Univerzita Karlova v Praze

Matematicko-fyzikální fakulta

BAKALÁŘSKÁ PRÁCE

Lukáš Drápal

Problém shody narozenin

Katedra pravděpodobnosti a matematické statistiky

Vedoucí bakalářské práce:	prof. RNDr. Jiří Anděl, DrSc.
Studijní program:	Matematika
Studijní obor:	Obecná matematika

Praha 2011

Děkuji prof. RNDr. Jiřímu Andělovi, DrSc. za vedení práce a cenné připomínky k její podobě. Dále děkuji Mgr. Janovi Jeronýmovi Zvánovcovi za pomoc s grafickou úpravou v LaTeXu.

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

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

V Praze dne 4. 8. 2011 Lukáš Drápal

Název práce: Problém shody narozenin

Autor: Lukáš Drápal

Katedra: Katedra pravděpodobnosti a matematické statistiky

Vedoucí bakalářské práce: prof. RNDr. Jiří Anděl, DrSc., Katedra pravděpodobnosti a matematické statistiky

Abstrakt: V bakalářské práci popisujeme problém shody narozenin za situace, kdy pravděpodobnosti narození nejsou stejné. Nejprve seznamujeme čtenáře s koncepty majorizace vektorů, schurovské konvexity funkcí a Bellových polynomů. Tyto partie využíváme při řešerši článků [6] a [8]. Uvádíme i obsah článku [7] a popisujeme chybu, které se v něm autor dopustil. Dále představujeme program v jazyku R, kterým simulujeme problém shody narozenin. Uvádíme výsledky, které jsem programem získali z dat narození obyvatel České republiky. Na závěr uvádíme některé aplikace problému, zejména problém shody příjmení v Japonsku, který je popsán v článku [8].

Klíčová slova: problém shody narozenin, problém shody příjmení, Bellovy polynomy, majorizace vektorů

Title: Birthday problem

Author: Lukáš Drápal

Department: Department of Probability and Mathematical Statistics

Supervisor: prof. RNDr. Jiří Anděl, DrSc., Department of Probability and Mathematical Statistics

Abstract: In the presented work we discuss the birthday problem with unlike probabilities. First, we introduce the concept of majorization of vectors, Schur-convexity of functions and Bell polynomials. Using these concepts we show the results from papers [6] and [8]. We also discuss the paper [7] and we point out its error. Then we present a program in language R that is simulating the problem. We use this program to calculate the probability for the true birthday problem in the Czech Republic and the effect of leap years. Finally, we show some applications of the birthday problem, especially the true surname problem in Japan [8].

Keywords: birthday problem, surname problem, Bell polynomials, majorization of vectors

Obsah

Úvod **2**

1 Obecné poznatky **4**
1.1 Poznatky z konvexní analýzy 4
1.2 Poznatky z kombinatoriky 7

2 Výpočet pravděpodobnosti a její aproximace **10**
2.1 Formulace problému 10
2.2 Základní problém a jeho aproximace 10
2.3 Obecný problém a jeho vztah k základnímu problému 11
2.4 Aproximace pomocí rozdílu 13
2.5 Konvergence $T(n, s, \mathbf{p})$ 16
2.6 Výpočet pravděpodobnosti pomocí Bellových polynomů 23

3 Programové vybavení **27**
3.1 Testování hypotézy o stejných pravděpodobnostech 27
3.2 Metody výpočtu 28
3.3 Funkce `narozeniny` 29
3.4 Zpracování dat 30

4 Aplikace problému **34**
4.1 Problém shody příjmení 34

Závěr **37**

Seznam použité literatury **38**

Přílohy **41**

Úvod

V *problému shody narozenin* nás zajímá, jaká je pravděpodobnost, že ve třídě neexistuje dvojice žáků se stejnými narozeninami. Když jsem představoval téma své bakalářské práce svým kamarádům, ptal jsem se jich, kolik si myslí, že je tato pravděpodobnost ve třídě o 25 žácích. Vždy se mi dostalo odpovědi, že tuto pravděpodobnost považují za velmi vysokou. Častá odpověď byla, že se pravděpodobnost blíží jedné. Ve skutečnosti je pravděpodobnost jen asi 0,43.

Většinou jsem se kamarádů ptal spíše na doplňkovou pravděpodobnost, že ve třídě *existuje* dvojice žáků se stejnými narozeninami. Jedním z důvodů, proč lidé odhadovali tuto pravděpodobnost velmi nízko, bylo, že vztáhli situaci pouze na sebe. Uvažovali tedy pravděpodobnost, že někdo má stejné narozeniny jako oni. Stejné narozeniny však může mít i dvojice ostatních lidí. Těchto dvojic je v uvažované třídě velmi mnoho. Protože se skutečná pravděpodobnost značně liší od intuitivního očekávání, nazývá se *problém shody narozenin* také *narozeninový paradox*.

Pokud předpokládáme, že pravděpodobnosti narození žáků jsou pro každý den roku stejné, má *problém* jednoduché řešení. Ukazuje se ovšem, že v průběhu roku se počty narození liší. V České republice se nejvíce lidí rodí v dubnu a v květnu. Odchylky v počtu narození jsou dost velké na to, abychom zamítli hypotézu o tom, že pravděpodobnosti narození jsou pro každý den v roce stejné. V práci popíšeme způsob, jak lze spočítat hledanou pravděpodobnost za situace, kdy pravděpodobnosti narození žáků nejsou stejné pro každý den v roce. Uveďme nyní obsah práce a vlastní přínos k tématu.

V první kapitole se seznámíme s obecnými poznatky z konvexní analýzy a kombinatoriky, které využijeme v dalším textu. Jedná se konkrétně o téma *majorizace* vektorů a s tím spojené *schurovské konvexity* funkcí a dále o téma *Bellových polynomů* a *exponenciálních vytvořujících funkcí*.

Když jsem se zabýval tématem *majorizace* vektorů, našel jsem v literatuře jinou definici, než jakou uváděl autor jednoho z článků, o kterých práce referuje. Ekvivalence definic nebyla na první pohled zřejmá. Tématem jsem se hlouběji zabýval a podařilo se mi ekvivalenci definic dokázat. Protože mi tento důkaz pomohl problematice *majorizace* vektorů lépe porozumět, rozhodl jsem se ho zařadit do textu práce. V první kapitole jsem rovněž vytvořil vlastní příklady *Bellových polynomů* a *majorizace* vektorů.

Ve druhé kapitole přesně formulujeme *problém shody narozenin* a krátce popisujeme řešení a odhady hledané pravděpodobnosti za situace, kdy pravděpodobnosti narození jsou stejné pro každý den v roce. Dále uvádíme obsah tří článků ze zadání bakalářské práce. Nejprve ukazujeme, že pravděpodobnost neexistence dvojice se stejnými narozeninami je nejvyšší za situace, kdy jsou pravděpodobnosti narození pro každý den stejné. Zde vycházíme ze článku dvojice autorů Joag-Dev a Proschan [6]. V důkazu využíváme znalost *Schurovské konvexity* funkcí.

Dále podle článku Nunnikhovena [7] uvádíme způsob, jak lze hledanou pravděpodobnost najít pomocí rozdílu skutečných a stejných pravděpodobností narození. Druhý uvedený způsob výpočtu hledané pravděpodobnosti vychází ze článku Mase [8]. Tento výpočet je založen na reprezentaci hledané pravděpodobnosti pomocí *Bellových polynomů* a využití vztahů, které pro ně platí.

Ve druhé kapitole jsem samostatně spočítal hodnoty hledané pravděpodobnosti pro stejné pravděpodobnosti narození žáků a různý počet žáků ve třídě. Spočítal jsem i hodnoty jednotlivých odhadů této pravděpodobnosti a odhady porovnal. Ve článcích Joag-Dev a Proschan [6] a Mase [8] jsem opravil velké množství drobných chyb a podrobně rozvedl a zdůvodnil jednotlivé kroky. Uvedl jsem i vlastní příklad aplikace *problému shody narozenin*, ve kterém počet dnů roku není konečný. Tato aplikace je uvedena jako motivace pro článek [8], který předpokládá obecně nekonečný počet dnů v roce.

Hlavním vlastním přínosem v druhé kapitole a i celé práce je však objevení závažné chyby ve článku Nunnikhovena [7]. Ve článku se tvrdí, že výrazy $T(n, s, \mathbf{p})$ rychle konvergují k nule. To by umožňovalo aproximaci pravděpodobnosti neexistence dvojice žáků se stejnými narozeninami pomocí součtu několika prvních výrazů $T(n, s, \mathbf{p})$. Konvergence výrazů $T(n, s, \mathbf{p})$ ovšem platí jen pro některé pravděpodobnosti narození žáků.

Nalezl jsem příklad pravděpodobnosti narození žáků, pro který výrazy $T(n, s, \mathbf{p})$ nekonvergují. Pro tento příklad jsem našel jednoduchý předpis pro hodnoty $T(n, s, \mathbf{p})$. Hodnoty, které by podle článku měly být blízko nuly, jsou rovny téměř jednomu miliónu. Dále jsem vytvořil program, který počítá $T(n, s, \mathbf{p})$ pro libovolné pravděpodobnosti narození žáků. Program jsem využil k výpočtu $T(n, s, \mathbf{p})$ pro mnoho variant pravděpodobnosti narození žáků.

Ve třetí kapitole představujeme program, který *problém shody narozenin* simuluje. Nejprve však zmiňujeme metody testování hypotézy o stejných pravděpodobnostech. Chceme totiž, aby program testoval, zda jsou pravděpodobnosti narození stejné pro všechny dny. Program rovněž počítá pravděpodobnost neexistence dvojice žáků se stejnými narozeninami pomocí metody ze článku Mase [8].

Program, kterým simulujeme *problém shody narozenin*, je sepsán v jazyku R. Zpracovali jsem jím data narození obyvatel České republiky. Zajímal nás vliv toho, že neuvažujeme stejné pravděpodobnosti narození, ale odhady pravděpodobnosti narození získané z českých dat. Rozdíl v pravděpodobnostech, že ve třídě neexistuje dvojice žáků se stejnými narozeninami, je pro nepřestupný rok menší než 0,0015. Dále jsme zjistili, že vliv 29. únoru na hledanou pravděpodobnost je v České republice nižší než 0,0005.

Ve čtvrté, závěrečné kapitole uvádíme některé aplikace *problému shody narozenin*. Zejména se jedná o *problém shody příjmení*. Pro situaci v Japonsku ho popsal Mase ve svém článku [8]. Při této příležitosti uděláme i velmi krátkou exkurzi do japonského jazyka, který jsem několik let studoval.

1. Obecné poznatky

V této kapitole uvedeme obecné poznatky, které využijeme v dalším textu.

1.1 Poznatky z konvexní analýzy

Koncept *majorizace vektorů* a *schurovské konvexity funkcí* se objevuje v rozličných partiích statistiky i optimalizace - od statistické mechaniky po modelování přenosu nemocí (Arnold [1]). Uvedeme jejich definice a pojmy objasníme na příkladech.

Definice 1. Řekneme, že vektor

$$\mathbf{a} = (x_1, \ldots, x_{i-1}, a_i, x_{i+1}, \ldots, x_{j-1}, a_j, x_{j+1} \ldots, x_n) \in \mathbb{R}^n$$

je *zaokrouhlením* vektoru

$$\mathbf{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n,$$

pokud $a_i + a_j = x_i + x_j$ a zároveň $|a_i - a_j| \leq |x_i - x_j|$.

Řekneme, že vektor

$$\mathbf{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$$

majorizuje vektor

$$\mathbf{y} = (y_1, y_2, \ldots, y_n) \in \mathbb{R}^n,$$

což budeme značit $\mathbf{x} \overset{m}{\geq} \mathbf{y}$, pokud existuje posloupnost vektorů

$$\mathbf{b_1}, \mathbf{b_2}, \ldots, \mathbf{b_m},$$

kde $m \in \mathbb{N}$, $\mathbf{b_1} = \mathbf{x}$, $\mathbf{b_m} = \mathbf{y}$ a $\mathbf{b_i}$ je *zaokrouhlením* vektoru $\mathbf{b_{i-1}}$ pro všechna $i = \{2, 3, \ldots, m\}$.

Někteří autoři „$\mathbf{x}$ majorizuje $\mathbf{y}$“ značí jako $\mathbf{x} \succ \mathbf{y}$. V této práci se však přidržíme značení $\mathbf{x} \overset{m}{\geq} \mathbf{y}$.

Tvrzení 1. *Nechť jsou dány vektory* $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ *a* $\mathbf{y} = (y_1, y_2, \ldots, y_n)$. *Nechť* $x_{(1)}$ *(respektive* $y_{(1)}$*) značí největší složku vektoru* $\mathbf{x}$ *(respektive* $\mathbf{y}$*),* $x_{(2)}$ *(respektive* $y_{(2)}$*) značí druhou největší složku* $\mathbf{x}$ *(respektive* $\mathbf{y}$*), a tak dále. Vektor* $\mathbf{x}$ majorizuje $\mathbf{y}$ *právě tehdy, když*

$$\sum_{j=1}^{n} x_{(j)} = \sum_{j=1}^{n} y_{(j)} \text{ a zároveň } \sum_{j=1}^{k} x_{(j)} \geq \sum_{j=1}^{k} y_{(j)}, \ k = 1, \ldots n-1.$$

Důkaz. Předpokládejme nejprve, že $\mathbf{x}$ majorizuje $\mathbf{y}$. Stačí ukázat, že *zaokrouhlení* vektoru zachovává konstantní součet všech jeho složek (zřejmé) a že nezvyšuje součet jeho k největších složek pro všechna $k < n$. Nechť bez újmy na obecnosti $x_i \geq x_j$ a $a_i \geq a_j$. Platí rovnost $x_i + x_j = a_i + a_j$, a tak můžeme k levé straně nerovnosti $a_i - a_j \leq x_i - x_j$ přičíst $a_i + a_j$ a k pravé straně přičíst $x_i + x_j$. Máme

$2a_i \leq 2x_i$, proto platí $a_i \leq x_i$. Probereme nyní všechny možnosti, jak mohou x_i, x_j (respektive a_i, a_j) spadat do k největších složek vektoru $\mathbf{x}$ (respektive $\mathbf{a}$).

Nechť nejprve x_i nespadá do $k \in \{1, \ldots, n-1\}$ největších složek vektoru $\mathbf{x}$, kde k je libovolné pevné. Protože $a_j \leq a_i \leq x_i$ a $x_j \leq x_i$, tak do k největších složek vektoru $\mathbf{a}$ nespadají ani a_i, a_j, x_j. Nerovnost $\sum_{j=1}^{k} x_{(j)} \geq \sum_{j=1}^{k} a_{(j)}$ tedy platí.

Nechť nyní x_i spadá do k největších složek vektoru $\mathbf{x}$, ale a_j nespadá do k největších složek vektoru $\mathbf{a}$. Pak buď a_i do součtu k největších složek vektoru $\mathbf{a}$ spadá, díky nerovnosti $x_i \geq a_i$ je součet k největších složek vektoru $\mathbf{x}$ vyšší nebo roven součtu k největších složek vektoru $\mathbf{a}$. Anebo do součtu nespadá a spadá do něj nějaká další složka vektoru $\mathbf{x}$ (a tedy i vektoru $\mathbf{a}$), která je menší nebo rovna x_i, a tak požadovaná nerovnost opět platí.

Konečně nechť do součtu k největších složek vektoru $\mathbf{x}$ spadá x_i a zároveň a_j spadá do součtu k největších složek vektoru $\mathbf{a}$. Víme, že $a_i \geq a_j$, a tak do součtu k největších složek vektoru $\mathbf{a}$ spadá i a_i. Buď do součtu složek $\mathbf{x}$ spadá i x_j a díky rovnosti $x_i + x_j = a_i + a_j$ je součet k největších složek vektoru $\mathbf{x}$ a $\mathbf{a}$ stejný. Anebo existuje složka $x_l,\ 1 \leq l \leq n,\ l \neq i, j$ vektoru $\mathbf{x}$, která je vyšší než x_j, a tak součet k největších složek vektoru $\mathbf{x}$ je vyšší než součet k největších složek vektoru $\mathbf{a}$, neboť $x_i + x_l > a_i + a_j$.

Ukázali jsme, že nerovnost $\sum_{j=1}^{k} x_{(j)} \geq \sum_{j=1}^{k} a_{(j)}$ platí pro všechny možnosti, jak mohou x_i, x_j (respektive a_i, a_j) spadat do k největších složek vektoru $\mathbf{x}$ (respektive $\mathbf{a}$). Proto požadovaná nerovnost platí pro všechna $k < n$.

Obráceně, nechť vektory $\mathbf{x}, \mathbf{y}$ splňují podmínku tvrzení. Pak buď platí $x_{(i)} = y_{(i)}$ pro všechna $i \in \{1, \ldots, n\}$, v takovém případě je okamžitě splněna definice 1. Anebo existují $i, j \in \{1, \ldots, n\}$, že $x_{(i)} > y_{(i)}$ a zároveň $x_{(j)} < y_{(j)}$. Navíc ukážeme, že lze najít taková i, j, že $x_{(i)} > x_{(j)}$. Pro spor předpokládejme, že taková i, j nalézt nelze. To znamená, že

$$x_{(j)} < y_{(j)} \ \wedge\ \forall k < j:\ x_{(k)} \leq y_{(k)}.$$

Pak ovšem platí

$$\sum_{k=1}^{j} x_{(k)} < \sum_{k=1}^{j} y_{(k)},$$

což je spor s podmínkou tvrzení.

Máme

$$i, j \in \{1, \ldots, n\},\ \text{že}\ x_{(i)} > y_{(i)}\ \wedge\ x_{(j)} < y_{(j)}\ \wedge\ x_{(i)} > x_{(j)}.$$

Označme $a_i := y_{(i)},\ a_j := x_{(i)} + x_{(j)} - y_{(i)}$. Nechť $x_{(i)}$ (respektive $x_{(j)}$) je v pořadí (nikoliv podle velikosti) s-tá (respektive t-tá) složka vektoru $\mathbf{x}$, bez újmy na obecnosti nechť $s < t$. Ukážeme, že vektor

$$\mathbf{a} = (x_1, \ldots, x_{s-1}, a_i, x_{s+1}, \ldots, x_{t-1}, a_j, x_{t+1} \ldots, x_n)$$

je zaokrouhlením vektoru $\mathbf{x}$. Platí

$$a_i + a_j = y_{(i)} + x_{(i)} + x_{(j)} - y_{(i)} = x_{(i)} + x_{(j)}.$$

Dále platí

$$|a_j - a_i| = |x_{(i)} + x_{(j)} - 2y_{(i)}| = |x_{(j)} - x_{(i)} + 2(x_{(i)} - y_{(i)})|.$$

Z nerovnosti $x_{(i)} > x_{(j)}$ plyne $x_{(j)} - x_{(i)} < 0$. Víme, že $x_{(i)} > y_{(i)}$, z čehož plyne $x_{(i)} - y_{(i)} > 0$. Zjistili jsme, že $|x_{(j)} - x_{(i)}|$ je rovno absolutní hodnotě záporného čísla a $|a_j - a_i|$ je rovno absolutní hodnotě stejného záporného čísla bez nějakého kladného čísla. Odtud plyne $|x_{(j)} - x_{(i)}| > |a_j - a_i|$. Vektor $\mathbf{a}$ je tedy *zaokrouhlením* vektoru $\mathbf{x}$.

Seřaďme složky vektorů $\mathbf{a}, \mathbf{x}, \mathbf{y}$ podle velikosti. Vidíme, že ve všech případech, kdy jsou si složky vektorů $\mathbf{x}$ a $\mathbf{y}$ rovny, jsou si rovny i složky vektorů $\mathbf{a}$ a $\mathbf{y}$. Navíc existuje alespoň jedna složka vektoru $\mathbf{a}$, která se shoduje se složkou vektoru $\mathbf{y}$, ale nikoliv se složkou vektoru $\mathbf{x}$. Složek vektoru $\mathbf{a}$, které se shodují s vektorem $\mathbf{y}$, je tedy alespoň o jedna více než složek vektoru $\mathbf{x}$, které se shodují s vektorem $\mathbf{y}$.

Počet složek vektorů $\mathbf{x}$ a $\mathbf{y}$, které se liší, je jen konečně mnoho. Z důkazu opačné implikace víme, že *zaokrouhlení* vektoru zachovává celkový součet a nezvyšuje součet jeho k největších složek pro všechna $k < n$. Proto $\mathbf{a}$ splňuje podmínku tvrzení. A tak můžeme opakovat tento postup a získat pomocí konečného počtu *zaokrouhlení* vektoru $\mathbf{x}$ vektor, který bude mít složky stejné hodnoty jako vektor $\mathbf{y}$, jen jeho složky nejsou nutně ve stejném pořadí. Transpozice dvou složek splňuje definici *zaokrouhlení*. Libovolnou permutaci konečného vektoru lze získat pomocí konečně mnoho transpozic, takže vskutku $\mathbf{x}$ majorizuje $\mathbf{y}$. □

Příklad 1. *Majorizace* vektorů nezávisí na pořadí jejich složek. Platí nasledující řetězec *majorizací* vektorů:

$$(10, -1, 0) \overset{m}{\geq} (9, 0, 0) \overset{m}{\geq} (1, 2, 6) \overset{m}{\geq} (2, 4, 3) \overset{m}{\geq} (3, 3, 3).$$

Výčet vektorů mezi $(10, -1, 0)$ a $(3, 3, 3)$ není ani zdaleka kompletní. Platí však, že vektor $(3, 3, 3)$ *nemajorizuje* žádný trojrozměrný vektor se součtem složek 9 kromě sama sebe. Ukážeme, že například $(1, 2, 6) \overset{m}{\geq} (2, 4, 3)$ vyhovuje definici 1 i podmínce z tvrzení 1. Pro splnění definice 1 vytvoříme řetězec *zaokrouhlení*:

$$(1, 2, 6) \sim (1, 4, 4) \sim (2, 3, 4) \sim (2, 4, 3).$$

Splnění podmínky z tvrzení 1 ukazuje tabulka 1.1.

Tabulka 1.1: Součty největších složek vektorů

	Vektor $(1, 2, 6)$	Vztah	Vektor $(2, 4, 3)$
Největší složka	6	$\geq$	4
Součet dvou největších složek	$2 + 6 = 8$	$\geq$	$4 + 3 = 7$
Součet všech složek	$1 + 2 + 6 = 9$	$=$	$2 + 4 + 3 = 9$

Problematiku *majorizace* vektorů nám může přiblížit následující geometrická interpretace. Nechť Π značí množinu všech permutací množiny $\{1, \ldots, n\}$. Definujme množinu všech permutací souřadnic vektoru $\mathbf{x}$, tedy $M := \{x_{\pi(1)}, \ldots, x_{\pi(n)}; \pi \in \Pi\}$. Platí $\mathbf{x} \overset{m}{\geq} \mathbf{y}$ právě tehdy, když $\mathbf{y}$ leží v konvexním obalu množiny M.

Příklad 2. Uvažujme vektor $(0, 1)$. Existuje jediná neidentická permutace složek tohoto vektoru, a to $(1, 0)$. Konvexní obal bodů $(0, 1)$ a $(1, 0)$ je úsečka spojující tyto dva body. Tato úsečka je množina všech bodů $\{[x, 1 - x]; x \in [0, 1]\}$. A tak vektor $(0, 1)$ *majorizuje* všechny vektory tvaru $\{[x, 1 - x]; x \in [0, 1]\}$. Z definice konvexního obalu plyne, že pro libovolný vektor $\mathbf{v}$ z této množiny existují $\alpha_1 \geq 0, \alpha_2 \geq 0,\ \alpha_1 + \alpha_2 = 1$, pro které platí $\mathbf{v} = \alpha_1(0, 1) + \alpha_2(1, 0)$.

Definice 2. Funkce $g(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$ je *Schurovsky konkávní*, pokud z $\mathbf{x} \overset{m}{\geq} \mathbf{y}$ plyne $g(\mathbf{x}) \leq g(\mathbf{y})$. Funkce $g(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$ je *Schurovsky konvexní*, pokud z $\mathbf{x} \overset{m}{\geq} \mathbf{y}$ plyne $g(\mathbf{x}) \geq g(\mathbf{y})$.

Příklad 3. Příkladem *Schurovsky konvexní* funkce je maximová funkce $f : \mathbb{R}^n \to \mathbb{R}$, $f(\mathbf{x}) = x_{(1)}$. Pokud platí $\mathbf{x} \overset{m}{\geq} \mathbf{y}$, pak z tvrzení 1 a $k = 1$ plyne $x_{(1)} \geq y_{(1)}$, tedy $f(\mathbf{x}) \geq f(\mathbf{y})$.

Příkladem *Schurovsky konkávní* funkce je funkce $f : \mathbb{R}^n \to \mathbb{R}$ daná předpisem $f(\mathbf{x}) = \prod_{i=1}^n x_i$. To ukážeme pro $\sum_{i=1}^n x_i = 1$ v sekci 2.3.

1.2 Poznatky z kombinatoriky

Definice 3. *Částečné Bellovy polynomy* $B_{n,k}(x_1, x_2, \ldots, x_{n-k+1})$ proměnných $x_1, x_2, \ldots, x_n$ jsou definovány rovností

$$\exp\left(u \sum_{m=1}^{\infty} x_m \frac{t^m}{m!}\right) = 1 + \sum_{n=1}^{\infty} \left\{ \sum_{k=1}^{n} u^k B_{n,k}(x_1, \ldots, x_{n-k+1}) \right\} \frac{t^n}{n!}.$$

Úplné Bellovy polynomy $Y_n(x_1, \ldots, x_n)$ jsou definovány vztahem

$$\exp\left(\sum_{m=1}^{\infty} x_m \frac{t^m}{m!}\right) = 1 + \sum_{n=1}^{\infty} Y_n(x_1, \ldots, x_n) \frac{t^n}{n!}.$$

Pro $u = 1$ dostáváme porovnáním rovností z definic

$$Y_n(x_1, \ldots, x_n) = \sum_{k=1}^{n} B_{n,k}(x_1, \ldots, x_{n-k+1}).$$

Tvrzení 2. Částečné Bellovy polynomy *lze explicitně vyjádřit jako*

$$B_{n,k}(x_1, \ldots, x_{n-k+1}) = \sum_{\substack{a_1+\ldots+a_{n-k+1}=k \\ 1a_1+2a_2+\ldots+(n-k+1)a_{n-k+1}=n}} \frac{n!}{a_1! \ldots a_{n-k+1}!} \prod_{i=1}^{n-k+1} \left(\frac{x_i}{i!}\right)^{a_i}. \tag{1.1}$$

Důkaz. Viz (Roman [3], s. 82). □

Příklad 4. Zvolme $n = 5$. Najdeme *částečné Bellovy polynomy* pro $k = 1, 2, 3, 4, 5$ a jejich součtem získáme *úplný Bellův polynom.* Pro $k = 1$ dostáváme *částečný Bellův polynom*

$$B_{5,2}(x_1, x_2, x_3, x_4, x_5) = \sum_{\substack{a_1+a_2+a_3+a_4+a_5=1 \\ a_1+2a_2+3a_3+4a_4+5a_5=5}} \frac{5!}{a_1!a_2!a_3!a_4!a_5!} \prod_{i=1}^{5} \left(\frac{x_i}{i!}\right)^{a_i}.$$

Podmínkám

$$a_1 + a_2 + a_3 + a_4 + a_5 = 1 \quad \text{a} \quad a_1 + 2a_2 + 3a_3 + 4a_4 + 5a_5 = 5$$

vyhovuje jediné řešení, a to $a_5 = 1$. Dostáváme

$$B_{5,1}(x_1, x_2, x_3, x_4, x_5) = \frac{5!}{0!0!0!0!1!} \cdot \frac{x_5}{5!} = x_5.$$

Zvolme nyní $k = 2$. Dostáváme *částečný Bellův polynom*

$$B_{5,2}(x_1, x_2, x_3, x_4) = \sum_{\substack{a_1+a_2+a_3+a_4=2 \\ a_1+2a_2+3a_3+4a_4=5}} \frac{5!}{a_1!a_2!a_3!a_4!} \prod_{i=1}^{4} \left(\frac{x_i}{i!}\right)^{a_i}.$$

Podmínkám

$$a_1 + a_2 + a_3 + a_4 = 2 \quad \text{a} \quad a_1 + 2a_2 + 3a_3 + 4a_4 = 5$$

vyhovují dvě řešení, a totiž

$$a_1 = a_4 = 1, a_2 = a_3 = 0 \quad \text{a} \quad a_1 = a_4 = 0, a_2 = a_3 = 1.$$

Dostáváme

$$B_{5,2}(x_1, x_2, x_3, x_4) = \frac{5!}{1!0!0!1!} \cdot \frac{x_1x_4}{1!4!} + \frac{5!}{0!1!1!0!} \cdot \frac{x_2x_3}{2!3!} = 5x_1x_4 + 10x_2x_3.$$

Tyto výsledky mají elegantní kombinatorický význam. Číslo n značí, kolik má množina prvků, v našem případě se tedy jedná o pětiprvkovou množinu. Číslo k značí, na kolik neprázdných podmnožin chceme tuto množinu rozdělit. Hodnota polynomu $B_{n,k}(x_1, x_2, \ldots, x_{n-k+1})$ lze interpretovat následujícím způsobem. U každého členu polynomu hodnota indexů x_i značí, kolik mají prvků podmnožiny vzniklé dělením, a koeficient u členu polynomu značí, kolik existuje dělení pětiprvkové množiny na podmnožiny daných velikostí.

A tak pokud chceme pětiprvkovou množinu rozdělit na jednu podmnožinu, hodnota $B_{5,1}(x_1, x_2, x_3, x_4, x_5) = x_5$ sděluje zcela zřejmý fakt — existuje jediné dělení a to na pětiprvkovou (tedy původní) množinu. Pokud chceme pětiprvkovou množinu rozdělit na dvě neprázdné podmnožiny, z hodnoty $B_{5,2}(x_1, x_2, x_3, x_4) = 5x_1x_4 + 10x_2x_3$ můžeme vyčíst, že existuje 5 způsobů, jak rozdělit pětiprvkovou množinu na jednoprvkové a čtyřprvkové podmnožiny, a 10 způsobů, jak rozdělit tutéž množinu na dvouprvkové a tříprvkové podmnožiny.

Obdobným způsobem jako pro $k = 1, 2$ spočteme *částečné Bellovy polynomy* pro $k = 3, 4, 5$:

$$\begin{aligned}
B_{5,3}(x_1, x_2, x_3) &= \frac{5!}{1!2!0!} \cdot \frac{x_1x_2^2}{1! \cdot (2!)^2} + \frac{5!}{2!0!1!} \cdot \frac{x_1^2x_3}{(1!)^2 \cdot 3!} = 15x_1x_2^2 + 10x_1^2x_3, \\
B_{5,4}(x_1, x_2) &= \frac{5!}{3!1!} \cdot \frac{x_1^3x_2}{(1!)^3 \cdot 2!} = 10x_1^3x_2, \\
B_{5,5}(x_1) &= \frac{5!}{5!} \cdot \frac{x_1^5}{(1!)^5} = x_1^5.
\end{aligned}$$

Součtem všech *částečných Bellových polynomů* dostáváme *úplný Bellův polynom*

$$Y_5(x_1, \ldots, x_5) = x_5 + 5x_1x_4 + 10x_2x_3 + 15x_1x_2^2 + 10x_1^2x_3 + 10x_1^3x_2 + x_1^5.$$

Definice 4. Nechť $k, n \in \mathbb{N}, n \geq k$. Definujme *Stirlingovo číslo druhého druhu* $\left\{ {n \atop k} \right\}$ jako počet rozkladů množiny $\{1, \ldots, n\}$ na k neprázdných podmnožin.

Úvahy v předchozím příkladu nás vedou k zamyšlení nad souvislostí *částečných Bellových polynomů* a *Stirlingových čísel druhého druhu. Částečné Bellovy polynomy* oproti *Stirlingovým číslům druhého druhu* nesou nejen informaci o počtu neprázdných podmnožin, ale i o velikosti těchto podmnožin. Stačí však položit $x_1 = x_2 = \ldots = x_{n-k+1} = 1$ a toto dodatečná informace je odstraněna. A tak platí

$$B_{n,k}(1, 1, \ldots, 1) = \left\{ {n \atop k} \right\}.$$

Důkaz tohoto tvrzení lze najít například v (Comtet [2], s. 135).

Uveďme nyní velmi užitečný rekurentní vztah pro *úplné Bellovy polynomy.*

Tvrzení 3. *Pro* úplné Bellovy polynomy *platí rekurentní vztah*

$$Y_n(x_1, \ldots, x_n) = \sum_{i=1}^{n} \binom{n-1}{i-1} x_i Y_{n-i}(x_1, \ldots, x_{n-i}). \tag{1.2}$$

Důkaz. Viz (Roman [3], kapitola 4.1.8). □

Bellovy polynomy může být výhodné charakterizovat pomocí speciálních druhů vytvořujících funkcí. Vytvořující funkce nám nabízejí další početní techniku, po opětovném převedení na *Bellovy polynomy* můžeme získat výsledky, které by se jinak získávaly jen velmi těžko. Uvedeme definici jednoho speciálního druhu vytvořující funkce.

Definice 5. *Exponenciální vytvořující funkcí* posloupnosti $\{a_n\}_{n=0}^{\infty}$ míníme řadu

$$G(\{a_n\}_{n=0}^{\infty}; x) = \sum_{n=0}^{\infty} a_n \frac{x^n}{n!}.$$

2. Výpočet pravděpodobnosti a její aproximace

2.1 Formulace problému

Uvažujme třídu, ve které je n žáků, a rok, který má d dnů. Nejčastěji budeme uvažovat nepřestupný rok, tedy $d = 365$. Tvrzení ale budeme formulovat obecně. Zajímá nás, jaká je pravděpodobnost, že všichni žáci mají narozeniny v různý den. Tuto pravděpodobnost, že neexistuje dvojice žáků se stejným dnem a měsícem narození, označíme R_n.

Budeme předpokládat, že den a měsíc narození jednotlivých žáků jsou nezávislé jevy. Tento předpoklad nebývá vždy splněn. Například pokud jsou ve třídě sourozenci, je pravděpodobnost, že mají stejný den a měsíc narození, alespoň taková jako pravděpodobnost, že jsou sourozenci dvojčata[1]. Bez tohoto předpokladu by však výpočet pravděpodobnosti byl velmi obtížný. Navíc je v literatuře často uváděn zjednodušující předpoklad, že každý žák má pravděpodobnost narození pro každý den v roce stejnou. Za toho předpokladu budeme hovořit o *základním problému shody narozenin* (dále jen krátce *základním problému*), bez tohoto předpokladu budeme hovořit o *obecném problému shody narozenin* (dále jen *obecném problému*).

2.2 Základní problém a jeho aproximace

Nejprve spočteme pravděpodobnost R_n, $n \leq d$ pro *základní problém*. Je celkem d^n kombinací narozenin žáků ve třídě. Z toho takových kombinací, že všichni žáci mají narozeniny v různý den, je $d \cdot (d-1) \cdot (d-2) \cdot \ldots \cdot (d-n-1)$. Dostáváme

$$R_n = \frac{d \cdot (d-1) \cdot (d-2) \cdot \ldots \cdot (d-n-1)}{d^n} = \frac{d!}{d^n (d-n)!}.$$

Pro $n > d$ neexistuje možnost, aby všichni žáci ve třídě měli narozeniny v různý den, a tak v tomto případě je $R_n = 0$.

Určitou intuitivní představu nám může dát následující úvaha. Ve třídě existuje $\binom{n}{2}$ dvojic žáků. Pravděpodobnost, že daná dvojice žáků má narozeniny v různý den, je $(1 - \frac{1}{d})$. Proto R_n může být aproximováno jako

$$D_n := \left(1 - \frac{1}{d}\right)^{\binom{n}{2}}.$$

Výsledek není přesný, neboť jevy *dvojice žáků nemá narozeniny ve stejný den* nejsou nezávislé. Když totiž víme, že Alice a Bob nemají narozeniny ve stejný den, je pravděpodobnost, že Cyril nemá narozeniny ve stejný den jako Alice ani jako Bob, nižší než bez této informace.

[1] Podíl vícečetných porodů byl v letech 1991–2006 v ČR vyšší než 0,9 viz [4], tedy pravděpodobnost, že jsou sourozenci dvojčata, je více než $\frac{1}{365}$.

Pravděpodobnost R_n lze bez problémů a rychle spočítat, a tak hledat aproximace není z výpočetního hlediska nutné. Aproximace ale může pomoci v případě, kdy dostatečné výpočetní prostředky nemáme k dispozici.

Následující dva odhady uvádí (Feller [5], s. 33). Pokud při výpočtu R_n po roznásobení zanedbáme členy, které mají mocninu d nižší než $n-1$, dostaneme odhad

$$E_n := \frac{d^n - d^{n-1} \cdot [1 + 2 + \ldots + (n-1)]}{d^n} = 1 - \frac{n \cdot (n-1)}{2d},$$

který dobře odhaduje R_n jen pro malé hodnoty n.

Přesnější odhad dostaneme, pokud využijeme Taylorův rozvoj $\log(1-x)$:

$$\log(1-x) = -x - \frac{x^2}{2} - \frac{x^3}{3} \ldots$$

Pro malé hodnoty x můžeme aproximovat $\log(1-x)$ pomocí Taylorova polynomu prvního řádu $-x$. Pro $\log(R_n)$ dostáváme odhad:

$$\begin{aligned}
\log(R_n) &= \log\left[\frac{d \cdot (d-1) \cdot (d-2) \cdot \ldots \cdot (d-n-1)}{d^n}\right] \\
&= \log\left[1 \cdot \left(1 - \frac{1}{d}\right) \cdot \left(1 - \frac{2}{d}\right) \cdots \left(1 - \frac{n-1}{d}\right)\right] \\
&= \log\left(1 - \frac{1}{d}\right) + \log\left(1 - \frac{2}{d}\right) + \cdots + \log\left(1 - \frac{n-1}{d}\right) \\
&\approx -\frac{1}{d} - \frac{2}{d} - \cdots - \frac{n-1}{d} = -\frac{n \cdot (n-1)}{2d}.
\end{aligned}$$

R_n tedy odhadneme

$$R_n \approx e^{-\frac{n \cdot (n-1)}{2d}} := F_n.$$

Tabulka 2.1 ukazuje hodnoty R_n a odhadů D_n, E_n, F_n pro $d = 365$ a různé hodnoty n. Z tabulky vidíme, že odhad E_n je velmi nepřesný — s přesností 0,01 odhaduje jen do $n = 10$, od $n \geq 28$ udává nesmyslné, záporné odhady pravděpodobnosti. Avšak odhady D_n a F_n jsou poměrně přesné. Odhad F_n je vždy o velmi málo větší než odhad D_n. Je zajímavé, jak jsou si tyto dva odhady blízko i pro velká n.

2.3 Obecný problém a jeho vztah k základnímu problému

V *obecném problému* shody narozenin uvažujeme, že pravděpodobnost, že se libovolný žák ve třídě narodil v den i, je p_i, kde $i = 1, 2, \ldots, d$ a $\sum_{i=1}^{d} p_i = 1$. Pro *obecný problém*, $n \leq d$, $j, k \in \mathbb{N}, j, k \leq d$ spočteme R_n jako

$$R_n = \sum_{i_j \neq i_k \text{ pro } j \neq k} p_{i_1} \cdot p_{i_2} \cdot \ldots \cdot p_{i_n} = n! \sum_{i_1 < i_2 < \ldots < i_n} p_{i_1} \cdot p_{i_2} \cdot \ldots \cdot p_{i_n}. \tag{2.1}$$

První rovnost platí, neboť sčítáme přes všechny kombinace, kdy se žáci nenarodili ve stejný den, a pravděpodobnost průniku nezávislých jevů je násobek jejich pravděpodobností. Druhá rovnost platí, neboť je $n!$ způsobů, jak za sebou

Tabulka 2.1: Hodnoty R_n a jeho odhadů pro $d = 365$

n	R_n	D_n	E_n	F_n
5	0,9728	0,9729	0,9726	0,9730
10	0,8831	0,8839	0,8767	0,8840
15	0,7471	0,7497	0,7123	0,7500
20	0,5886	0,5938	0,4795	0,5942
25	0,4313	0,4391	0,1781	0,4396
30	0,2936	0,3032	-0,1917	0,3037
35	0,1856	0,1955	-0,6301	0,1959
40	0,1088	0,1177	-1,137	0,1180
45	0,0590	0,0661	-1,712	0,0664
50	0,0296	0,0347	-2,356	0,0349
100	$3{,}07 \cdot 10^{-7}$	$1{,}27 \cdot 10^{-6}$	-12,56	$1{,}29 \cdot 10^{-6}$
200	$1{,}61 \cdot 10^{-30}$	$1{,}95 \cdot 10^{-24}$	-53,52	$2{,}10 \cdot 10^{-24}$
300	$6{,}25 \cdot 10^{-82}$	$3{,}65 \cdot 10^{-54}$	-121,9	$4{,}32 \cdot 10^{-54}$
366	0	$2{,}60 \cdot 10^{-80}$	-182	$3{,}34 \cdot 10^{-80}$

$p_{i_1}, p_{i_2}, \ldots, p_{i_n}$ uspořádat. Pro $n > d$ neexistuje možnost, aby všichni žáci ve třídě měli narozeniny v různý den, a tak v tomto případě je $R_n = 0$.

Zajímá nás, jak pravděpodobnost R_n závisí na hodnotách $p_1, p_2, \ldots, p_d$. Hodnotu R_n můžeme chápat jako funkci proměnné $\mathbf{p} = (p_1, p_2, \ldots, p_d)$. Pravděpodobnost R_n při dané hodnotě $\mathbf{p}$ budeme značit $R_n(\mathbf{p})$.

Tvrzení 4. *Pravděpodobnost $R_n(\mathbf{p})$ pro* obecný problém *je nejvyšší, pokud platí $p_1 = p_2 = \ldots = p_d = \frac{1}{d}$, tedy za situace* základního problému. *Navíc $R_n(\mathbf{p})$ je* Schurovsky konkávní *funkce.*

Důkaz. Důkaz provedeme způsobem, jaký zmiňují Joag-Dev a Proschan [6].

Uvažujme nejprve následující jednoduchý model. Ve třídě jsou dva žáci, pravděpodobnost narození žáků v určitém období je rovna c, $c \in (0, 1]$. Pravděpodobnost narození v první polovině tohoto období je rovna p, $0 \leq p \leq c$ a pravděpodobnost narození v druhé polovině období je $c - p$. Pravděpodobnost narození v odlišných polovinách je $p(c-p) + p(c-p) = 2p(c-p)$. Tuto hodnotu můžeme chápat jako funkci p a podle p také zderivovat:

$$[2p(c-p)]' = 2(c-p) - 2p = 2c - 4p.$$

Tedy $2p(c-p)$ je rostoucí funkce na intervalu $p \in [0, \frac{c}{2}]$ a klesající na intervalu $p \in [\frac{c}{2}, c]$. Navíc je funkce $2p(c-p)$ symetrická kolem $\frac{c}{2}$. Vidíme, že čím větší je rozdíl p a $c - p$, tím nižší je hledaná pravděpodobnost.

Nyní se vraťme k *obecnému problému*, pro jednoduchost uvažujme $n = 3$. A tak pro $i, j, k \in \mathbb{N},\ i, j, k \in [1, d]$ platí

$$R_n(\mathbf{p}) = \sum_{j \neq k \wedge i \neq j,k} p_i \cdot p_j \cdot p_k.$$

Předpokládejme, že bez újmy na obecnosti je p_1 a p_2 pevně dané a platí $0 < p_1 \leq p_2 < 1$. Podívejme se na to, jak se hodnota funkce $R_n(\mathbf{p})$ změní, pokud změníme p_1 a p_2 tak, že zvýšíme rozdíl $p_2 - p_1$, ale zachováme součet $p_1 + p_2$. Členy součtu,

ve kterých není ani p_1, ani p_2, se nezmění. Ze členů, ve kterých je p_1, ale není p_2, můžeme vytknout p_1 a dostat

$$p_1 \cdot \sum_{i,j \geq 3 \wedge i \neq j} p_i \cdot p_j. \tag{2.2}$$

Ze členů, ve kterých je p_2, ale není p_1, můžeme vytknout p_2 a dostat

$$p_2 \cdot \sum_{i,j \geq 3 \wedge i \neq j} p_i \cdot p_j. \tag{2.3}$$

Sečtením (2.2) a (2.3) dostaneme

$$(p_1 + p_2) \cdot \sum_{i,j \geq 3 \wedge i \neq j} p_i \cdot p_j.$$

Součet $p_1 + p_2$ zůstal stejný, a tak součet (2.2) + (2.3) změna p_1 a p_2 neovlivní.

Změna $f(\mathbf{p})$ tedy závisí pouze na členech rozvoje, ve kterých je obsaženo p_1 a zároveň p_2, tedy tvaru

$$p_1 \cdot p_2 \cdot \sum_{i \geq 3} p_i.$$

To je ale přesně typ problému, který jsme uvažovali na začátku, máme totiž fixní součet $p_1 + p_2$. Díky počáteční úvaze víme, že se součin $p_1 \cdot p_2$ sníží, když zvýšíme rozdíl $p_2 - p_1$ a zachováme jejich součet. A tak se sníží i celková hodnota funkce $f(\mathbf{p})$.

Každá přípustná hodnota vektoru $\mathbf{p}$ lze vytvořit z vektoru $(\frac{1}{d}, \frac{1}{d}, \ldots, \frac{1}{d})$ tím, že vezmeme jeho i-tou a j-tou složku, zachováme jejich součet a zvýšíme jejich rozdíl. Po konečně mnoha krocích vytvoříme vektor $\mathbf{p}$. Při každém tomto kroku se sníží hodnota funkce $f(\mathbf{p})$, jak jsme ukázali. Odtud plyne *Schurovská konkávnost* funkce $R_n(\mathbf{p})$. Pro přehlednost jsme důkaz provedli jen pro $n = 3$, ale nic nebrání zobecnit výsledek pro $n \leq d$. □

2.4 Aproximace pomocí rozdílu

Na rozdíl od *základního problému* je výpočet přesné hodnoty $R_n(\mathbf{p})$ pro *obecný problém* výpočetně náročný. A tak pro výpočet $R_n(\mathbf{p})$ pro hodnoty n, které nejsou blízko 0 ani d, je nalezení aproximace nutností. Podrobněji výpočetní náročnost popíšeme v kapitole 3.

V této sekci představíme aproximaci, kterou navrhl Nunnikhoven ve svém článku [7].

Každý ze členů $p_j,\ j = 1, \ldots, d$ můžeme napsat ve tvaru odchylky od *základního problému*

$$p_j = \frac{1}{d} + e_j, \text{ kde } \sum_{j=1}^{d} e_j = 0.$$

Když toto aplikujeme na vzorec (2.1), dostáváme

$$R_n(\mathbf{p}) = n! \sum_{i_1 < i_2 < \ldots < i_n} \left(\frac{1}{d} + e_{i_1}\right) \cdot \left(\frac{1}{d} + e_{i_2}\right) \cdot \ldots \cdot \left(\frac{1}{d} + e_{i_n}\right).$$

Po roznásobení můžeme výrazy rozdělit do $n+1$ skupin. V každé z nich jsou výrazy tvaru

$$n!\left(\frac{1}{d}\right)^{n-s}(e_{j_1}\cdot e_{j_2}\cdot\ldots\cdot e_{j_s}),\ \text{kde } s=0,1,\ldots,n,\{j_1,\ldots,j_s\}\subset\{i_1,\ldots,i_n\}.$$

Pro dané $s=0,1,\ldots,n$ označíme součet všech výrazů ze skupiny jako $T(n,s,\mathbf{p})$, platí tedy

$$R_n=\sum_{s=0}^{n}T(n,s,\mathbf{p}).$$

Spočteme přesnou hodnotu $T(n,s,\mathbf{p})$ pro $s=0,1,2$.

Pro $s=0$ dostáváme

$$T(n,0,\mathbf{p})=n!\sum_{i_1<\ldots<i_n}\left(\frac{1}{d}\right)^n.$$

Sčítáme $\binom{d}{n}$ členů, tedy

$$T(n,0,\mathbf{p})=n!\frac{d!}{n!\,(d-n)!}\left(\frac{1}{d}\right)^n=\frac{d!}{d^n\,(d-n)!}.$$

To odpovídá R_n pro *základní problém.*

Pro $s=1$ máme

$$T(n,1,\mathbf{p})=n!\sum_{i_1<\ldots<i_n}\left(\frac{1}{d}\right)^{n-1}\cdot(e_{i_1}+e_{i_2}+\ldots+e_{i_n}).$$

Součet má $n\binom{d}{n}$ členů. Počet výskytů každé z hodnot $e_j,\ j=1,\ldots,d$ v tomto součtu je

$$\frac{n\frac{d!}{n!\,d-n!}}{d}=\frac{(d-1)!}{(n-1)!\,(d-n)!}=\binom{d-1}{n-1}.$$

Vzorec pro $T(n,1,\mathbf{p})$ tedy můžeme zapsat jako

$$T(n,1,\mathbf{p})=n!\left(\frac{1}{d}\right)^{n-1}\binom{d-1}{n-1}\sum_{j=1}^{d}e_j=0. \tag{2.4}$$

Poslední rovnost platí, neboť $\sum_{j=1}^{d}e_j=0$ a ostatní činitelé jsou konečná čísla.

Pro $s=2$ dostáváme

$$T(n,2,\mathbf{p})=n!\sum_{i_1<\ldots<i_n}\left(\frac{1}{d}\right)^{n-2}\sum_{j_1<j_2}e_{j_1}\cdot e_{j_2},\ \text{kde } \{j_1,j_2\}\subset\{i_1,\ldots,i_n\}.$$

Je $\binom{n}{2}$ možností, jak zvolit j_1,j_2, a tak má součet $\binom{d}{n}\binom{n}{2}$ členů. Celkový počet kombinací j_1 a j_2 takových, že $1\le j_1<j_2\le d$ je $\binom{d}{2}$. Počet výskytů v součtu každé této kombinace j_1 a j_2 je

$$\frac{\binom{d}{n}\binom{n}{2}}{\binom{d}{2}}=\frac{\frac{d!}{n!\,(d-n)!}\frac{n!}{(n-2)!\,2}}{\frac{d!}{(d-2)!\,2}}=\frac{(d-2)!}{(d-n)!\,(n-2)!}=\binom{d-2}{n-2}.$$

Odtud získáváme

$$T(n,2,\mathbf{p}) = n!\left(\frac{1}{d}\right)^{n-2}\binom{d-2}{n-2}\sum_{1\le j_1<j_2\le d} e_{j_1}\cdot e_{j_2}.$$

Existují dvě permutace e_{j_1}, e_{j_2}, a tak můžeme součet upravit:

$$\sum_{1\le j_1<j_2\le d} e_{j_1}\cdot e_{j_2} = \frac{1}{2}\sum_{j_1=1}^{d} e_{j_1}\sum_{\substack{j_2=1\\ j_2\neq j_1}}^{d} e_{j_2} = \frac{1}{2}\sum_{j_1=1}^{d} e_{j_1}\left[\sum_{j_2=1}^{d}(e_{j_2}) - e_{j_1}\right] =$$

$$= \frac{1}{2}\sum_{j_1=1}^{d} e_{j_1}\cdot(0-e_{j_1}) = -\frac{1}{2}\sum_{j_1=1}^{d} e_{j_1}^2.$$

Proto platí

$$T(n,2,\mathbf{p}) = \frac{-n!}{2d^{n-2}}\binom{d-2}{n-2}\sum_{j=1}^{d} e_j^2.$$

Obdobným způsobem dostaneme

$$T(n,3,\mathbf{p}) = \frac{n!}{3d^{n-3}}\binom{d-3}{n-3}\sum_{j=1}^{d} e_j^3$$

a

$$T(n,4,\mathbf{p}) = \frac{n!}{4!d^{n-4}}\binom{d-4}{n-4}\left[-6\sum_{j=1}^{d} e_j^4 + 3\left(\sum_{j=1}^{d} e_j^2\right)^2\right].$$

Uvažujme nyní obecné $1 \le s \le n$. Platí

$$T(n,s,\mathbf{p}) = n!\sum_{i_1<\ldots<i_n}\left(\frac{1}{d}\right)^{n-s}\sum_{j_1<\ldots<j_s} e_{j_1}\cdot e_{j_2}\cdot\ldots\cdot e_{j_s},$$

kde $\{j_1,\ldots,j_s\} \subset \{i_1,\ldots,i_n\}$. Tento součet má $\binom{d}{n}\binom{n}{s}$ členů. Celkový počet kombinací $j_1,\ldots,j_s$ takových, že $1 \le j_1 < \ldots < j_s \le d$ je $\binom{d}{s}$. Počet výskytů v součtu každé této kombinace $j_1,\ldots,j_s$ je

$$\frac{\binom{d}{n}\binom{n}{s}}{\binom{d}{s}} = \frac{\frac{d!}{n!\,(d-n)!}\frac{n!}{(n-s)!\,s!}}{\frac{d!}{(d-s)!\,s!}} = \frac{(d-s)!}{(d-n)!\,(n-s)!} = \binom{d-s}{n-s}.$$

Výraz $T(n,s,\mathbf{p})$ tak můžeme upravit na:

$$T(n,s,\mathbf{p}) = n!\left(\frac{1}{d}\right)^{n-s}\binom{d-s}{n-s}\sum_{1\le j_1<\ldots<j_s\le d} e_{j_1}\cdot\ldots\cdot e_{j_s}. \tag{2.5}$$

Výrazně jednodušší předpis pro $T(n,s,\mathbf{p})$ zatím není znám.

Nunnikhoven dále ve svém článku [7] ukazuje, že $T(n,s,\mathbf{p})$ vyšších řádů (tedy pro vysoká s) jsou blízko nuly. V důkazu se dopouští chyby, obecně toto tvrzení není pravda. Podrobně situaci rozebereme v následující sekci. Nyní ale uveďme jiný výsledek ze zmíněného článku, a to rekurentní vztah pro $T(n,s,\mathbf{p})$. Ten

nám umožní přesný výpočet $T(n, s, \mathbf{p})$ vyšších řádů, který mimo jiné budeme potřebovat pro zdůvodnění chyby ve zmíněném důkazu.

Pro $f \in \mathbb{N},\ 1 \leq f \leq d$ označme

$$O_{s,f}(\mathbf{p}) = \sum_{1 \leq j_1 < \ldots < j_s \leq f} e_{j_1} \cdot \ldots \cdot e_{j_s}. \tag{2.6}$$

Podle (2.5) platí

$$T(n, s, \mathbf{p}) = n! \left(\frac{1}{d}\right)^{n-s} \binom{d-s}{n-s} O_{s,d}(\mathbf{p}). \tag{2.7}$$

Rozdělíme $O_{s,f}$ na dva součty. Pro j_s jsou dvě možnosti. Buď platí $j_s < f$, v takovém případě sčítáme přes všechna $1 \leq j_1 < \ldots < j_s \leq f - 1$, tedy

$$\sum_{1 \leq j_1 < \ldots < j_s \leq f-1} e_{j_1} \cdot \ldots \cdot e_{j_{s-1}} = O_{s,f-1}(\mathbf{p}).$$

Anebo platí $j_s = f$, pak díky $j_{s-1} < j_s$ platí $j_{s-1} \leq f - 1$. Můžeme e_f vytknout a sčítat přes všechna $1 \leq j_1 < \ldots < j_{s-1} \leq f - 1$, tedy

$$e_f \cdot \sum_{1 \leq j_1 < \ldots < j_{s-1} \leq f-1} e_{j_1} \cdot \ldots \cdot e_{j_{s-1}} = e_f \cdot O_{s-1,f-1}(\mathbf{p}).$$

Dostali jsme

$$O_{s,f}(\mathbf{p}) = O_{s,f-1}(\mathbf{p}) + e_f \cdot O_{s-1,f-1}(\mathbf{p}), \tag{2.8}$$

přičemž platí počáteční podmínky

$$O_{1,k}(\mathbf{p}) = \sum_{j=1}^{k} e_j \quad \text{a} \quad O_{j,j}(\mathbf{p}) = e_1 \cdot \ldots \cdot e_j, \text{ kde } j, k \in \mathbb{N},\ j, k \in \{1, \ldots, d\}.$$

První počáteční podmínku napovídá (2.4), pro $k < d$ sčítáme jen přes prvních k pozic. Pro $s = d$ obsahuje (2.6) jediný člen, který napovídá druhou počáteční podmínku, pro $j < d$ se omezíme jen na prvních j pozic.

Kombinace vztahů (2.7) a (2.8) nám umožnuje přesně spočítat $T(n, s, \mathbf{p})$. Protože platí

$$R_n(\mathbf{p}) = \sum_{s=0}^{n} T(n, s, \mathbf{p}),$$

tak nám tato kombinace vztahů umožňuje i přesně spočítat hodnotu $R_n(\mathbf{p})$.

2.5 Konvergence $T(n, s, \mathbf{p})$

V této sekci rozebereme chybný důkaz Nunnikhovenova tvrzení, že $T(n, s, \mathbf{p})$ jsou blízko nuly pro vysoká s. Kdyby toto tvrzení bylo pravdivé, umožňovalo by aproximaci $R_n(\mathbf{p})$ pomocí součtu několika prvních $T(n, s, \mathbf{p})$. Dále pomocí programu spočteme hodnoty $T(n, s, \mathbf{p})$ pro různé hodnoty vektoru pravděpodobností $\mathbf{p}$.

Uvedeme důkaz, jak ho předkládá Nunnikhoven, až do chvíle než dochází k chybě. Vyjdeme ze vzorce 2.5. Jeho podobu

$$T(n, s, \mathbf{p}) = n! \left(\frac{1}{d}\right)^{n-s} \binom{d-s}{n-s} \sum_{1 \leq j_1 < \ldots < j_s \leq d} e_{j_1} \cdot \ldots \cdot e_{j_s}$$

upravíme pomocí alternativního zápisu

$$\sum_{1\le j_1<\ldots<j_s\le d} e_{j_1}\cdot\ldots\cdot e_{j_s}.$$

Je celkem $s!$ permutací $e_{j_1},\ldots,e_{j_s}$, a tak můžeme součet upravit na

$$\begin{aligned}\sum_{1\le j_1<\ldots<j_s\le d} e_{j_1}\cdot\ldots\cdot e_{j_s} &= \frac{1}{s!}\sum_{j_1=1}^{d}\sum_{\substack{j_2=1\\ j_2\neq j_1}}^{d}\cdots\sum_{\substack{j_s=1\\ j_s\neq j_1,\ldots,j_{s-1}}}^{d} e_{j_1}\cdot\ldots\cdot e_{j_s}\\ &= \frac{1}{s!}\sum_{j_1=1}^{d} e_{j_1}\sum_{\substack{j_2=1\\ j_2\neq j_1}}^{d} e_{j_2}\cdots\sum_{\substack{j_{s-1}=1\\ j_{s-1}\neq j_1,\ldots,j_{s-2}}}^{d} e_{j_{s-1}}\sum_{\substack{j_s=1\\ j_s\neq j_1,\ldots,j_{s-1}}}^{d} e_{j_s}.\end{aligned}$$

Pro $T(n,s,\mathbf{p})$ dostáváme:

$$T(n,s,\mathbf{p})=n!\left(\frac{1}{d}\right)^{n-s}\binom{d-s}{n-s}\frac{1}{s!}\sum_{j_1=1}^{d} e_{j_1}\sum_{\substack{j_2=1\\ j_2\neq j_1}}^{d} e_{j_2}\cdots\sum_{\substack{j_{s-1}=1\\ j_{s-1}\neq j_1,\ldots,j_{s-2}}}^{d} e_{j_{s-1}}\sum_{\substack{j_s=1\\ j_s\neq j_1,\ldots,j_{s-1}}}^{d} e_{j_s}.$$

Podívejme se, jak vypadá předcházející výraz, totiž $T(n,s-1,\mathbf{p})$:

$$\begin{aligned}T(n,s-1,\mathbf{p}) = n!\left(\frac{1}{d}\right)^{n-s+1}\binom{d-s+1}{n-s+1}\frac{1}{(s-1)!}\sum_{j_1=1}^{d} e_{j_1}\sum_{\substack{j_2=1\\ j_2\neq j_1}}^{d} e_{j_2}\cdots\\ \cdots\sum_{\substack{j_{s-1}=1\\ j_{s-1}\neq j_1,\ldots,j_{s-2}}}^{d} e_{j_{s-1}}.\end{aligned}$$

Platí

$$\begin{aligned}\binom{d-s}{n-s} &= \frac{(d-s)!}{(d-n)!\,(n-s)!} = \frac{(d-s+1)!}{(d-n)!\,(n-s+1)!}\cdot\frac{n-s+1}{d-s+1} =\\ &= \binom{d-s+1}{n-s+1}\frac{n-s+1}{d-s+1}.\end{aligned}$$

Proto můžeme $T(n,s,\mathbf{p})$ získat tak, že každý člen v $T(n,s-1,\mathbf{p})$ vynásobíme

$$\frac{d}{s}\frac{n-s+1}{d-s+1}\sum_{\substack{j_s=1\\ j_s\neq j_1,\ldots,j_{s-1}}}^{d} e_{j_s}.$$

Nyní nastává chvíle, kdy Nunnikhoven ve svém článku chybně argumentuje. Tvrdí, že $\frac{d}{d-s+1}$ bude blízko jedné a $\frac{n-s+1}{s}$ bude malé, převážně mezi jedničkou a desítkou. To je pravda. Dodává ovšem, že

$$\sum_{\substack{j_s=1\\ j_s\neq j_1,\ldots,j_{s-1}}}^{d} e_{j_s}$$

bude blízko nuly a to již obecně pravda není.

Nunnikhoven argumentuje krátce, jen uvádí pravdivou rovnost $\sum_{j_s=1}^{d} e_{j_s} = 0$. Z toho ovšem neplyne, že

$$\sum_{\substack{j_s=1 \\ j_s \neq j_1,\ldots,j_{s-1}}}^{d} e_{j_s}$$

je blízko nuly. Sice platí

$$\left| \sum_{\substack{j_s=1 \\ j_s \neq j_1,\ldots,j_{s-1}}}^{d} e_{j_s} \right| \leq 1,$$

ale vůbec nic nenaznačuje, že by se tento člen měl s rostoucím s zmenšovat nebo blížit nule.

Uvedeme příklad, kdy tomu tak není. Pro $d = 365$ a případ, že se žáci s pravděpodobností 1 narodí 1. ledna, je $e_1 = \frac{365-1}{365}$, $e_2 = e_3 = \ldots = e_{365} = \frac{-1}{365}$. Pak pro každé s existují $j_1, \ldots, j_{s-1}$ takové, že

$$\left| \sum_{\substack{j_s=1 \\ j_s \neq j_1,\ldots,j_{s-1}}}^{d} e_{j_s} \right| \geq \frac{1}{2}.$$

Stačí volit

$$\begin{aligned} &j_1 = 1 \text{ a } j_2, \ldots, j_{s-1} \text{ libovolné} \quad \text{pro } s < 183 \\ &j_1, \ldots, j_{s-1} > 1 \quad \text{pro } s \geq 183. \end{aligned}$$

I pro méně extrémní případy, kdy je pravděpodobnost narození nenulová pro mnoho dnů, nemusí nutně

$$\left| \sum_{\substack{j_s=1 \\ j_s \neq j_1,\ldots,j_{s-1}}}^{d} e_{j_s} \right|$$

být zanedbatelně malé. Například v případě, kdy se pravděpodobnosti narození $\mathbf{p}$ značně odlišují od stejných pravděpodobností a $e_{j_1}, \ldots, e_{j_{s-1}}$ mají stejné znaménko.

Následující úvahy ukazují, že obecně není pravda, že $T(n, s, \mathbf{p})$ konvergují k nule pro vysoká s. To ověříme i sestavením programu a numerickým výpočtem. Ještě než tyto výpočty uvedeme, vraťme se k případu, kdy se s pravděpodobností jedna žáci rodí 1. ledna.

K výpočtu $T(n, s, \mathbf{p})$ pro tento případ využijeme vzorec (2.5). Můžeme rozlišit dva případy — buď j_1 je rovno 1 či nikoliv. Pokud platí $j_1 = 1$, pak ze součtu

$$\sum_{1 \leq j_1 < \ldots < j_s \leq d} e_{j_1} \cdot \ldots \cdot e_{j_s}$$

můžeme vytknout e_1 a máme

$$e_1 \cdot \sum_{2 \leq j_2 < \ldots < j_s \leq d} e_{j_2} \cdot \ldots \cdot e_{j_s}.$$

Pokud platí $j_1 > 1$, pak se součet

$$\sum_{1 \le j_1 < \ldots < j_s \le d} e_{j_1} \cdot \ldots \cdot e_{j_s}$$

redukuje na

$$\sum_{2 \le j_1 < \ldots < j_s \le d} e_{j_1} \cdot \ldots \cdot e_{j_s}.$$

Dohromady máme

$$\sum_{1 \le j_1 < \ldots < j_s \le d} e_{j_1} \cdot \ldots \cdot e_{j_s} = e_1 \cdot \sum_{2 \le j_2 < \ldots < j_s \le d} e_{j_2} \cdot \ldots \cdot e_{j_s} + \sum_{2 \le j_1 < \ldots < j_s \le d} e_{j_1} \cdot \ldots \cdot e_{j_s}. \quad (2.9)$$

Platí $e_1 = \frac{d-1}{d}$ a $e_j = \frac{-1}{d}, 2 \le j \le d$. Součet

$$\sum_{2 \le j_2 < \ldots < j_s \le d} e_{j_2} \cdot \ldots \cdot e_{j_s}$$

má $\binom{d-1}{s-1}$ členů. Součet

$$\sum_{2 \le j_1 < \ldots < j_s \le d} e_{j_1} \cdot \ldots \cdot e_{j_s}$$

má $\binom{d-1}{s}$ členů. Proto vzorec (2.9) můžeme upravit na

$$\sum_{1 \le j_1 < \ldots < j_s \le d} e_{j_1} \cdot \ldots \cdot e_{j_s} = \frac{d-1}{d} \binom{d-1}{s-1} \left(\frac{-1}{d}\right)^{s-1} + \binom{d-1}{s} \left(\frac{-1}{d}\right)^{s}.$$

Z obou sčítanců vytkneme $\frac{(-1)^{s-1}}{d^s}$:

$$\sum_{1 \le j_1 < \ldots < j_s \le d} e_{j_1} \cdot \ldots \cdot e_{j_s} = \left[(d-1)\binom{d-1}{s-1} - \binom{d-1}{s}\right] \frac{(-1)^{s-1}}{d^s}.$$

Dosadíme do vzorce (2.5) a explicitně vyjádříme kombinační čísla:

$$\begin{aligned} T(n,s,\mathbf{p}) &= n! \left(\frac{1}{d}\right)^{n-s} \binom{d-s}{n-s} \sum_{1 \le j_1 < \ldots < j_s \le d} e_{j_1} \cdot \ldots \cdot e_{j_s} \\ &= n! \frac{1}{d^{n-s}} \frac{(d-s)!}{(d-n)!(n-s)!} \left[(d-1)\frac{(d-1)!}{(s-1)!\,(d-s)!} - \frac{(d-1)!}{s!\,(d-s-1)!}\right] \cdot \\ &\cdot \frac{(-1)^{s-1}}{d^s} \\ &= n! \frac{1}{d^n} \frac{(d-s)!}{(d-n)!(n-s)!} \left[(d-1)\frac{(d-1)!}{(s-1)!\,(d-s)!} - \frac{(d-1)!\,(d-s)}{s!\,(d-s)!}\right] \cdot \\ &\cdot (-1)^{s-1}. \end{aligned}$$

Pokrátíme $(d-s)!$ a ze závorky vytkneme $(d-1)!$. Dostáváme

$$T(n,s,\mathbf{p}) = \frac{n!}{(n-s)!} \frac{1}{d^n} \frac{(d-1)!}{(d-n)!} \left[\frac{d-1}{(s-1)!} - \frac{d-s}{s!}\right] \cdot (-1)^{s-1}.$$

Ze závorky vytkneme $\frac{1}{s!}$ a upravíme:

$$\begin{aligned} T(n,s,\mathbf{p}) &= \frac{n!}{(n-s)!s!}\frac{(d-1)!}{d^n\,(d-n)!}\left[s\cdot(d-1)-(d-s)\right](-1)^{s-1} \\ &= \frac{n!}{(n-s)!s!}\frac{(d-1)!}{d^n\,(d-n)!}\left[sd-s-d+s\right](-1)^{s-1} \\ &= \frac{n!}{(n-s)!s!}\frac{(d-1)!}{d^n\,(d-n)!}\left[d\cdot(s-1)\right](-1)^{s-1}. \end{aligned}$$

Pokrátíme d a převedeme na kombinační číslo:

$$T(n,s,\mathbf{p}) = \binom{n}{s}\frac{(d-1)!}{d^{n-1}\,(d-n)!}(s-1)(-1)^{s-1}. \tag{2.10}$$

Vidíme, že když je pravděpodobnost 1 narození 1. ledna, tak $T(n,s,\mathbf{p})$ záleží na s pomocí vztahu $(-1)^{s-1}(s-1)\binom{n}{s}$. Není tedy pravda, že by se hodnoty $T(n,s,\mathbf{p})$ rychle snižovaly, naopak dosahují svého maxima kolem $s=\frac{n}{2}$.

Podařilo se nám teoreticky odvodit protipříklad tvrzení, že $T(n,s,\mathbf{p})$ vyšších řádů jsou blízko nuly. Výsledek bude ještě názornější, až představíme data získané pomocí numerického řešení.

Program `Tnsp` počítá hodnoty $T(n,s,\mathbf{p})$ pomocí rekurzivních vztahů (2.7) a (2.8) ze závěru předchozí sekce. Program umí spočítat $T(n,s,\mathbf{p})$ pro libovolný vektor pravděpodobností $\mathbf{p}$ a počet žáků ve třídě nižší než 21. Součtem $T(n,s,\mathbf{p})$ pak program zjistí pravděpodobnost $R_n(\mathbf{p})$. Zdrojový kód programu je k nalezení v *příloze 1*.

Počet žáků, pro který program funguje, by šel zvýšit buď převedením do jiného programovacího jazyka nebo jiným způsobem výpočtu v rámci programu — to by ovšem mohlo vést ke snížení přesnosti výpočtu. Nejedná se o program, kterým budeme simulovat *problém shody narozenin*, ten představíme v kapitole 3. A tak lze považovat počet žáků, pro který je program schopen pracovat, za uspokojivý.

Nejprve program použijeme na zmiňovaný případ, kdy se žáci rodí s pravděpodobností jedna 1. ledna. Výsledky pro $5, 10, 15$ a 20 žáků jsou vidět v tabulce 2.2. Správné hodnoty pro $T(n,1,\mathbf{p})$ a pravděpodobnost $R_n(\mathbf{p})$ jsou rovny 0, odchylka je dána reprezentací dat v počítači. Názorně je vidět, že hodnoty $T(n,s,\mathbf{p})$ mohou být velmi vysoké. Uvedené hodnoty bychom mohli spočítat i pomocí vzorce (2.10).

Až do konce této sekce uvažujme, že je ve třídě 20 žáků. Program dále aplikujeme na dvojici dat, které zmiňuje Nunnikhoven ve svém článku [7]. Prvními daty byly pravděpodobnosti narození v USA. Zde vyšlo $T(20,2,\mathbf{p})$ řádově 10^{-4} a $T(20,s,\mathbf{p})$ vyšších řádů rychle konvergovaly k nule, všechny byly řádově menší nebo rovny 10^{-7}.

Další situací, kterou Nunnikhoven zmiňuje, jsou pravděpodobnosti narození rovnoměrně rozdělené mezi prvních 304 dnů nepřestupného roku. Program `Tnsp` jsme použili pro obdobnou situaci, kdy jsou pravděpodobnosti narození rovnoměrně rozdělené mezi prvních 300 dnů. Výsledek ukazuje tabulka 2.3. Hodnoty $T(20,3,\mathbf{p})$ a $T(20,4,\mathbf{p})$ jsou řádově 10^{-3}, hodnota $T(20,5,\mathbf{p})$ je řádově 10^{-4}. Opět $T(20,s,\mathbf{p})$ vyšších řádů konvergují k nule.

Vidíme, že pro data, které Nunnikhoven použil, jeho tvrzení skutečně platí. Pro data narození z USA navíc platí Nunnikhovenův postřeh, že téměř veškerá

Tabulka 2.2: Hodnoty $T(n, s, \mathbf{p})$ pro pravděpodobnost jedna narození 1. ledna

	$n = 5$	$n = 10$	$n = 15$	$n = 20$
$T(n, 0, \mathbf{p})$	0,9728	0,88305	0,74709	0,58856
$T(n, 1, \mathbf{p})$	-1,528·10^{-17}	-2.775·10^{-17}	-3,521·10^{-17}	-3,6986·10^{-17}
$T(n, 2, \mathbf{p})$	-10	-40	-78	-111
$T(n, 3, \mathbf{p})$	19	211	680	1341
$T(n, 4, \mathbf{p})$	-15	-556	-3059	-8554
$T(n, 5, \mathbf{p})$	4	890	8974	36500
$T(n, 6, \mathbf{p})$		-927	-18696	-114063
$T(n, 7, \mathbf{p})$		636	28845	273752
$T(n, 8, \mathbf{p})$		-278	-33653	-518988
$T(n, 9, \mathbf{p})$		71	29914	790838
$T(n, 10, \mathbf{p})$		-8	-20192	-978663
$T(n, 11, \mathbf{p})$			10198	988548
$T(n, 12, \mathbf{p})$			-3739	-815552
$T(n, 13, \mathbf{p})$			941	547504
$T(n, 14, \mathbf{p})$			-146	-296564
$T(n, 15, \mathbf{p})$			10	127751
$T(n, 16, \mathbf{p})$				-42774
$T(n, 17, \mathbf{p})$				10735
$T(n, 18, \mathbf{p})$				-1901
$T(n, 19, \mathbf{p})$				211
$T(n, 20, \mathbf{p})$				-11
$R_n(\mathbf{p})$	5,052·10^{-15}	-2,498·10^{-13}	-1,313·10^{-11}	-6,183·10^{-10}

odchylka je popsána výrazem $T(20, 2, \mathbf{p})$. Ten je založen na střední kvadratické odchylce od pravděpodobností *základního problému* — je to násobek $\sum_{j=1}^{d} e_j^2$. Tedy veličiny, kterou bychom přirozeně volili k popisu velikosti rozdílu od stejných pravděpodobností.

Podívejme se nyní na hodnoty $T(n, s, \mathbf{p})$ pro další vektory pravděpodobností. Vraťme se znovu do situace, kdy jsou pravděpodobnosti narození rovnoměrně rozděleny mezi několik prvních dnů roku o 365 dnech. Hodnoty $T(20, s, \mathbf{p})$ za situace, kdy jsou pravděpodobnosti narození rovnoměrně rozděleny mezi $300, 100, 50, 30, 15, 10$ a 5 prvních dnů roku, ukazuje tabulka 2.3.

Se snižujícím počtem dnů, mezi které jsou pravděpodobnosti narození rovnoměrně rozděleny, hodnoty $T(20, s, \mathbf{p})$ rychle vzrůstají. Pro nízký počet dnů je vidět jistý druh symetričnosti, který jsme viděli už v situaci, kdy pravděpodobnost narození 1. ledna byla jedna. Zároveň se občas vyskytnou hodnoty, které jsou v absolutní hodnotě výrazně nižší než hodnoty předchozí i následující. Příkladem je hodnota $T(20, 10, \mathbf{p})$ pro pravděpodobnosti narození rovnoměrně rozdělené mezi prvních 30 dnů.

Ve skutečných datech jsou rozdíly v počtech narození jen malé, a tak na ně lze použít Nunnikhovenem navrhovaná aproximace pomocí součtu několika prvních členů $T(n, s, \mathbf{p})$. Otázkou ovšem zůstává, jak lze jednoduše rozpoznat, kdy je tato aproximace možná. Uveďme, s jakými obtížemi se musíme potýkat, pokud bychom chtěli tyto předpoklady formulovat.

Tabulka 2.3: Hodnoty $T(n, s, \mathbf{p})$ pro $d = 365$ a pravděpodobnosti narození rovnoměrně rozdělené mezi několik prvních dnů roku

Počet dnů	300	100	50	30
$T(20, 0, \mathbf{p})$	0,5886	0,5886	0,5886	0,5886
$T(20, 1, \mathbf{p})$	$6{,}8 \cdot 10^{-18}$	$-5{,}1 \cdot 10^{-18}$	$-2{,}2 \cdot 10^{-17}$	$-2{,}9 \cdot 10^{-17}$
$T(20, 2, \mathbf{p})$	-0,07	-0,8	-1,9	-3,4
$T(20, 3, \mathbf{p})$	$-1{,}7 \cdot 10^{-3}$	0,04	0,3	1,2
$T(20, 4, \mathbf{p})$	$3{,}0 \cdot 10^{-3}$	0,5	2,5	7,7
$T(20, 5, \mathbf{p})$	$1{,}4 \cdot 10^{-4}$	-0,04	-0,8	-4,7
$T(20, 6, \mathbf{p})$	$-6{,}8 \cdot 10^{-5}$	-0,1	-1,6	-7,9
$T(20, 7, \mathbf{p})$	$-4{,}3 \cdot 10^{-6}$	0,02	0,7	7,1
$T(20, 8, \mathbf{p})$	$8{,}3 \cdot 10^{-7}$	0,02	0,5	3,3
$T(20, 9, \mathbf{p})$	$6{,}5 \cdot 10^{-8}$	$-3{,}2 \cdot 10^{-3}$	-0,3	-4,8
$T(20, 10, \mathbf{p})$	$-5{,}3 \cdot 10^{-9}$	$-1{,}9 \cdot 10^{-3}$	-0,08	$3{,}8 \cdot 10^{-4}$
$T(20, 11, \mathbf{p})$	$-5{,}0 \cdot 10^{-10}$	$3{,}2 \cdot 10^{-4}$	0,06	1,5
$T(20, 12, \mathbf{p})$	$1{,}5 \cdot 10^{-11}$	$9{,}6 \cdot 10^{-5}$	$2{,}7 \cdot 10^{-3}$	-0,4
$T(20, 13, \mathbf{p})$	$2{,}0 \cdot 10^{-12}$	$-1{,}7 \cdot 10^{-5}$	$-6{,}6 \cdot 10^{-3}$	-0,2
$T(20, 14, \mathbf{p})$	$-8{,}9 \cdot 10^{-15}$	$-2{,}4 \cdot 10^{-6}$	$4{,}5 \cdot 10^{-4}$	0,08
$T(20, 15, \mathbf{p})$	$-3{,}5 \cdot 10^{-15}$	$4{,}2 \cdot 10^{-7}$	$2{,}9 \cdot 10^{-4}$	$3{,}7 \cdot 10^{-3}$
$T(20, 16, \mathbf{p})$	$-2{,}6 \cdot 10^{-17}$	$2{,}5 \cdot 10^{-8}$	$-3{,}6 \cdot 10^{-5}$	$-5{,}4 \cdot 10^{-3}$
$T(20, 17, \mathbf{p})$	$2{,}3 \cdot 10^{-18}$	$-4{,}1 \cdot 10^{-9}$	$-4{,}2 \cdot 10^{-6}$	$4{,}5 \cdot 10^{-4}$
$T(20, 18, \mathbf{p})$	$2{,}6 \cdot 10^{-20}$	$-7{,}3 \cdot 10^{-11}$	$6{,}2 \cdot 10^{-7}$	$7{,}8 \cdot 10^{-5}$
$T(20, 19, \mathbf{p})$	$-3{,}5 \cdot 10^{-22}$	$1{,}1 \cdot 10^{-11}$	$6{,}1 \cdot 10^{-9}$	$-8{,}9 \cdot 10^{-6}$
$T(20, 20, \mathbf{p})$	$-2{,}4 \cdot 10^{-24}$	$-4{,}6 \cdot 10^{-14}$	$-1{,}2 \cdot 10^{-9}$	$-2{,}8 \cdot 10^{-9}$

Počet dnů	15	10	5
$T(20, 0, \mathbf{p})$	0,5886	0,5886	0,5886
$T(20, 1, \mathbf{p})$	$-3{,}1 \cdot 10^{-17}$	$-3{,}4 \cdot 10^{-17}$	$-3{,}5 \cdot 10^{-17}$
$T(20, 2, \mathbf{p})$	-7,2	-10,9	-22,1
$T(20, 3, \mathbf{p})$	5,3	12,4	51,9
$T(20, 4, \mathbf{p})$	31,2	66,7	206,7
$T(20, 5, \mathbf{p})$	-43,3	-148,1	-1083,7
$T(20, 6, \mathbf{p})$	-47,3	-85,7	1017,8
$T(20, 7, \mathbf{p})$	119,7	538,2	4104,9
$T(20, 8, \mathbf{p})$	-14,9	-396,8	-13917,4
$T(20, 9, \mathbf{p})$	-125,7	-518,7	15174,3
$T(20, 10, \mathbf{p})$	90,9	1020	7194,1
$T(20, 11, \mathbf{p})$	30,6	-382,4	-43856,2
$T(20, 12, \mathbf{p})$	-60,9	-459,7	61593,0
$T(20, 13, \mathbf{p})$	17,2	544,7	-41941,1
$T(20, 14, \mathbf{p})$	9,0	-161,6	4593,9
$T(20, 15, \mathbf{p})$	-6,4	-65,3	18885,5
$T(20, 16, \mathbf{p})$	0,8	60,9	-19730,0
$T(20, 17, \mathbf{p})$	0,3	-14,9	10527,2
$T(20, 18, \mathbf{p})$	-0,09	0,003	-3364,0
$T(20, 19, \mathbf{p})$	$4{,}6 \cdot 10^{-3}$	0,5	614,4
$T(20, 20, \mathbf{p})$	$3{,}4 \cdot 10^{-4}$	-0,05	-49,8

Připomeňme, že $T(n, s, \mathbf{p})$ má tvar

$$T(n, s, \mathbf{p}) = n! \sum_{i_1 < \ldots < i_n} \left(\frac{1}{d}\right)^{n-s} \sum_{j_1 < \ldots < j_s} e_{j_1} \cdot e_{j_2} \cdot \ldots \cdot e_{j_s},$$

kde $\{j_1, \ldots, j_s\} \subset \{i_1, \ldots, i_n\}$ a že tento součet má $\binom{d}{n}\binom{n}{s}$ členů. Jejich počet tedy nabývá maxima kolem $s = \frac{n}{2}$.

Aby se vzrůstajícím s výrazy $T(n, s, \mathbf{p})$ konvergovaly, musí být kompenzován růst počtu jejich členů buď tím, že se zmenšuje velikost členů, anebo se členy musejí navzájem odečítat.

Platí

$$\sum_{j=1}^{d} e_j = 0.$$

Proto pokud nejsou všechna e_j rovny nule, tak existují $k, l \in \{1, \ldots, d\}$, že e_k je kladné a e_l je záporné. Ve výrazu $T(n, s, \mathbf{p})$ sčítáme přes všechna $i_1 < \ldots < i_n$, z čehož plyne, že se i ve výpočtu $T(n, s, \mathbf{p})$ objeví členy kladné i záporné hodnoty. Ke vzájemnému odečítání členů tedy skutečně dochází.

Proto se zdá být obtížné optimálně formulovat předpoklady, za kterých výrazy $T(n, s, \mathbf{p})$ konvergují. Pokud bychom formulovali předpoklady tak, aby konvergovala posloupnost součtu absolutních hodnot členů výrazu $T(n, s, \mathbf{p})$, pak by byly předpoklady zbytečně přísné. Nevyužili bychom totiž toho, že se jednotlivé členy navzájem odečítají. Vliv tohoto odečítaní může být velký. Nezdá se však, že by tento vliv bylo obecně jednoduché popsat a využít toho při formulování předpokladů.

Největší význam toho, že jsme ukázali, že obecně $T(n, s, \mathbf{p})$ nemusí být blízko nuly, pravděpodobně spočívá v aplikacích *problému shody narozenin*. V těch se totiž pravděpodobnosti mohou značně odlišovat od stejných. Poté by použití zmíněné aproximace mohlo mít fatální důsledky.

2.6 Výpočet pravděpodobnosti pomocí Bellových polynomů

V aplikacích *problému shody narozenin* se někdy setkáváme s případem, kdy d není známé. S tím se setkáme například v *problému shody příjmení*, který popíšeme v sekci 4.1. Někdy ale počet dnů d nemusí být shora omezený. Tak je tomu v následující analogii *problému shody narozenin*.

Máme n mincí (ty jsou ekvivalenty žáků). S každou mincí házíme do té doby, než padne panna. Minci přiřadíme číslo rovné počtu padlých rubů do okamžiku, než padla panna. Pokud je mince symetrická, jsou pravděpodobnosti, že minci přiřadíme $0, 1, 2, 3, \ldots$ po řadě $\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \ldots$ Zajímá nás pravděpodobnost, že mezi n mincemi neexistuje dvojice mincí, které jsme přiřadili stejné číslo (tato pravděpodobnost je ekvivalent R_n). Číslo, které přiřadíme minci, není shora omezené — pro libovolné nezáporné číslo je pravděpodobnost, že toto číslo přiřadíme, kladná.

V této analogii již není možné pro přesný výpočet použít vzorec (2.1), protože by se jednalo o nekonečný součet. Přesto lze pravděpodobnost popsaná v této

analogii spočítat zcela přesně. K tomu v této podkapitole vytvoříme dostatečný aparát. Budeme hledat pravděpodobnost $R_n(\mathbf{p})$ za situace $\sum_{i=1}^{\infty} p_i = 1,\ p_i \geq 0$. Budeme postupovat jako Mase ve svém článku [8].

Označme

$$P_m = \sum_{i=1}^{\infty} p_i^m.$$

Dodefinujme $\forall \mathbf{p}:\ R_0(\mathbf{p}) = 1$. Toto dodefinování má smysl, neboť v prázdné třídě s pravděpodobností 1 nebude dvojice žáků se stejnými narozeninami. V případě, že existuje i takové, že $p_i = 1$, tak platí $R_n(\mathbf{p}) = 0$ pro $n > 1$. V dalších úvahách se omezíme na případ, kdy $\forall i : p_i < 1$ a $n > 1$.

Exponenciální vytvořující funkce posloupnosti $\{R_n(\mathbf{p})\}_{n=0}^{\infty}$ je

$$G(\{R_n(\mathbf{p})\}_{n=0}^{\infty}; t) = \sum_{n=0}^{\infty} R_n(\mathbf{p})\frac{t^n}{n!} = \frac{t^0}{0!}R_0(\mathbf{p}) + \sum_{n=1}^{\infty} R_n(\mathbf{p})\frac{t^n}{n!} = 1 + \sum_{n=1}^{\infty} R_n(\mathbf{p})\frac{t^n}{n!}. \tag{2.11}$$

Naším cílem nyní bude upravit *exponenciální vytvořující funkci* $G(\{R_n(\mathbf{p})\}_{n=0}^{\infty}; t)$ do tvaru součtu *úplných Bellových polynomů*. Nejprve do vztahu 2.11 dosadíme vyjádření $R_n(\mathbf{p})$ ze vztahu (2.1).

$$\begin{aligned} G(\{R_n(\mathbf{p})\}_{n=0}^{\infty}; t) &= 1 + \sum_{n=1}^{\infty} n! \sum_{i_1<\ldots<i_n} p_{i_1} \cdot \ldots \cdot p_{i_n} \frac{t^n}{n!} \\ &= 1 + \sum_{n=1}^{\infty} \sum_{i_1<\ldots<i_n} p_{i_1} \cdot \ldots \cdot p_{i_n} t^n \end{aligned}$$

Ukážeme, že platí

$$1 + \sum_{n=1}^{\infty} \sum_{i_1<\ldots<i_n} p_{i_1} \cdot \ldots \cdot p_{i_n} t^n = \prod_{n=1}^{\infty} (1 + p_n t).$$

Když roznásobíme nekonečný součin vpravo, dostaneme číslo 1 a zbylé výrazy můžeme uspořádat do skupin podle mocniny $t^n, n \in \mathbb{N}$. V každé této skupině jsou výrazy tvaru $p_{i_1} \cdot \ldots \cdot p_{i_n} t^n$ pro $i_1, \ldots, i_n, n \in \mathbb{N}$, přičemž $i_1, \ldots, i_n$ jsou navzájem různé. Navíc se každá kombinace čísel $i_1, \ldots, i_n$ bez ohledu na pořadí vyskytuje právě jednou. Tuto kombinaci čísel $i_1, \ldots, i_n$ můžeme uspořádat podle velikosti a dostat tak vnitřní součet výrazu vlevo. Ekvivalence je dokázana a platí

$$G(\{R_n(\mathbf{p})\}_{n=0}^{\infty}; t) = 1 + \sum_{n=1}^{\infty} \sum_{i_1<\ldots<i_n} p_{i_1} \cdot \ldots \cdot p_{i_n} t^n = \prod_{n=1}^{\infty} (1 + p_n t).$$

Budeme dále upravovat $G\left(\{R_n(\mathbf{p})\}_{n=0}^{\infty}; t\right)$ — převedeme nekonečný součin

$$\prod_{n=1}^{\infty} (1 + p_n t)$$

na exponenciální tvar a uplatníme Taylorův rozvoj:

$$\begin{aligned} G(\{R_n(\mathbf{p})\}_{n=0}^{\infty}; t) &= \exp\left[\sum_{n=1}^{\infty} \log(1 + p_n t)\right],\ t \in (-1, 1] \\ &= \exp\left[\sum_{n=1}^{\infty} \left(\sum_{i=1}^{\infty} (-1)^{i-1} \frac{p_n^i\, t^i}{i}\right)\right],\ t \in (-1, 1]. \end{aligned}$$

Ukážeme, že je řada absolutně konvergentní, abychom mohli zaměnit pořadí sumace:

$$\sum_{n=1}^{\infty}\left(\sum_{i=1}^{\infty}\left|(-1)^{i-1}\frac{p_n^i}{i}t^i\right|\right) \leq \sum_{n=1}^{\infty}\left(\sum_{i=1}^{\infty}\left|\frac{p_n^i}{i}\right|\right) \leq \sum_{n=1}^{\infty}\left(\sum_{i=1}^{\infty}p_n^i\right) = \sum_{n=1}^{\infty}\frac{p_n}{1-p_n}.$$

Označme $p_{n_0} = \max\limits_{n\in\mathbb{N}} p_n$, pak

$$\sum_{n=1}^{\infty}\frac{p_n}{1-p_n} \leq \frac{1}{1-p_{n_0}}\sum_{n=1}^{\infty}p_n = \frac{1}{1-p_{n_0}} \in \mathbb{R}.$$

Po záměně sumace máme:

$$\begin{aligned}
G(\{R_n(\mathbf{p})\}_{n=0}^{\infty};t) &= \exp\left[\sum_{i=1}^{\infty}(-1)^{i-1}\frac{t^i}{i}\sum_{n=1}^{\infty}p_n^i\right],\ t\in(-1,1] \\
&= \exp\left[\sum_{i=1}^{\infty}(-1)^{i-1}\frac{(i-1)!\cdot t^i}{i!}P_i\right],\ t\in(-1,1] \\
&= 1+\sum_{i=1}^{\infty}Y_i[\ldots,(-1)^{j-1}(j-1)!P_j,\ldots]\frac{t^i}{i!},\ t\in(-1,1]. \qquad (2.12)
\end{aligned}$$

Poslední rovnost plyne z definice *úplného Bellova polynomu*.

Povedlo se nám upravit *exponenciální vytvořující funkci* $G(\{R_n(\mathbf{p})\}_{n=0}^{\infty};t)$ do tvaru součtu *úplných Bellových polynomů*. Porovnáním vztahů (2.11) a (2.12) dostáváme

$$R_n(\mathbf{p}) = Y_n[1,\ldots,(-1)^{j-1}(j-1)!P_j,\ldots,(-1)^{n-1}(n-1)!P_n], \qquad (2.13)$$

kde první složku jsme získali dosazením $j=1$. Tento vztah upravíme dvojím způsobem.

Zaprvé, *úplný Bellův polynom* vyjádříme jako součet *částečných Bellových polynomů*, ty vyjádříme explicitně podle vzorce (1.1). Tedy

$$\begin{aligned}
R_n(\mathbf{p}) &= \sum_{k=1}^{n}B_{n,k}[1,\ldots,(-1)^{i-1}(i-1)!P_i,\ldots,(-1)^{n-k}(n-k)!P_{n-k+1}] \\
&= \sum_{k=1}^{n-1}\left\{B_{n,k}[1,\ldots,(-1)^{i-1}(i-1)!P_i,\ldots,(-1)^{n-k}(n-k)!P_{n-k+1}]\right\}+ \\
&+ B_{n,n}(x_1) \\
&= \sum_{k=1}^{n-1}\left\{\sum_{\substack{a_1+\ldots+a_{n-k+1}=k\\ 1a_1+2a_2+\ldots+(n-k+1)a_{n-k+1}=n}}\frac{n!}{a_1!\ldots a_{n-k+1}!}\prod_{i=1}^{n-k+1}\left[\frac{(-1)^{i-1}(i-1)!P_i}{i!}\right]^{a_i}\right\}+ \\
&+ B_{n,n}(x_1) \\
&= \sum_{k=1}^{n-1}\left\{\sum_{\substack{a_1+\ldots+a_{n-k+1}=k\\ 1a_1+2a_2+\ldots+(n-k+1)a_{n-k+1}=n}}\frac{n!}{a_1!\ldots a_{n-k+1}!}\prod_{i=1}^{n-k+1}\left[\frac{(-1)^{i-1}P_i}{i}\right]^{a_i}\right\}+ \\
&+ B_{n,n}(x_1). \qquad (2.14)
\end{aligned}$$

Vyjádříme explicitně $B_{n,n}(x_1)$ a upravíme podmínky řady do jednoduššího tvaru. Pro $B_{n,n}(x_1)$ platí:

$$B_{n,n}(x_1) = \sum_{\substack{a_1=n \\ 1a_1=n}} \frac{n!}{a_1!} \prod_{i=1}^{1} \left[\frac{(-1)^{i-1}P_i}{i}\right]^{a_i} = \frac{n!}{n!}\left[\frac{(-1)^{1-1}P_1}{1}\right]^{n} = 1.$$

Uvědomme si, že podmínka

$$a_1 + \ldots + (n-k+1)a_{n-k+1} = n$$

implikuje, že

$$a_1 + \ldots + a_{n-k+1} \in [1, n].$$

Sčítáme přes všechna

$$a_1 + \ldots + (n-k+1)a_{n-k+1} = n \ \wedge \ a_1 + \ldots + a_{n-k+1} = k, \ k \in \mathbb{N}, \ k \in [1, n-1],$$

a proto se podmínky redukují na

$$a_1 + \ldots + (n-k+1)a_{n-k+1} = n \ \wedge \ a_1 + \ldots + a_{n-k+1} \neq n, \ k \in \mathbb{N}, \ k \in [1, n-1],$$

tedy

$$a_1 + \ldots + (n-k+1)a_{n-k+1} = n \ \wedge \ a_1 \neq n, \ k \in \mathbb{N}, \ k \in [1, n-1].$$

Protože pro $a_1, \ldots, a_n$ nemáme žádné omezení (mohou být rovny 0), můžeme podmínky upravit na

$$a_1 + \ldots + na_n = n \ \wedge \ a_1 \neq n.$$

A tak můžeme (2.14) vyjádřit jako

$$R_n(\mathbf{p}) = 1 + \sum_{\substack{1a_1+2a_2+\ldots+na_n=n \\ a_1\neq n}} \frac{n!}{a_1!\ldots a_n!} \prod_{i=1}^{n} \left(\frac{(-1)^{i-1}P_i}{i}\right)^{a_i}, \tag{2.15}$$

což nám umožňuje výpočet $R_n(\mathbf{p})$ i pro obecně nekonečné d.

Nyní upravíme vztah (2.13) druhým způsobem — aplikujeme na něj rekurentní vztah pro *úplné Bellovy polynomy* (1.2). Dostáváme

$$R_n(\mathbf{p}) = \sum_{i=1}^{n} \binom{n-1}{i-1} (-1)^{i-1}(i-1)!\, P_i\, Y_{n-i}(x_1, \ldots, x_{n-i}).$$

Protože dle (2.13) platí $Y_{n-i}(x_1, \ldots, x_{n-i}) = R_{n-i}(\mathbf{p})$, tak máme

$$R_n(\mathbf{p}) = \sum_{i=1}^{n} (-1)^{i-1} \frac{(n-1)!}{(n-i)!} P_i\, R_{n-i}(\mathbf{p}). \tag{2.16}$$

Tím jsme získali velmi užitečný rekurentní vztah pro $R_n(\mathbf{p})$.

Mase dále ve svém článku [8] uvádí, jak lze pomocí *Bellových polynomů* aproximovat pravděpodobnost $R_n(\mathbf{p})$. V době, kdy Mase článek psal, byly výpočetní prostředky na mnohem nižší úrovni, a tak se při numerickém výpočtu musel spokojit s aproximacemi. V dnešní době nám však výpočetní prostředky dovolují spočítat pravděpodobnost $R_n(\mathbf{p})$ (téměř) přesně podle vzorce (2.16). Více situaci rozebereme v kapitole 3. Z tohoto důvodu a díky tomu, že by bylo nutné k popisu aproximací zavést mnoho speciální notace, tyto aproximace neuvedeme.

3. Programové vybavení

V zadání bakalářské práce bylo vytvořit program pro simulaci *problému shody narozenin*. V této kapitole ho popíšeme. Program bude mimo jiné testovat hypotézu, že všechny pravděpodobnosti narození $p_1, \ldots, p_d$ jsou stejné, tedy rovny $\frac{1}{d}$. Uveďme nejprve, jak lze tuto hypotézu testovat.

3.1 Testování hypotézy o stejných pravděpodobnostech

Nechť máme k dispozici data narození $m \in \mathbb{N}$ lidí. Nechť X_i, $i \in \{1, \ldots, d\}$ značí počet lidí narozených ve dnu i. Sdružené rozdělení těchto veličin se nazývá *multinomické* a platí

$$P(X_1 = x_1, \ldots, X_d = x_d) = \frac{m!}{x_1! \cdots x_d!} p_1^{x_1} \cdots p_d^{x_d},$$
$$x_i = 0, 1, \ldots, m \quad (i = 1, \ldots, d), \qquad x_1 + \cdots + x_d = m,$$

kde p_i jako obvykle značí pravděpodobnosti narození v den i, $i \in \{1, \ldots, d\}$. Značíme $(X_1, \ldots, X_d) \sim M(m; p_1, \ldots, p_d)$.

Chceme testovat hypotézu, že skutečné pravděpodobnosti narození $\mathbf{p}$ se rovnají nějakému vektoru pravděpodobností $\mathbf{q} = (q_1, \ldots, q_d)$. Nejčastěji budeme uvažovat vektor stejných pravděpodobností, tedy $q_i = \frac{1}{d}, i \in \{1, \ldots, d\}$. Zajímá nás, zda platí $p_i = q_i$ pro všechna $i \in \{1, \ldots, d\}$. Označme H_0 hypotézu, že $p_1 = q_1, \ldots, p_d = q_d$ a H_1 alternativní hypotézu, že existuje $i \in \{1, \ldots, d\}$, $p_i \neq q_i$. Uveďme tvrzení, které nám umožní testovat hypotézu H_0.

Tvrzení 5. *Jestliže* $(X_1, \ldots, X_d) \sim M(m; p_1, \ldots, p_d)$, *pak* Pearsonova statistika

$$\chi^2 = \sum_{i=1}^{d} \frac{(X_i - mp_i)^2}{mp_i}$$

má při $m \to \infty$ *asymptoticky rozdělení* χ^2_{d-1}.

Důkaz. Viz (Anděl [10], s. 270). □

Proto pokud platí $\chi^2 \geq \chi^2_{d-1}(\alpha)$, zamítáme na hladině α hypotézu H_0. Ovšem tento test je asymptotický, a tak lze použít pouze při dostatečně velkém výběru m. V literatuře je obvykle uváděna podmínka, aby pro každé $i \in \{1, \ldots, d\}$ platilo $mp_i \geq 5$. V současné době se však častěji vyžaduje splnění slabšího *Yarnoldova kritéria*, které požaduje, aby platilo

$$mp_i \geq \frac{5r}{d} \text{ pro všechna } i = 1, \ldots, d \text{ při } d \geq 3,$$

kde r je počet dnů, pro které platí $mp_i < 5$.

Vzhledem k tomu, že programem budeme zpracovávat data narození všech obyvatel ČR, jsou obě dvě kritéria bez problémů splněna. Nemusíme tedy uvádět další kritéria či testy, které lze použít v případě, kdy *Yarnoldovo kritérium* není naplněno. Další testy a kritéria lze najít například v Simonoff [11].

3.2 Metody výpočtu

V této sekci uvedeme, jaké jsou možné metody pro tvorbu programu a jaké jsou jejich výhody a nevýhody. Jak bude program vypadat? Vstupní data programu budou počty narození lidí v jednotlivých dnech roku. Program odhadne pravděpodobnosti narození pro jednotlivé dny metodou výběrových průměrů. Na jejich základě spočítá pravděpodobnost $R_n(\mathbf{p})$, přičemž předpokládáme, že všichni žáci ve třídě spadají do skupiny lidí, od kterých byla získána vstupní data. Program by měl dále testovat hypotézu, že pravděpodobnosti narození jsou stejné pro každý den.

První vzorec v textu, který by nám umožňoval spočítat pravděpodobnost $R_n(\mathbf{p})$, je vzorec (2.1). Jeho zjevnou nevýhodou je sčítání obrovského množství členů, konkrétně $\binom{d}{k}$ členů. I když jsem věděl, že toto nebude správná cesta, napsal jsem pro ilustraci krátký program, který by pravděpodobnost $R_n(\mathbf{p})$ počítal tímto způsobem. Zdrojový text programu lze najít v *příloze 2*. Pro $d = 365$ už pro $n = 5$ trvalo programu půl hodiny, než se dobral k výsledku. Program musel sečíst

$$\binom{365}{5} \approx 5,3 \cdot 10^{10}$$

členů. Vzhledem k tomu, že pro $n = 20$ by se jednalo už o

$$\binom{365}{20} \approx 4,3 \cdot 10^{32}$$

členů, potvrzuje se, že toto opravdu není schůdná cesta a použití sofistikovanějšího vzorce je nutné.

Další způsob výpočtu $R_n(\mathbf{p})$ jsme se dozvěděli ve článku Nunnikhovena [7]. Jednalo by se o modifikaci již zmíněného programu `Tnsp` z *přílohy 1*. Rozhodl jsem se ale pro třetí cestu.

Třetí možný způsob výpočtu $R_n(\mathbf{p})$ zmínil Mase [8]. Jedná se o vztah (2.15) a rekurentní vztah (2.16). Díky jednodušší implementaci jsem zvolil vzorec (2.16). Mase ve svém článku [8] píše, že pomocí tohoto vzorce lze jednoduše provést výpočet, pokud dokážeme násobit velmi velká a velmi malá čísla s vysokou přesností. Například pro $n = 25$ násobíme čísla velikosti řádově 10^{61} a 10^{-62}. To tehdejší výpočetní prostředky s dostatečnou přesností neumožňovaly, proto tento vzorec Mase nedoporučoval. Nyní jsou však výpočetní prostředky na dostatečné úrovni, aby toto bez problémů zvládly. Při výpočtu pravděpodobnosti $R_n(\mathbf{p})$ budeme postupně počítat $R_1(\mathbf{p}), R_2(\mathbf{p}), \ldots, R_{n-1}(\mathbf{p})$, protože k výpočtu $R_l(\mathbf{p}), l \in \{1, \ldots d\}$ potřebujeme znalost $R_1(\mathbf{p}), \ldots, R_{l-1}(\mathbf{p})$.

Nyní víme, jak spočítat pravděpodobnosti $R_n(\mathbf{p})$. Potřebujeme ještě testovat hypotézu o stejných pravděpodobnostech narození. Tyto testy jsou již naprogramovány ve statistických programech, a tak jejich použití nám ušetří nemálo práce. Rozhodl jsem se použít program R, který umožňuje i vytváření vlastních funkcí. Jeho nevýhodou je, že cykly v něm mají relativně velkou časovou náročnost oproti méně sofistikovaným programům. Vzhledem k tomu, že budeme hojně používat cyklus *for*, mohlo by to způsobovat komplikace. Ukazuje se, že pro standartní problém, $d = 365$ je výpočetní doba v řádu desítek sekund, což je uspokojivé. Pro aplikace problému by však bylo vhodnější naprogramovat část, ve které se počítá $R_n(\mathbf{p})$, v nějakém méně vyspělém jazyce.

3.3 Funkce `narozeniny`

Nyní popíšeme vytvořený program. Jedná se o funkci `narozeniny` v jazyce R. Funkci lze nalézt v *příloze 3*. V programu R spustíme skript `narozeniny.r` a již můžeme tuto funkci využívat. Povinnou vstupní hodnotou funkce je počet žáků ve třídě. Další proměnné mají nastavené implicitní hodnoty, a tak je jejich specifikování volitelné.

První volitelnou proměnnou je `dnu` — počet dnů v roce se standardní hodnotou 365. Dále se jedná o volitelný parametr pravděpodobností `q`. Funkce `narozeniny` totiž nejenže dokáže testovat, zda jsou pravděpodobnosti stejné, ale i zda jsou rovny nějakému konkrétnímu vektoru pravděpodobností `q` s počtem složek rovným d. Této vlastnosti využijeme například při práci s daty narození získaných jak v přestupných, tak v nepřestupných rocích. Pokud vektor `q` není zadán, uvažujeme vektor stejných pravděpodobností $(\frac{1}{d}, \ldots, \frac{1}{d})$.

Dalším vstupním parametrem je hladina testu hypotézy, že skutečné pravděpodobnosti se rovnají vektoru `q`, standartní hodnota proměnné `hladina` je $0,05$. Posledním údajem je odkaz na uložená data v proměnné `data`, implicitním souborem je `data_narození.csv` uložený v pracovním adresáři programu R. O požadavcích na podobu tohoto souboru se zmíníme později.

Pokud je zadán vyšší počet žáků než je počet dní v roce, funkce ihned vrátí hledanou pravděpodobnost — tedy 0. V opačném případě má program šest částí, které krátce popíšeme.

Nejprve načteme data. Data by měla být v ASCII podobě v kódování Unicode (tedy `UTF-16LE`), jednotlivé hodnoty od sebe musejí být odděleny středníky a být uloženy na jednom řádku. To znamená, že soubor obsahující data musí mít první hodnotu uloženou na první řádce, nesmí obsahovat znaky pro ukončení řádku mezi jednotlivými hodnotami a naopak musí být tímto znakem ukončen.

Následně spočteme hodnotu Pearsonova χ^2 testu. Pokud bude jeho p–hodnota vyšší než zvolená hladina, nezamítáme nulovou hypotézu, že skutečné pravděpodobnosti narození jsou rovny $\mathbf{q}$. V opačném případě nulovou hypotézu zamítáme.

Další část se uskuteční pouze za předpokladu, že vektor $\mathbf{q}$ je vektor stejných pravděpodobností a zároveň nezamítáme nulovou hypotézu. V tomto případě spočteme pravděpodobnost R_n za platnosti nulové hypotézy podle vzorce pro *základní problém.* Následující část již je společná pro všechny možnosti, odhadneme v ní pravděpodobnosti narození $\mathbf{p}$ pomocí výběrových průměrů. Z nich dále spočítáme hodnoty pole P. J-tá hodnota tohoto pole značí pravděpodobnost

$$P_j = \sum_{k=1}^{d} p_k^j.$$

Nyní již můžeme spočítat hodnoty $R_n(\mathbf{p})$ podle vzorce 2.16 a vypsat výsledek. Pokud bylo zadáno více žáků, než je dnů, je výstupem pouze 0. V opačném případě je výstupem vektor, jehož první složka je rovna pravděpodobnosti $R_n(\mathbf{p})$, kde $\mathbf{p}$ je vektor pravděpodobností narození odhadnutý výběrovým průměrem. Druhá složka je rovna 1, pokud zamítáme nulovou hypotézu, a rovna 0, pokud nulovou hypotézu nezamítáme. V druhém případě a za předpokladu, že byl testován vektor stejných pravděpodobností, je výstupem i třetí složka, a to pravděpodobnost R_n za platnosti nulové hypotézy, tedy pro *základní problém.*

Pro srozumitelnost jsem napsal i funkci `narozenipopis`, která je uvedena v *příloze 4*. Tato funkce používá funkci `narozeniny` a slovně interpretuje její výsledky. Poté co jsme v programu R spustili skript `narozeniny.r`, spustíme skript `narozeninypopis.r` a můžeme tuto funkci využívat. Funkce `narozenipopis` má stejné vstupní parametry jako funkce `narozeniny`.

3.4 Zpracování dat

V této sekci zpracujeme reálná data narození občanů České republiky. Ještě předtím než se dáme do zpracování reálných dat, nastíníme, proč lze očekávat, že násobení velmi velkých a velmi malých čísel nezpůsobuje chybu ve výpočtu.

Vytvořil jsem soubor, který měl všech 365 hodnot stejných. Pro $n = 5$ byl výstup z funkce `narozeniny`

$$0.9728644 \quad 0 \quad 0.9728644.$$

Vyšly tedy stejné hodnoty oběma metodami. Stejně tomu bylo i pro $n = 10, 15, \ldots, 100$. Díky prováděným operacím ve výpočtu pravděpodobnosti pro *základní problém* lze považovat tento výpočet za velmi přesný. U metody rekurzivního výpočtu $R_n(\mathbf{p})$ nevedlo položení $\mathbf{p} = (\frac{1}{d}, \ldots, \frac{1}{d})$ ke zjednodušení výpočtu. Lze tedy očekávat, že tato metoda bude přesná i pro jiná $\mathbf{p}$.

Data jsem získal z webu Ministerstva vnitra České republiky [12] v sekci *Statistika lokalit a dnů narození* a položkou *Kraje*. Přímý odkaz na soubor dne 2011-07-13 byl: `http://www.mvcr.cz/soubor/leden-iso-zdkr-csv.aspx`. Soubor je i součástí CD přiloženého k práci. Ze souboru jsem použil jenom celkový součet dat z krajů — data počtu narození lidí pro všechny dny a celou Českou republiku.

Z těchto dat jsem nejprve vybral data obsahující všechna data kromě 60. položky, která znázorňuje počet lidí narozených 29. února. Na tyto data jsem zavolal funkci `narozeniny` pro počet žáků $n = 5, 10, \ldots, 100$. Zamítáme nulovou hypotézu o stejných pravděpodobnostech narození. Výsledné hodnoty pravděpodobností jsem označil $R_n(p_{cr365})$. Srovnání s pravděpodobnostmi pro $d = 365$ a *základní problém*, které označíme $R_n(p_{u365})$, nabízí tabulka 3.1.

Nyní vezměme všechny data včetně počtu lidí narozených 29. února. Pro jednoduchost utvořme vektor $\mathbf{q}$ délky 366, který má hodnotu $\frac{4}{1461}$ pro všechny složky kromě šedesáté, která má hodnotu $\frac{1}{1461}$. Tyto hodnoty reflektují, že je přestupný zpravidla každý čtvrtý rok. Nyní zavolejme funkci `narozeniny` s danými parametry:

$$\texttt{narozeniny(n, dnu = 366, q = q)},$$

kde $n = 5, 10, .., 100$. Opět zamítáme nulovou hypotézu, že skutečné pravděpodobnosti narození odpovídají vektoru $\mathbf{q}$. Takto získané hodnoty pravděpodobnosti $R_n(\mathbf{p})$ označme $R_n(p_{cr366})$. V tabulce 3.1 vidíme srovnání s pravděpodobnostmi pro $d = 366$ a *základní problém*, které značíme $R_n(p_{u366})$.

Z tabulky 3.1 vidíme lehce překvapivou věc — nejmenší rozdíl mezi pravděpodobnostmi je pro česká data s 29. únorem a pro stejné pravděpodobnosti narození bez 29. února. Víme, že $R_n(\mathbf{p})$ je největší za situace *základního problému*, odtud plyne $R_n(p_{cr366}) \leq R_n(p_{u366})$ a $R_n(p_{u365}) \geq R_n(p_{cr365})$. Se vzrůstajícím počtem dnů se pravděpodobnost $R_n(\mathbf{p})$ snižuje, ale v našem případě ne o tolik, aby to

Tabulka 3.1: Hodnoty $R_n(\mathbf{p})$ pro české a uniformní pravděpodobnosti a $d = 365, 366$

n	$R_n(p_{u365})$	$R_n(p_{cr365})$	$R_n(p_{cr366})$	$R_n(p_{u366})$
5	0,9728644	0,9727624	0,9727935	0,9729379
10	0,8830518	0,8826353	0,8827635	0,8833546
15	0,7470987	0,7462764	0,7465325	0,7477021
20	0,5885616	0,5873897	0,5877592	0,5894304
25	0,4313003	0,4299448	0,4303774	0,4323156
30	0,2936838	0,2923461	0,2927783	0,2946966
35	0,1856168	0,1844611	0,1848391	0,1865016
40	0,1087682	0,1078811	0,1081749	0,1094552
45	0,0590241	0,05841364	0,05861839	0,05950254
50	0,02962642	0,02924766	0,02937632	0,02992693
55	0,01373771	0,01352505	0,01359822	0,01390858
60	0,005877339	0,005769032	0,005806782	0,005965504
65	0,002316893	0,00226679	0,002284483	0,002358230
70	0,000840424	0,0008193551	0,0008268941	0,0008580496
75	0,0002801218	0,0002720659	0,0002749873	0,0002869581
80	$8{,}566805\ 10^{-5}$	$8{,}286759\ 10^{-5}$	$8{,}389698\ 10^{-5}$	$8{,}807979\ 10^{-5}$
85	$2{,}400267\ 10^{-5}$	$8{,}286759\ 10^{-5}$	$2{,}344763\ 10^{-5}$	$2{,}477625\ 10^{-5}$
90	$6{,}151644\ 10^{-6}$	$5.897788\ 10^{-6}$	$5{,}993709\ 10^{-6}$	$6{,}377101\ 10^{-6}$
95	$1{,}439829\ 10^{-6}$	$1{,}373733\ 10^{-6}$	$1{,}399062\ 10^{-6}$	$1{,}499484\ 10^{-6}$
100	$3{,}072489\ 10^{-7}$	$2{,}916485\ 10^{-7}$	$2{,}977127\ 10^{-7}$	$3{,}215642\ 10^{-7}$

vykompenzovalo rozdíl mezi českými daty a stejnými pravděpodobnostmi. Platí tedy $R_n(p_{u366}) \geq R_n(p_{u365}) \geq R_n(p_{cr366}) \geq R_n(p_{cr365})$pro $n = 5, 10, \ldots, 100$.

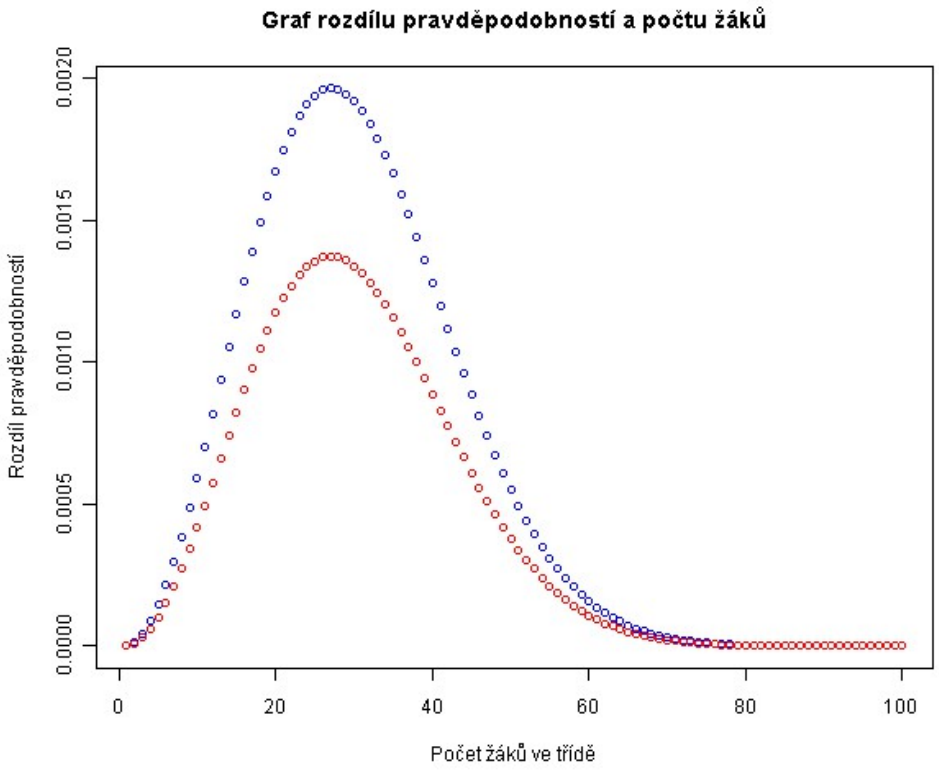
Možná, že zajímavější než samotné hodnoty pravděpodobností, jsou hodnoty rozdílu mezi jednotlivými pravděpodobnostmi. Nejprve se podívejme na rozdíly mezi pravděpodobnostmi $R_n(\mathbf{p})$ pro česká data a pro *základní problém* na obrázku 3.1. Modře je uveden rozdíl $R_n(p_{u366}) - R_n(p_{cr366})$ a červeně rozdíl $R_n(p_{u365}) - R_n(p_{cr365})$ pro $n = 1, 2, 3, \ldots, 100$. Rozdíl mezi pravděpodobnostmi $R_n(\mathbf{p})$ je vyšší pro data s 29. únorem, neboť v tomto případě se pravděpodobnosti narození více odlišují od stejných pravděpodobností. I tak však rozdíl pravděpodobností $R_n(\mathbf{p})$ nepřekračuje 0,002.

Na obrázku 3.2 vidíme rozdíl pravděpodobnosti $R_n(\mathbf{p})$ pro česká data bez 29. únoru a s 29. únorem, tedy $R_n(p_{cr365}) - R_n(p_{cr366})$. Vidíme, že 29. únor má na pravděpodobnost $R_n(\mathbf{p})$ nižší vliv než $0,0005$ pro všechna $n = 1, 2, \ldots, 100$.

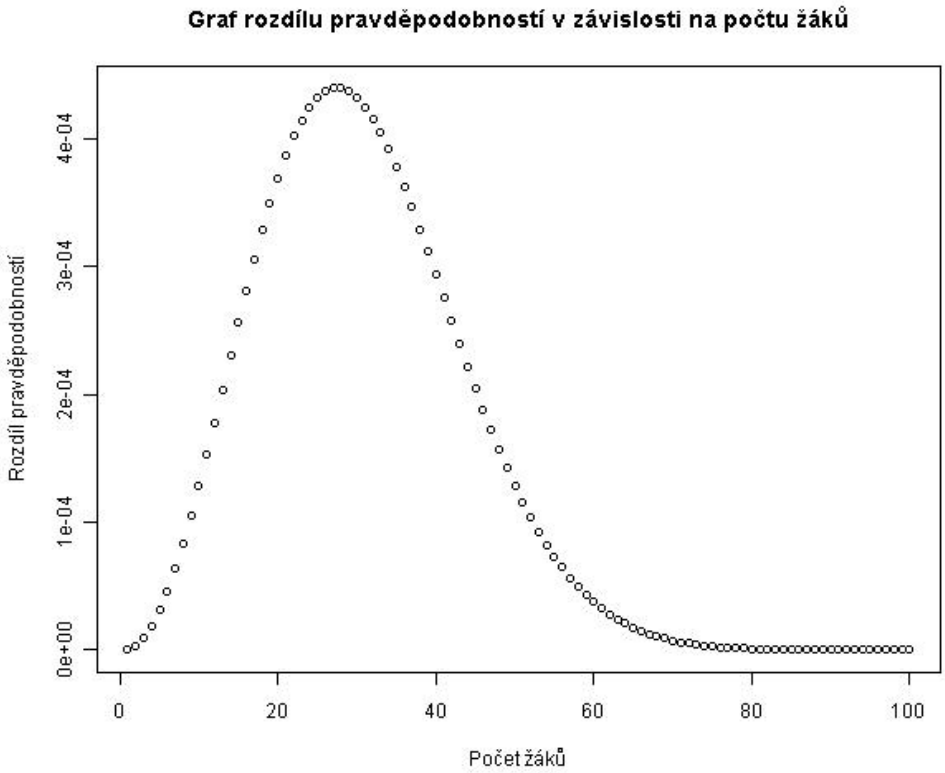
Na závěr se podívejme, jak jsou počty narození rozděleny v průběhu roku. To ukazuje obrázek 3.3. Borja a Haigh ve svém článku [13] zmiňují evropský model, ve kterém je největší počet narození na jaře a začátkem léta s lokálním maximem počtu narození v září. Naproti tomu v americkém modelu je toto lokální maximum v září maximem globálním. Z obrázku 3.3 vidíme, že data narození obyvatel České republiky odpovídají evropskému modelu.

Při analýze obrázku 3.3 je důležité si uvědomit, že se jedná o data sebrané za mnoho let. Pokud bychom analyzovali data pouze za jeden rok, pravděpodobně bychom shledali, že mají vyšší výběrový rozptyl. A to například díky tomu, že o víkendech bývá v porodnicích o něco méně pracovní síly a provádí se méně

Obrázek 3.1: Rozdíl pravděpodobnosti $R_n(\mathbf{p})$ pro česká data a pro *základní problém* v závislosti na počtu žáků



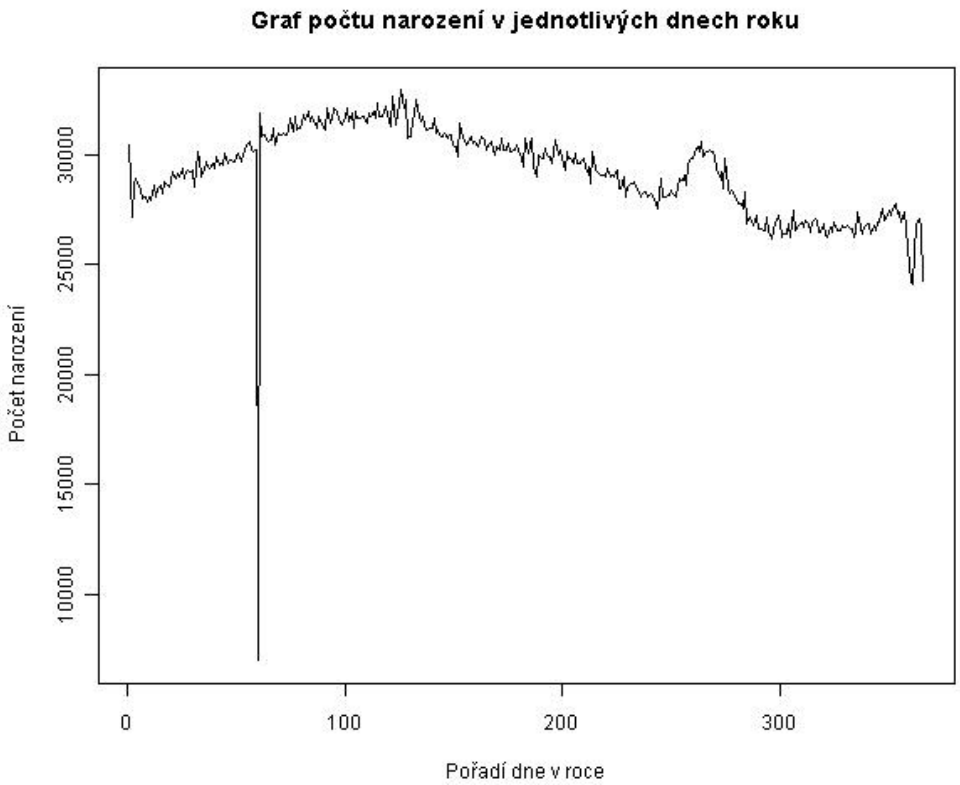
Obrázek 3.2: Rozdíl pravděpodobnost $R_n(\mathbf{p})$ pro česká data bez 29. února a s 29. únorem v závislosti na počtu žáků



porodů císařským řezem. Díky tomuto faktu naopak v pondělí a v úterý bývá porodů více. Tím, že jsou data sebrané z delšího časového úseku, je tento vliv umenšen.

Z obrázku 3.3 je ihned zřetelný propad v počtu narození 29. února. Dále vidíme velkou fluktuaci v počtu narození koncem roku. Když se podíváme na počty narození v závěru roku, zjišťujeme, že 25., 26. a 31. prosince je méně porodů oproti okolním dnům. To koresponduje se dny, na které by si personál mohl chtít brát více volna. Naopak 1. ledna je porodů více, než je průměrné. To by mohl být vliv různých odměn, které jsou vypláceny pro první narozené děti daného roku. Toto zjištění mne vedlo k tomu, abych se podíval na počty narození ve dnech, které byly státními svátky po valnou většinu sledovaného období, konkrétně počty narození 1. května a 28. října. Avšak zde jsem nepozoroval výrazný pokles v počtu narození oproti okolním dnům.

Obrázek 3.3: Počty narození v závislosti na dnu v roce



4. Aplikace problému

V této kapitole představíme některé aplikace *problému shody narozenin*. Existují některé přímé aplikace tohoto *problému*, kde n skutečně značí počet žáků ve třídě a d počet dnů v roce. Příkladem může být situace, kdy se každému žákovi dává k narozeninám dort, a nás bez znalosti narozenin žáků zajímá, zda bude stačit mít připraveno na každý den v roce pouze jeden dort.

Častějšími aplikacemi jsou ale případy, kdy d je obecně počet hodnot nějakého jevu a n velikost náhodného výběru. *Problém shody narozenin* můžeme vyjádřit i pomocí příkladu s kuličkami a přihrádkami, který se často používá v teorii pravděpodobnosti. Dny můžeme chápat jako přihrádky s neomezenou kapacitou, žáky ve třídě jako kuličky, které umisťujeme s určitou pravděpodobností do těchto přihrádek. Zajímá nás pak pravděpodobnost, že v žádné přihrádce není dvě a více kuliček.

Jako zástupce aplikací *obecného problému shody narozenin* na poli biologie zmiňme článek Gail [14], který popisuje chování speciálního druhu enzymu. Tento článek je cenný i tím, že přinesl matematický náhled na tuto problematiku, který dále rozvinul Nunnikhoven ve svém již zmiňovaném článku [7]. Z dalších oborů, ve kterých *problém shody narozenin* našel uplatnění, zmiňme například kryptografii a jeho využití v oblasti *hashovacích funkcí* viz [15]. V následující sekci zmíníme *problém shody příjmení*, který se odehrává za podobné situace jako *problém shody narozenin*.

4.1 Problém shody příjmení

V *problému shody příjmení* značí d počet všech příjmení obecně velké skupiny lidí, ze které je náhodně vybráno n osob. Výrazem R_n zde míníme pravděpodobnost, že mezi těmito n osobami neexistuje dvojice se stejným příjmením. Tento problém byl motivací pro sepsání již zmiňovaného článku profesora Mase [8]. Zabýval se *problémem shody příjmení* v zemi svého původu — v Japonsku. Uvedeme těžkosti, se kterými se musel potýkat, a výsledky, ke kterým došel.

K tomu, abychom mohli spočítat pravděpodobnost pro *problém shody příjmení*, musíme umět odhadnout pravděpodobnosti toho, že náhodná osoba bude mít dané příjmení. Mase ale neměl k dispozici data počtů jednotlivých příjmení, aby mohl pravděpodobnosti odhadnout výběrovým průměrem. Dokonce ani neznal přesný počet těchto příjmení, jen věděl, že jich je kolem 120 000. Protože neměl k dispozici lepší data, musel pracovat s počty uzavřených pojistek pro dané příjmení získaných od dvou pojišťoven. Někteří lidé uzavřeli více pojistek, tedy se v seznamu objevili vícekrát. To nebylo možno rozpoznat, a tak autor musel doufat, že toto nezpůsobí velkou chybu. Další těžkost souvisela s reprezentací dat v počítači. K tomuto, abychom ji porozuměli, udělejme si krátkou odbočku do japonského jazyka.

V japonštině existují dva druhy abecedy, a to *hiragana* a *katakana*. Oběma těmito abecedami lze popsat všech 45 japonských slabik a hlásku „n“. Navíc jsou v japonštině znaky *kanji*, jejichž přesný počet není znám, ve velkých slovnících jich lze nalézt kolem 50 000. Tyto znaky či jejich kombinace představují jednotlivá slova. Výslovnost těchto znaků lze popsat pomocí některé z abeced (používá

se *hiragana*). Často se stává, že znaky, které mají rozdílný význam, mají stejné čtení. A zde leží jádro těžkosti profesora Mase. Z počítače měl k dispozici pouze výslovnosti příjmení. Nebyl tedy schopen rozpoznat, jak se dané příjmení zapisuje pomocí znaků *kanji*, tedy o jaké příjmení se jedná. Musel přistoupit ke zjednodušení a všechna příjmení se stejnou výslovností brát jako totožná. Například příjmení *Ito*, které má dva běžné zápisy a 16 vzácných zápisů pomocí znaků *kanji*, reprezentuje všech těchto 18 zápisů dohromady.

Tabulka 4.1 ukazuje pravděpodobnosti nejčastějších příjmení získaných z dat uzavřených životních pojistek u společnosti Daiiti. Z tabulky 4.1 vyplývá, že několik nejčastějších příjmení má pravděpodobnosti dosti vysoké. Pravděpodobnosti u dalších příjmení rychle klesají a trvá velmi dlouho, než se částečný součet pravděpodobností přiblíží k jedné. Ještě součet pravděpodobností 5000 nejčastějších příjmení má celkovou pravděpodobnost jen 92,3%. Je tedy velmi mnoho příjmení, které jsou poměrně dost četná. Z tohoto důvodu Mase nepoužil žádné z běžných pravděpodobnostních rozdělení.

Tabulka 4.1: Pravděpodobnosti nejčastějších japonských příjmení

Příjmení	Sato	Suzuki	Takahashi	Tanaka
Pravděpodobnost v %	1,583	1,332	1,132	1,061
Příjmení	Watanabe	Ito	Nakamura	Yamamoto
Pravděpodobnost v %	1,007	0,950	0,864	0,856
Příjmení	Kobayashi	Saito	Kato	Yoshida
Pravděpodobnost v %	0,812	0,799	0,720	0,670
Příjmení	Yamada	Sasaki	Matsumoto	Yamaguchi
Pravděpodobnost v %	0,661	0,590	0,527	0,519
Příjmení	Kimura	Inoue	Abe	Hayashi
Pravděpodobnost v %	0,476	0,470	0,464	0,425

Mase pozoroval, že data četnosti prvních 200 příjmení jdou velmi dobře aproximovat funkcí tvaru

$$f(n) = d/n^{\alpha},$$

kde α je kolem 0,7. Pro takovéto α by řada $\sum_{n=1}^{\infty} f(n)$ divergovala, proto funkci $f(n)$ Mase vynásobil c^n, kde $c < 1$. Díky tomu pro odhad pravděpodobností příjmení zvolil funkci tvaru

$$p(n; a, b, c, d) = d\frac{c^n}{(n+b)^a}, \ n = 1, 2, \ldots, 120000.$$

Aby se skutečně jednalo o pravděpodobnosti, musí platit

$$d = \left(\sum_{n=1}^{120000} \frac{c^n}{(n+b)^a} \right)^{-1}. \tag{4.1}$$

Když metodou lineární regrese proložíme pravděpodobnosti nejčastějších dvou set příjmení funkcí $p(n; a, b, c, d)$, dostaneme $c = 1,002$. To, že platí $c > 1$, znemožňuje extrapolaci $p(n; a, b, c, d)$ pro $n > 200$. Proto Mase zvolil jiný postup. Pro každou hodnotu c, která byla menší než 1 a zároveň blízko 1 — abychom

dosáhli pomalého snižování $p(n; a, b, c, d)$ — spočítal metodou lineární regrese parametry a, b, d. Vybral to c, pro které

$$\sum_{n=1}^{120000} p(n; a, b, c, d)$$

bylo nejblíže jedné. Dostal $a = 0,7570, b = 3,648, c = 0,9996$. Dosazením do 4.1 získáváme $d = 0,05049$. S takovýmito hodnotami dostáváme pro data ze životních pojistek u společnosti Daiiti koeficient determinace $R^2 = 99,27\%$ a pro data společnosti Daihyaku $R^2 = 97,98\%$.

Nyní již máme odhadnuté pravděpodobnosti jednotlivých příjmení a pomocí rekurzivního vztahu 2.16 můžeme spočítat pravděpodobnosti R_n pro *problém shody příjmení*, že mezi n osobami neexistuje dvojice se stejným příjmením. Ty ukazuje tabulka 4.2.

Dá se předpokládat, že pravděpodobnosti narození žáků v Japonsku se od stejných pravděpodobností neodlišují výrazně výše než v ČR. Za tohoto předpokladu můžeme porovnat tabulky 3.1 a 4.2. Zjišťujeme, že v Japonsku ve třídě o n žácích je pravděpodobnost, že ve třídě neexistuje dvojice se stejnými narozeninami, nižší než pravděpodobnost, že ve třídě neexistuje dvojice se stejným příjmením. Rozdíl však není tak velký, jak by se na první dojem mohlo zdát, a to zejména pro nízký počet žáků ve třídě.

Tabulka 4.2: Hodnoty R_n pro *problém shody příjmení*

Počet osob	5	10	15	20	25	30
Pravděpodobnost R_n	0,9786	0,9086	0,8019	0,6740	0,5405	0,4140
Počet osob	35	40	45	50	60	70
Pravděpodobnost R_n	0,3035	0,2130	0,1434	0,0927	0,0344	0,0110
Počet osob	80	90	100			
Pravděpodobnost R_n	0,0030	0,0007	0,0001			

Závěr

Na závěr uveďme, jakým směrem bychom se v řešení *problému shody narozenin* mohli dále ubírat. Mohli bychom zkoumat konvergenci výrazů $T(n, s, \mathbf{p})$ ze článku Nunnikhovena [7] a zjistit předpoklady, za kterých lze pravděpodobnost $R_n(\mathbf{p})$ aproximovat pomocí součtu několika prvních výrazů $T(n, s, \mathbf{p})$.

Zajímavé by bylo získat data ze skutečných tříd a zjistit, o kolik se odhadovaná pravděpodobnost $R_n(\mathbf{p})$ liší od skutečné. Zjistili bychom, jak moc velkou chybu způsobuje předpoklad, že den a měsíc narození jednotlivých žáků jsou nezávislé jevy. Mezi další vlivy, které ovlivňují skutečnou pravděpodobnost, může patřit i to, že žáci jsou většinou podobného stáří a často pocházejí ze společného mikroregionu. Data jsou naopak sbírána z větších správních celků a často za delší časové období.

Uveďme ještě variaci *problému shody narozenin.* Pokud by nás zajímala pravděpodobnost, že neexistuje dvojice žáků, která se narodila ve stejném kalendářním týdnu, jednalo by se pouze o analogii *problému shody narozenin.* Pravděpodobnost narození bychom sečetli pro celý týden dohromady a d by se v tomto případě bylo rovno počtu týdnů. Pokud by nás ale zajímalo, zda existuje dvojice žáků, která má narozeniny v libovolných k po sobě jdoucích dnech, jednalo by se už o nový typ problému.

Seznam použité literatury

[1] ARNOLD, Barry. Majorization: Here, There and Everywhere. *Statistical Science.* 2007, 22, 3, s. 407-413.

[2] COMTET, Louis. *Advance combinatorics : the art of finite and infinite expansions.* [s.l.] : Springer, 1974. 343 s. ISBN 9027704414.

[3] ROMAN, Steven. *The Umbral Calculus.* New York : Academic Press, 1984. s. 193. ISBN 0125943806.

[4] *Český statistický úřad* [online]. c2011, 12.4.2011 [cit. 2011-04-30]. Vícečetné porody v Jihomoravském kraji v letech 1991 až 2006. Dostupné z WWW: `<http://www.czso.cz/xb/redakce.nsf/i/vicecetne_porody_v_jihomoravskem_kraji_v_letech_1991_az_2006>`.

[5] FELLER, William. *An Introduction to Probability Theory and Its Applications.* 3rd edition. New York : John Wiley, 1968. 509 s. ISBN 9780471257080.

[6] JOAG-DEV, Kumar; PROSCHAN, Frank. Birthday Problem with Unlike Probabilities. *The American Mathematical Monthly.* 1992, 99, 1, s. 10-12. Dostupný také z WWW: `<http://www.jstor.org/stable/2324542>`.

[7] NUNNIKHOVEN, Thomas S. . A Birthday Problem Solution for Nonuniform Birth Frequencies. *The American Statistician.* 1992, 46, 4, s. 270-274. Dostupný také z WWW: `<http://www.jstor.org/stable/2685309>`.

[8] MASE, Shigeru. Approximations to the birthday problem with unequal occurrence probabilities and their application to the surname problem in Japan. *Annals of the Institute of Statistical Mathematics.* 1992, 44, 3, s. 479-499.

[9] Bell polynomials. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 2 April 2004, last modified on 30 June 2011 [cit. 2011-07-05]. Dostupné z WWW: `<http://en.wikipedia.org/wiki/Bell_polynomials>`.

[10] ANDĚL, Jiří. *Základy matematické statistiky.* Druhé opravené vydaní. Praha : MATFYZPRESS, 2007. 358 s. ISBN 80-7378-001-1.

[11] SIMONOFF, Jeffrey S. *Analyzing Categorical Data.* 1st edition. New York : Springer, 2003. 511 s. ISBN 9780387007496.

[12] *Ministerstvo vnitra České republiky* [online]. c2010, 2.3.2011 [cit. 2011-07-13]. Četnost jmen a příjmení - Ministerstvo vnitra České republiky. Dostupné z WWW: `<http://www.mvcr.cz/clanek/cetnost-jmen-a-prijmeni-722752.aspx>`.

[13] BORJA, Mario Cortina; HAIGH, John. The birthday problem. *Significance.* 2007, 4, 3, s. 124-127.

[14] GAIL, M.H., et al. A Solution to the Generalized Birthday Problem with Application to Allozyme Screening for Cell Culture Contamination. *Journal of Applied Probability.* 1979, 16, 2, s. 242-251.

[15] VAN OORSCHOT, Paul C; WIENER Michael J. Parallel Collision Search with Application to Hash Functions and Discrete Logarithms. *2nd ACM Conference on Computer and Communications Security*, Fairfax, Virginia, USA. 1994, s. 210-218. Dostupný také z WWW: `<www.scs.carleton.ca/~paulv/papers/acmccs94.pdf>`.

Seznam tabulek

1.1 Součty největších složek vektorů 6

2.1 Hodnoty R_n a jeho odhadů pro $d = 365$ 12
2.2 Hodnoty $T(n, s, \mathbf{p})$ pro pravděpodobnost jedna narození 1. ledna . 21
2.3 Hodnoty $T(n, s, \mathbf{p})$ pro narození v prvních dnech roku 22

3.1 Hodnoty $R_n(\mathbf{p})$ pro české a uniformní pravděpodobnosti 31

4.1 Pravděpodobnosti nejčastějších japonských příjmení 35
4.2 Hodnoty R_n pro *problém shody příjmení* 36

Přílohy

Příloha 1

Kód v jazyku Pascal, použit byl překladač FreePascal.

```
const dnu = 365;    //pocet dnu v roce
var p: array[1..dnu] of extended;    //pravdepodobnosti narozeni pro jednotlive
    dny v roce
var n: byte;    //pocet zaku ve tride
var i,j: integer;    //pomocne promenne
var e: array[1..dnu] of extended;    // i-ta pozice znazornuje odchylku p_i od
    stejnych pravdepodobnosti narozeni
var R: extended;    // pravdepodobnost, ze s n zaky ve tride neexistuje dvojice zaku,
    ktera ma narozeniny ve stejny den
var O: array[0..dnu,0..dnu] of extended;    //pole O pouzivane pri
    rekurzivnim vypoctu
var Tn: array[0..dnu] of extended;    //s-ta pozice je hodnota T(n,s,p)

procedure UrceniPravdepodobnosti;    //volime pravdepodobnosti narozeni zaku
begin
  for i := 1 to 300 do
    p[i] := 1/300;
  for j := 301 to 365 do
    p[j] := 0;
  for i := 1 to dnu do
    e[i] := p[i] - (1 / dnu);    //vypocet odchylky od stejnych pravdepodobnosti
end;

procedure UrceniPoleO;    //nastavime pocatecni podminky pole O
begin
  O[0,dnu] := 1;    //aby vzorec v procedure SpocteniPravdepodobnosti fungoval i pro
      s = 0, musime takto dodefinovat; pri vypoctu dalsich clenu pole O se tato hodnota
      nevola
  O[1,1] := e[1];
  for i := 1 to (dnu - 1) do begin    //zadavame pocatecni podminky pole O
    O[1,i+1] := O[1,i] + e[i+1];
    O[i+1,i+1] := O[i,i] * e[i+1];
  end;
end;

function VypocetOdchylky(a,b: integer):real;    //vraci hodnotu O(a,b)
begin
  if (O[a,b] <> 0) then    //pokud jsme uz hodnotu O(a,b) spocitali, nemusime nic
      pocitat
    VypocetOdchylky := O[a,b]
  else begin
    O[a,b] := VypocetOdchylky(a,b-1) + e[b] *
        VypocetOdchylky(a-1,b-1); //vypocet podle rekurzivniho vzorce pro pole
        O
    VypocetOdchylky := O[a,b];
  end;
end;

procedure SpocteniPravdepodobnosti;    //vypocet pole Tn a celkove
    pravdepodobnosti R
```

```
var nasobek: qword;     //ve vzorci pro T(n,s,p) tato promenna reprezentuje n!/(n-s)!;
    qword je 8-bitova promenna nabyvajici celych hodnot od 0 do asi 1,8·10^19
var cinitel: extended;     //ve vzorci pro T(n,s,p) tato promenna reprezentuje
    d^(n-s)·(d-s)!/(d-n)!
var s:byte;
begin
  for s := 0 to n do begin     //spocteme T(n,s,p) (tedy Tn[s]) podle vzorce 2.7,
      souctem vsech T(n,s,p) ziskame R
    nasobek := 1;
    cinitel := 1;
    for i := 1 to s do
      nasobek := nasobek * (n + 1 - i);
    for j := 1 to n-s do
      cinitel := cinitel * (dnu - s + 1 - j) / dnu;
    Tn[s] := nasobek * cinitel * O[s,dnu];     //uplatnime vzorec pro T(n,s,p)
    R := R + Tn[s];     //R je souctem T(n,s,p) pro s od 0 do n
  end;
end;

begin
  writeln('Kolik je zaku ve tride?');
  readln(n);
  if n > 20 then writeln('Zadejte cislo nizsi nez 21.')
  else begin
    UrceniPravdepodobnosti;
    UrceniPoleO;
    for j := 1 to n do     //potrebujeme spocitat T(n,s,p) pro s=1,...,n
      VypocetOdchylky(j,dnu);
    SpocteniPravdepodobnosti;
    writeln('Hodnoty pole T(',n,'s,p) jsou po rade od nuly:');
    for j := 0 to n do
      writeln(Tn[j]);
    writeln('Pravdepodobnost, ze neexistuje dvojice zaku, ktera by
        mela narozeniny ve stejny den je: ',R);
    readln;
  end;
end.
```

Příloha 2

Kód v jazyku Pascal, použit byl překladač FreePascal.

```
const dnu = 365;     //pocet dnu v roce
var p: array[1..dnu] of extended;     //pravdepodobnosti pro jednotlive dny v roce
var zaci: array[1..dnu] of integer;     //n-ta pozice udava, kdy ma n-ty zak
    narozeniny
var n: integer;     //pocet zaku ve tride
var R: extended;     //pravdepodobnost, ze neexistuje dvojice zaku, ktera ma narozeniny
    ve stejny den
var i,j,k,l : integer;     //pomocne promenne
var pridat: real;     //pomocne promenne
var zmeneno: boolean;     //pomocna promenna

procedure urceni_pravdepodobnosti;     //zde jen testovaci data
begin
  for i := 1 to dnu do
```

```
    p[i] := (1 / dnu);
end;
procedure vypocet_pravdepodobnosti;    //vypocet podle vzorce 2.1
begin
  for i := 1 to n do    //vytvoreni pocatecni n-tice v poli zaci
    zaci[i] := i;
  while (zaci[1] < dnu - n + 2) do begin
    pridat := 1;
    for i := 1 to n do    //spocteni pravdepodobnosti, ze zaci maji narozeniny, tak
        jak je urceno v poli zaci
      pridat := pridat * p[zaci[i]];
    R := R + pridat;    //pravdepodobnost dane n-tice pridame k celkove
        pravdepodobnosti
    zmeneno := false;    //budeme hledat dalsi n-tici moznych narozenin zaku
    if zaci[n] < 365 then zaci[n] := zaci[n] + 1    //nic nam nebrani
        zvysit zaci[n]
    else begin
      for k := 1 to n - 2 do begin    //prozkoumame prvky pole zaci az na
          posledni clen (ten je roven 365) a prvni clen - ten budeme menit zvlast
        if zaci[n-k] < dnu-k  then begin    //potrebujeme najit novou
            kombinaci takovou, aby prvky pole nasledujici po zaci[n-k] byly rostouci, tedy
            zaci[n-k] musi byt mensi nez d-k
          zaci[n-k] := zaci[n-k]+1;    //zvysime zaci[n-k] o nejmensi moznou
              hodnotu
          zmeneno := true;
          for j := 1 to k do
            zaci[n-k+j] := zaci[n-k]+j;    //zmenime nasledujici prvky tak,
                aby posloupnost byla rostouci a nejmensi mozna
          break;
          end
      end;
      if zmeneno = false then begin    //neprosli jsme jen prvni clen pole zaci,
          zvysime ho, pokud projde pocatecni podminkou, najdeme novou kombinaci, jinak
          ukoncime vypocet
        zaci[1] := zaci[1]+1;
        for l := 1 to n - 1 do
          zaci[1+l] := zaci[1] + l;
      end;
    end;
  end;
end;

begin
  writeln('Kolik je zaku ve tride?');
  readln(n);
  if n > dnu then writeln('Pravdepodobnost, ze neexistuje dvojice
      zaku, ktera by mela narozeniny ve stejny den je: ')
  else begin
    urceni_pravdepodobnosti;
    vypocet_pravdepodobnosti;
    for k := 1 to n do
      R := R * k;
    writeln('Pravdepodobnost, ze neexistuje dvojice zaku, ktera by
        mela narozeniny ve stejny den je: ',R);
    readln;
  end;
end.
```

Příloha 3

Skript v programu R.

```
narozeniny <- function(pocet_zaku, dnu=365, q, hladina = 0.05,
    data="data_narozeni.csv") {
if (missing(q)) {
        q <- rep(1/dnu,dnu)        #Pokud v datech nejsou zadany pravdepodobnosti,
            ktere chceme testovat, volime stejne pravdepodobnosti.
}
P <- numeric(pocet_zaku)        #J-ta pozice vektoru znaci soucet vsech p_i
    umocnenych na j-tou mocninu.
R <- numeric(pocet_zaku+1)        #J-ta pozice vektoru znaci pravdepodobnost, ze ve
    tride o j-1 zacich neexistuje dvojice zaku se stejnym datem narozeni. Posun je nutny kvuli
    tomu, ze k vypoctu potrebujeme znat i hodnotu pro nula zaku.
o <- rep(1/dnu,dnu)

if (pocet_zaku>dnu) 0
else {
#Nacteni dat
datanarozeni <- read.table(data, fileEncoding = "UTF-16LE", sep =
    ";")

#Vypocet Pearsonova chi-kvadrat testu
nezamitame <- (chisq.test(datanarozeni,p=q)$p.value > hladina)
    #Pokud je p-hodnota Pearsonova testu vyssi nez zvolena hladina, nezamitame hypotezu o
    tom, ze pravdepodobnosti jsou rovny q.

#Spocteni pravdepodobnosti neexistence dane dvojice pro stejne pravdepodobnosti
stejne <- identical(o,q)        #Je q vektor stejnych pravdepodobnosti?
if (nezamitame & stejne) {
        ps <- 1        #Pravdepodobnost za predpokladu, ze jsou pravdepodobnosti
            narozeni stejne.
        for (i in (dnu-pocet_zaku+1):dnu) {
                ps <- ps * i/dnu        #Aplikace vzorce vypoctu pravdepodobnosti
                    pro zakladni problem
        }
}

#Spocteni vektoru pravdepodobnosti narozeni p
unlist(datanarozeni, use.names=FALSE)
celkem <- sum(datanarozeni)
p <- (1/celkem) * datanarozeni

#Spocteni P_j
pomocne_pole <- rep(1,dnu)        #Vytvorime vektor pomocne_pole ze samych
    jednicek.
for(j in 1:pocet_zaku) {        #Potrebujeme vytvorit j pozic.
        soucet <- 0
        for(i in 1:dnu) {
                pomocne_pole[[i]] <- pomocne_pole[[i]] * p[i]
                    #Pomocne_pole[[i]] obsahuje p[i] umocnene na j-1 mocninu.
                    Umocnime ho na j-tou mocninu.
                soucet <- soucet + pomocne_pole[[i]]        #Secteme
                    vsechny p umocnene na j-tou mocninu.
        }
P[[j]] <- soucet        #Priradime soucet k danemu prvku pole.
}
```

```
#Spocteni pravdepodobnosti podle vzorce 2.16
R[1] <- 1        #Znaci pravdepodobnost pro nula zaku ve tride.
for(k in 1:pocet_zaku) {        #Potrebujeme vypocist dalsich k mist vektoru R.
        for(i in 1:k) {
                nasobek <- 1        #Nasobek reprezentuje konstantu ze vzorce 2.16.
                if (i>1) {        #Pro i = 1 neni potreba nic pocitat, nasobek se ma
                    rovnat 1.
                        for(j in (k-i+1):(k-1)) {
                                nasobek <- nasobek * j * (-1)
                        }
                }
                R[k+1] <- R[k+1] + nasobek * P[[i]] * R[k-i+1]
                    #Vypocet podle vzorce 2.16. Oproti vzorci ma R o jedna vyssi index
                    protoze zaciname od indexu 1 a ne od 0 jako ve vzorci.
        }
}
if (nezamitame & stejne) x <- c(R[[pocet_zaku+1]], 0, ps)
        else if(nezamitame) x <- c(R[[pocet_zaku+1]], 0)
        else if (nezamitame == FALSE) x <- c(R[[pocet_zaku+1]], 1)
        x
}
}
```

Příloha 4

Skript v programu R.

```
narozeninypopis <- function(pocet_zaku, dnu=365, q, hladina = 0.05,
    data = "data_narozeni.csv") {
if (missing(q)) {
        q <- rep(1/dnu,dnu)        #Pokud v datech nejsou zadany pravdepodobnosti,
            ktere chceme testovat, volime stejne pravdepodobnosti.
}
if (pocet_zaku > dnu) {
        cat("Ve tride je vice zaku nez je dnu. Proto je
            pravdepodobnost 0.")
}
else {
x <- narozeniny(pocet_zaku, dnu, q, hladina, data)        #Vytvorime vektor,
    abychom pak mohli lehko odkazovat.
if (x[2] == 0 & is.na(x[3])) {
        cat("Pravdepodobnost je ", x[1] ,". Nezamitame nulovou
            hypotezu o stejnych pravdepodobnostech. \n")
}
else if (x[2] == 0) {
        cat("Pravdepodobnost je ", x[1] ,". Nezamitame nulovou
            hypotezu o stejnych pravdepodobnostech. Za platnosti
            nulove hypotezy je pravdepodobnost ", x[3] ," \n")
}
else if (x[2] == 1){
        cat("Pravdepodobnost je ", x[1] ,". Zamitame nulovou
            hypotezu o stejnych pravdepodobnostech. \n")
}
}
}
```