením upravených dat do modelu je jejich formátování, neboť v mnoha případech je vhodné do modelu nahrát například setříděná data.

3.4.1 Cíl

Cílem této fáze je

- Vytvoření nových záznamů nebo hodnot kombinací informací z více zdrojů
- Syntaktické úpravy, které zachovávají význam dat

3.4.2 Výstup

Výstupem této fáze je upravený dataset. Tento soubor obsahuje:

- Sloučená data z dvou nebo více zdrojů
- Uspořádaná data, například
    - Setřídění pořadí záznamů (řádků)
    - Setřídění pořadí atributů (sloupců), například
        - Každý záznam bude začínat jednoznačným identifikátorem

3.4.3 Činnost

Pokud data pocházejí z více zdrojů, je nezbytné pro integraci dat do jednoho souboru

- Zkontrolovat, jestli je možné integraci vůbec provést
    - Existence jednoznačného klíče
    - Zpracovatelná velikost výsledné databáze
- Uložit výsledky integrace
    - Nutné uložení integrovaného souboru z důvodu
        - Opakování procesu modelování
        - Úprava výběru dat v budoucnosti, například
            - Po změně formátu zdrojových databází
            - Po změně procesu modelování
- Znovu zvážit výběr dat
    - Zahnutí dříve vyloučených hodnot
    - Vyloučení hodnot dříve zahrnutých ve výběru

Při kombinování informací z více zdrojů existují dva různé přístupy

- Slučování dat
  - Kombinace dvou souborů se stejnými záznamy a různými atributy
  - Výsledkem je soubor obsahující více sloupců
- Připojování dat
  - Kombinace dvou souborů se stejnými atributy a různými záznamy
  - Výsledkem je soubor obsahující více řádků
  - Spojení probíhá na základě podobných polí, například podle
    - Identifikačního čísla osoby

Pokud je pro účely následného modelování potřeba, aby data měla předem definovanou strukturu, existují následující možnosti

- Přeuspořádání atributů
  - Změna pořadí sloupců
  - Pro účely modelování je vhodné, aby všechny záznamy měly na prvním místě například
    - Identifikační číslo osoby
    - Výši primárního / odvozeného důchodu
    - Poštovní směrovací číslo
    - Rok přiznání důchodu
- Změna pořadí záznamů
  - Pro následné modelování je často vhodné změnit pořadí záznamů v databázi, například podle
    - Data narození
    - Výše primárního / odvozeného důchodu
- Přeformátování hodnot

3.4.4 Doporučení

Během formátování dat je důležité využít všechny známé informace a zkušenosti.

3.4.5 Specifický příklad

Také specifický příklad na integraci dat zaměříme, jako v předchozích případech, na přípravu tvorby matice přechodu ze stavu zaměstnaný do neaktivní. Budeme se zabývat především propojením všech tří v současnosti dostupných individuálních datových zdrojů, tedy STATMIN ANOD, STATMIN VZ a INEP. Tato část navazuje na některé příklady v předchozích kapitolách (především v částech věnovaných výběru dat a jejich transformaci), neboť tyto úlohy spolu úzce souvisí.

V prvním kroku vyjdeme z dříve připravené dočasné tabulky `##DB_temp_INEP_cast` a doplníme ji informacemi z databáze STATMIN ANOD z předchozího roku (viz následující skript) na základě funkce `left join`, neboť je potřeba mít pro každý záznam k dispozici informaci, jaký je jeho zdravotní stav.

```
IF object_id('tempdb.dbo.##DB_temp_spojeni_cast') is NOT NULL DROP TABLE
##DB_temp_spojeni_cast
select A.*
      ,B.[rok_priznani]
      ,B.[pohlavi_anod]
      ,B.[typ_primarniho]
      ,B.[typ_odvozeneho]
```

```
into ##DB_temp_spojeni_cast
from [##DB_temp_INEP_cast] A
          left join [ANOD_predchozi] B on A.ID=B.ID
GO
```

Podobným způsobem doplníme informace o stavu invalidity jedinců ke konci aktuálního roku opět z databáze STATMIN ANOD. Před propojením je nutné přejmenovat sloupce této databáze, neboť slučujeme všechny informace do jedné tabulky, kde nesmí být dva sloupce se stejným jménem. Proveďme přejmenování např. následujícím způsobem

```
IF object_id('tempdb.dbo.##DB_temp_ANOD_soucasny') is NOT NULL DROP TABLE
##DB_temp_ANOD_soucasny
select ID ,typ_primarniho as typ_primarniho_konec
          ,typ_odvozeneho as typ_odvozeneho_konec
into ##DB_temp_ANOD_soucasny
from [ANOD_soucasny]
```

V dalším kroku doplníme z této nově vytvořené tabulky zbytek potřebných informací k předchozímu částečnému propojení

```
IF object_id('tempdb.dbo.##DB_temp_spojeni') is NOT NULL DROP TABLE ##DB_temp_spojeni
select A.*
      ,B.[typ_primarniho_konec]
      ,B.[typ_odvozeneho_konec]
into ##DB_temp_spojeni
from [##DB_temp_spojeni_cast] A
          left join ##DB_temp_ANOD_soucasny B on A.ID=B.ID
GO
```

Vraťme se dále k podkapitole o transformaci dat, na jejímž závěru jsme vytvořili dočasnou tabulku obsahující mezery v zaměstnanosti a dále pomocí nich odvodili několik dalších atributů. Pro tento účel jsme provedli propojení s tabulkou ##DB_temp_marginalni, ve které je uložena část databáze INEP pro vybraný rok doplněna o informace o změnách zdravotních stavů. Toto doplnění je možné provést např. následujícím příkazem

```
if object_id('tempdb.dbo.##DB_temp_doplneniMezer') is NOT NULL drop table
##DB_temp_doplneniMezer
select A.*
          ,B.[mezera_OD]
          ,B.[mezera_DO]
into ##DB_temp_doplneniMezer
from ##DB_temp_marginalni A
          left join ##DB_temp_mezery B on A.ID=B.ID
GO
```

Použitím posledního skriptu jsme tak využili všechny v současnosti dostupné individuální databáze a ukázali, jak je možné propojit všechny potřebné informace z těchto datových zdrojů do jedné souhrnné tabulky, se kterou je možné dále pracovat.

4. Závěr

V tomto dokumentu jsme popsali metodické postupy pro zjištění a zajištění potřebné datové kvality jakožto nutného předpokladu pro vytvoření klíčových podmínek k rozvoji mikrosimulačního modelu důchodového pojištění.

Navržené metodické postupy jsme demonstrovali na praktických příkladech v prostředí MS SQL, během kterých se podařilo odhalit některé zjevné nedostatky v datové kvalitě dostupných podkladových zdrojů (databáze INEP, STATMIN VZ a STATMIN ANOD). Pro některé z nalezených nedostatků bylo již v rámci praktických příkladů možné nalézt vhodné možnosti řešení úpravy datového podkladu a zvýšení datové kvality. Nicméně i přes tuto skutečnost bylo zjevné, že datová kvalita dostupných datových zdrojů není ideální. Proto doporučujeme otevření diskuze na úrovni zdrojů dat s jejich dodavateli směrem k zajištění a přípravě potřebných náprav přímo u zdroje.

Zároveň chápeme, že některé datové zdroje byly primárně vytvořeny za jiným účelem než následnému využití v mikrosimulačním modelu, např. databáze INEP bylo určena primárně jako jednotný datový podklad pro tvorbu modelpointů a tudíž je vytvořena transformací vstupních dat ČSSZ, které tento účel postihují. Nicméně pro odvození jiných vstupů používaných v rámci modelu není tato transformace vhodná – např. není možné zjistit konkrétní období v rámci daného roku, kdy byl konkrétní modelpoint aktivním OSVČ. V rámci diskuze na úrovni zdrojů dat tak doporučujeme také otevření otázky vstupního datového modelu na nižší úrovni než pro výsledné databáze (INEP, STATMIN VZ, STATMIN ANOD), pro zajištění možnosti získání vstupních dat, případně částečně transformovaných vstupních dat, které je poté možné dále zpracovat a transformovat přímo v prostředí MPSV.

V rámci sběru dat a jejich uložení do MS SQL prostředí jsme dále narazili na nutnost úprav podkladových dat tak, aby byly převedeny do formátu dále využitelného pro jejich další využití. Tyto úpravy byly často ne zcela triviálního charakteru, a vyžadovaly několika krokovou transformaci počínaje vlastním loadem dat, přes jejich přetypování atd. Vzhledem k velkému rozsahu a objemu podkladových dat (zejména v databázi INEP), jsme pak pro následnou efektivní práci s datovými podklady identifikovali potřebu implementace efektivního datového modelu podpořeného využitím vhodných databázových konstruktů a nástrojů, např. efektivní indexace tabulek a odpovídajících ETL procesů.

Tyto nálezy považujeme za klíčové, a domníváme se, že bez implementace kvalitního datového modelu nebude jednoduché dosáhnout stavu, ve kterém nově dostupné datové zdroje budou využitelné pro další účely. Implementace datového modelu by měla pokrývat:

- Identifikaci vhodné úrovně vstupních dat přímo na datových zdrojích ČSSZ na základě datového požadavků MPSV
- Úpravu datových výstupů ČSSZ dle tohoto požadavku
- Nastavení databázových jobů pro plnění vstupních dat do iniciálních tabulek
- Nastavení následných procedur pro zpracování a transformaci vstupních dat
- Prověření datové kvality a automatické zpracování nápravných mechanismů přímo nad transformovanými daty
- Vytvoření koncové vrstvy (datamartu) s transformovanými datovými zdroji dle potřeb a specifikace MPSV
- Zajištění potřebného reportingu

Implementace datového modelu a automatizovaných rutin nicméně sama o sobě není finální odpovědí na otázku datové kvality a přípravy dat pro další využití. Vzhledem k tomu, že celý proces CRISP-DM je svou povahou cyklický, je na základě získaných zkušeností ze zpracování a následného využití dat nezbytné nabyté informace zohlednit a využít k efektivnějšímu zpracování dat v budoucnosti, a případné úpravě mechanismu zpracování dat. Úpravou vhodné úrovně datových výstupů ČSSZ je možné dosáhnout kontroly nad následnými transformacemi přímo v prostředí MPSV.

EY | Assurance | Tax | Transactions | Advisory

Informace o EY

EY je předním celosvětovým poskytovatelem odborných poradenských služeb v oblasti auditu, daní, transakčního a podnikového poradenství. Znalost problematiky a kvalita služeb, které poskytujeme, přispívají k posilování důvěry v kapitálové trhy i v ekonomiky celého světa. Výjimečný lidský a odborný potenciál nám umožňuje hrát významnou roli při vytváření lepšího prostředí pro naše zaměstnance, klienty i pro širší společnost.

Název EY zahrnuje celosvětovou organizaci a může zahrnovat jednu či více členských firem Ernst & Young Global Limited, z nichž každá je samostatnou právnickou osobou. Ernst & Young Global Limited, britská společnost s ručením omezeným garancí, služby klientům neposkytuje. Pro podrobnější informace o naší organizaci navštivte prosím naše webové stránky ey.com.

ey.com