

EKONOMICKÁ UNIVERZITA V BRATISLAVE
FAKULTA HOSPODÁRSKEJ INFORMATIKY

Evidenčné číslo: 103006/I/2022/36124048429489156

Využitie programovacieho jazyka R
pre poistenia m -tíc osôb

Diplomová práca

2022

Bc. Szilvia Horváthová

EKONOMICKÁ UNIVERZITA V BRATISLAVE
FAKULTA HOSPODÁRSKEJ INFORMATIKY

Využitie programovacieho jazyka R
pre poistenia m -tíc osôb

Diplomová práca

Študijný program:	Aktuárstvo
Študijný odbor:	Ekonómia a manažment
Školiace pracovisko:	Katedra matematiky a aktuárstva
Vedúci záverečnej práce:	Mgr. Ing. Ingrid Krčová, PhD.

Bratislava 2022

Bc. Szilvia Horváthová

Čestné vyhlásenie

Čestne vyhlasujem, že záverečnú prácu som vypracovala sama a uviedla som všetku použitú literatúru.

Bratislava 2022

.....
(podpis študenta)

Pod'akovanie

Touto cestou by som sa chcela poďakovať vedúcej diplomovej práce Mgr. Ing. Ingrid Krčovej, PhD. za jej cenné rady, odborné vedenie a metodické usmernenie, ktoré mi poskytla pri vypracovaní tejto práce. Taktiež by som sa chcela poďakovať mojim najbližším, ktorí ma neustále podporovali a stáli pri mne.

Abstrakt

HORVÁTHOVÁ, Szilvia: *Využitie programovacieho jazyka R pre poistenia m -tíc osôb*. – Ekonomická univerzita v Bratislave. Fakulta hospodárskej informatiky; Katedra matematiky a aktuárstva. – Vedúci záverečnej práce: Mgr. Ing. Ingrid Krčová, PhD. – Bratislava: FHI EUBA, 2022, 76 s.

Cieľom tejto práce je oboznámiť čitateľa s možnosťami využitia programovacieho jazyka R v prípade poistenia m -tíc osôb. Práca poskytuje teoretické poznatky ako aj praktické riešenie danej problematiky. Práca sa člení na štyri kapitoly. Prvá kapitola analyzuje súčasný stav problematiky v poistnej praxi a zaoberá sa stručnou charakteristikou programovacích jazykov používaných v životnom poistení. Podrobne popisuje balík *lifecontingencies* programovacieho jazyka R ako základné rozšírenie určené pre poistno-matematické výpočty. Druhá kapitola vytyčuje ciele práce. Ďalšia kapitola je venovaná teoretickým poznatkom potrebných k praktickým aplikáciám. Štvrtá časť práce je venovaná vytvoreniu vlastných funkcií a verifikácii vlastných algoritmov na konkrétnych praktických ukážkach.

Kľúčové slová: aktuárske balíky, životné poistenie, poistenie m -tíc osôb, softvér R.

Abstract

HORVÁTHOVÁ, Szilvia: The usage of programming language R for multiple life insurance. – University of Economics in Bratislava. Faculty of Economic Informatics; Department of Mathematics and Actuarial Science – Head of the final work: Mgr. Ing. Ingrid Krčová, PhD. - Bratislava: FHI EUBA, 2022, 76 pages.

The aim of this work is to acquaint the reader with the possibilities of the usage of programming language R in the case of multiple life insurance. The work provides theoretical knowledge and practical solutions to the problem. The thesis is divided into four chapters. The first chapter analyzes the current status of problems in insurance practice and deals with a brief description of the programming languages used in life insurance. It describes in detail the lifecontingencies package of the R programming language as a basic extension for insurance mathematical calculations. The second chapter outlines the goals of the work. The next chapter is devoted to theoretical knowledge needed for practical applications. The fourth part is dedicated to the creation of own functions and verification of own algorithms on examples.

Keywords: actuarial packages, life insurance, multiple life insurance, software R.

Obsah

Úvod	10
1 Súčasný stav riešenej problematiky doma a v zahraničí	11
1.1 Základné pojmy v životnom poistení	11
1.2 Poistenie m -tíc osôb	13
1.3 Programovacie jazyky používané v poistnej praxi	15
1.4 Balík lifecontingencies programovacieho jazyka R	17
2 Ciele práce	20
3 Metodika práce a metódy skúmania	21
3.1 Základné princípy poistenia osôb	21
3.2 Aktuárska báza	22
3.3 Ukazovatele úmrtnostných tabuliek pre m -tice osôb	24
3.3.1 Funkcia prežitia a pravdepodobnosti pre spojený život	24
3.4 Komutačné čísla pre spojený život	25
3.5 Úmrtnostné tabuľky pre m -tici osôb	27
3.6 Základné poistné produkty	27
3.6.1 Poistenie spojeného života na dožitie	27
3.6.2 Poistenie dôchodkov pre spojený život	28
3.6.3 Poistenie na prvé úmrtie	29
3.6.4 Zmiešané poistenie pre spojený život	30
3.7 Ďalšie stavy v poistení m -tíc osôb	30
3.7.1 Stav do posledného úmrtia v m -tici osôb	30
3.7.2 Stav, keď práve r osôb v m -tici osôb je nažive	32
3.7.3 Stav, keď aspoň r osôb v m -tici osôb je nažive	32
3.7.4 Zložené stavy m -tice osôb	33
3.8 Jednostranné dôchodky	33
3.9 Spojité poistenia	34
3.10 Zostavenie úmrtnostných tabuliek v prostredí R	35
3.11 Výpočet súčasných hodnôt poistných produktov v prostredí R	38

3.11.1 Pravdepodobnosti existencie daného stavu	38
3.11.2 Poistenie na úmrtie	40
3.11.3 Dôchodky pre m -tice osôb	42
4 Výsledky práce a diskusia	45
4.1 Aktuárske úmrtnostné tabuľky pre dvojicu osôb pomocou vlastnej funkcie	45
4.2 Jednostranné dôchodky	47
4.3 Spojité poistenie na úmrtie pre m -tice osôb	49
4.4 Z-metóda a upravená Z-metóda	50
4.5 Poistenia so stavmi, keď práve a aspoň r osoba je nažive	51
Záver	55
Použitá literatúra	57
Prílohy	

Úvod

Poistná prax je úzko spojená s aplikovaním softvérov a programovacích jazykov. Každý aktúar vo svojej práci využíva technickú podporu formou počítačových programov. Používanie voľne dostupného štatistického softvéru R sa stáva čoraz rozšírenejším medzi aktúarmi, nakoľko je to efektívny nástroj na výpočty, na analýzy, na vizualizáciu a na vykonanie Monte Carlo simulácií.

Diplomová práca je zameraná na poskytnutie teoretických základov pri poistení m -tíc osôb a používanie programovacieho prostredia R pri výpočtoch spojených s touto problematikou. Práca je rozdelená do štyroch kapitol.

Prvá kapitola obsahuje všeobecný prehľad životného poistenia. Analyzuje súčasný stav problematiky v praxi a stručne charakterizuje programovacie jazyky používané v poisťovniach. Druhá časť kapitoly je zameraná na charakteristiku príkazov a zaoberá sa s príkazmi, ktoré sú používané na konštrukciu úmrtnostných tabuliek pre jednoduchý život. Túto kapitolu tvoria príkazy, ktoré slúžia na výpočet pravdepodobnosti existencie nejakých špecifických stavov v m -tici a výpočet súčasných hodnôt základných poistných produktov pre tieto stavy.

Druhá kapitola definuje ciele diplomovej práce, charakterizuje čiastkové ciele, ktoré sú nevyhnutné na dosiahnutie hlavného cieľa: využitie programovacieho jazyka R pre poistenia m -tíc osôb.

Tretia kapitola analyzuje problematiku z diskrétného prístupu. Skoncipujeme tu spôsob výpočtu základných druhov poistných produktov a spôsoby platenia poistného v poistení m -tíc osôb.

Posledná kapitola je praktická časť záverečnej práce. Nakoľko balík lifecontingencies len čiastočne rieši problematiku m -tíc osôb, táto kapitola sa zaoberá tvorbou vlastných funkcií. Obsahuje príklady poistenia m -tíc osôb, riešenie ilustračných príkladov a overenie funkčnosti rozšírenia balíka v prostredí štatistického softvéru R.

Práca má uviesť čitateľa do programovacieho sveta a do problematiky poistenia m -tíc osôb. Čitateľ by mal byť schopný používať jednotlivé príkazy a efektívne ich aplikovať aj na výpočet poistného pre daný poistný produkt, ako aj v oblasti testovania citlivosti vzhľadom na zmenu predpokladov v aktuárskej báze.

1 Súčasný stav riešenej problematiky doma a v zahraničí

Poisťovňa plní úlohu stabilizátora životnej úrovne v ekonomike každého vyspelého štátu. Hlavnou úlohou je zachovávanie životnej úrovne poistenca v prípade neočakávanej udalosti, ktorá má náhodný charakter. Skladá sa z dvoch odvetví, z neživotného a životného poistenia. Životné poistenie sa zaoberá rizikami ako je smrť a život, pričom tieto riziká modeluje pomocou úmrtnostných tabuliek. Pod pojmom poistenia m -tice osôb rozumieme také poistenie, pri ktorom vznik poistnej udalosti závisí od smrti alebo života aspoň dvoch osôb.

Prvá kapitola tejto diplomovej práce poskytuje základné pojmy, informácie a prehľad o problematike životného poistenia a poistenia viac osôb. Poskytuje široký prehľad o súčasných poistných produktoch, s ktorými poisťovne obchodujú na slovenskom poistnom trhu. Rozoberá a opisuje programovacie jazyky, ktoré používajú poisťovne pri svojich výpočtoch a analýzach. Ďalej analyzuje balík lifecontingencies, ktorý je voľne dostupný a tiež príkazy používané pri oceňovaní poistných produktov určené pre m -ticu osôb.

„**Životné poistenie** je zmluvný vzťah medzi poistníkom a poisťovňou, v ktorom sa poisťovňa zaväzuje vyplatiť dohodnuté poistné plnenie v prípade smrti poisteného alebo dožitia dohodnutého veku, alebo po uplynutí poistnej doby. Poistník je povinný platiť poistné v pravidelných splátkach – bežné poistné (ročne, polročne, štvrťročne, mesačne) alebo jednorazovo pri uzavretí poistnej zmluvy“¹.

1.1 Základné pojmy v životnom poistení

V životnom poistení sa najčastejšie používajú nasledujúce pojmy²:

- **Poisťovňa (poistiteľ)** – právnická osoba, ktorá vykonáva poisťovaciu činnosť na základe povolenia Národnej banky Slovenska. Podľa 39/2015 Z. z. Zákona o poisťovníctve musí mať poisťovňa právnu formu akciovej spoločnosti alebo európskej spoločnosti³.
- **Poistená osoba** – osoba, na ktorú sa poistenie vzťahuje.

¹ Zdroj: <https://openiazoch.zoznam.sk/zivotne-poistenie/co-je-to>

² POTOCKÝ, R. 2012. Modely v životnom a neživotnom poistení. Prvé vydanie, Vydavateľstvo STATIS Bratislava

³ Zdroj: <https://www.slov-lex.sk/pravne-predpisy/SK/ZZ/2015/39/>

- **Poistník** – osoba, ktorá uzavrela poistnú zmluvu. Niekedy sa poistná osoba zhoduje s poistníkom.
- **Poistná udalosť** – udalosť, ktorá má náhodný charakter. Je predmetom poistenia a pri jej vzniku je poisťovňa povinná vyplatiť poistné plnenie poistenej osobe alebo jej pozostalým.
- **Poistné plnenie** – plnenie, ktoré sa poistiteľ zaväzuje v poistnej zmluve poskytnúť v prípade vzniku poistnej udalosti.
- **Poistné riziko** – nebezpečenstvo, ktoré môže viesť k vzniku poistnej udalosti.
- **Poistné** – vopred dohodnutá suma, ktorú je poistník povinný platiť poisťovne počas poistnej doby.
- **Pripoistenie** – je rozšírenie základného poistenia o ďalšie riziká, ide napríklad o pripoistenie finančnej straty v prípade práceneschopnosti⁴.
- **Poistný kmeň** – súbor poistných zmlúv, ktorú poisťovňa spravuje. Tiež sa používa výraz portfólio poisťovacej spoločnosti⁵.

Delenie životného poistenia

V teórii ani v praxi neexistuje jednoznačné delenie životného poistenia. Autorky SEKEROVÁ a BILÍKOVÁ vo svojej publikácii uvádzajú podľa rôznych kritérií nasledujúce delenie životného poistenia⁶:

1. podľa dĺžky trvania na
 - a) doživotné,
 - b) dočasné,
2. podľa účinnosti na
 - a) okamžité,
 - b) odložené,
3. podľa počtu zahrnutých rizík na
 - a) jednoduché,
 - b) združené,
4. podľa počtu zahrnutých osôb na
 - a) individuálne,

⁴ Zdroj: <https://www.financnykompas.sk/clanok/pripoistenie>

⁵ Zdroj: <https://poistovne.sk/poistny-kmen>

⁶ SEKEROVÁ, M. – BILÍKOVÁ, M. Poistná matematika. Bratislava: Vydavateľstvo EKONÓM, 2005.

b) skupinové.

Autor POTOCKÝ vo svojej knihe uvádza oveľa stručnejšie rozdelenie životného poistenia. Používa delenie podľa účelu poistenia⁷:

1. dočasná ochrana,
2. doživotná ochrana,
3. sporenie,
4. investície.

1.2 Poistenie m -tíc osôb

Špeciálnym typom životného poistenia je poistenie m -tíc osôb. Výplata poistného plnenia v prípade takýchto poistení závisí od smrti alebo života minimálne dvoch osôb. Poistovne ponúkajú široké spektrum poistných produktov pre poistenie m -tíc osôb. Najčastejším prípadom je poistenie rodiny (v zmysle najbližšej rodiny, ako je: manžel, manželka a ich deti), ale existujú poistovne, ktoré ponúkajú poistenie s možnosťou poistenia viac osôb bez kritérií príbuzenstva.

Najčastejším poistným produktom, ktoré poistovne na Slovensku ponúkajú, je poistenie na úmrtie pre viac osôb. Výplata poistnej sumy sa uskutoční v prípade úmrtia jednej z poistených osôb. Produkt slúži aj na zabezpečenie úveru. V tomto prípade poistovne ponúkajú možnosť poistiť sa na klesajúce poistné plnenie, ktoré je v čase úmrti vo výške nesplatennej časti úveru, istiny a úroku. Poistenie je vhodné pre manželské páry, ktoré splácajú spoločne hypotekárny úver. **Allianz – Slovenská poistovňa, a. s.** vo svojom poistnom produkte „Šťastný život“⁸ ponúka podobné poistenie s pevnou alebo klesajúcou poistnou sumou. Poistný produkt je možné rozšíriť pripoisteniami, ako je napríklad poistenie trvalých následkov úrazu. Poistovňa Allianz má tiež vo svojej ponuke zmiešané poistenie „Flexi Život“⁹, ktoré zabezpečí výplatu poistnej sumy (poistného plnenia) aj v prípade, ak všetky poistené osoby sa dožijú doby stanovenej v poistnej zmluve, teda poistnej doby.

⁷ POTOCKÝ, R. 2012. Modely v životnom a neživotnom poistení. Prvé vydanie, Vydavateľstvo STATIS Bratislava

⁸ Zdroj: https://www.allianzsp.sk/sk_SK/sukromne-osoby/zivot/stastny-zivot.html#

⁹ Zdroj: https://www.allianzsp.sk/sk_SK/sukromne-osoby/zivot/flexi-zivot.html#

Veľmi podobný poistný produkt má v ponuke aj **ČSOB Poist'ovňa, a. s.** s názvom „Vital“¹⁰, ktorý ponúka možnosť poistenia pre rodiny alebo skupiny do deväť osôb.

Generali Poist'ovňa, a. s. má vo svojom portfóliu poistných produktov poistenie pre prípad smrti v prospech dieťaťa, to znamená, že výplata poistnej sumy (plnenia) sa uskutoční v prípade smrti rodiča, ak smrť nastala počas poistnej doby¹¹.

Poist'ovňa **MetLife Europe Insurance d. a. c.** ponúka vo svojom poistnom produkte „TrendLife“¹² poistenie na úmrtie s možnosťou spolupoistenia troch dospelých a maximálne desiatich detí.

NN Životná poist'ovňa, a. s. v poistnom produkte „NN Partner“¹³ umožní poistiť až sedem osôb, nakoľko poistené osoby nemusia byť v príbuzenskom vzťahu. Ponúka veľký rozsah poistného krytia vo výbere doplnkových pripoistení pre každého poisteného a doplnkové pripoistenia je možné v čase trvania poistného vzťahu meniť.

Stále väčší a väčší medzi poistníkmi je dopyt po zabezpečení rodiny formou jednostranných dôchodkov. Ide o poistenie, ktoré po úmrtí poistenej osoby umožní vyplácať pravidelné dávky v rovnakých časových intervaloch pre osoby uvedené v poistnej zmluve. Tieto typy dôchodkov sa používajú najmä pri zabezpečení úverov, kde dávky sú vo výške splátky úveru. Poist'ovne tiež ponúkajú podobné poistné produkty s možnosťou pripoistenia, ako je napríklad zabezpečenie príjmu, teda ak sa poistená osoba stane invalidom, tak v pravidelných dávkach dostane určité percento z dohodnutej poistnej sumy. Podobný jednostranný dôchodok má v ponuke **Allianz – Slovenská poist'ovňa, a. s.** s názvom „Moja Farbička“¹⁴. Je to komplexná ochrana dieťaťa, ktorá spočíva vo finančnom zabezpečení dieťaťa v prípade neočakávanej smrti jedného alebo oboch rodičov. K tomuto poistnému produktu poist'ovne ponúkajú aj finančný príspevok, ktorý v prípade, že poistené dieťa utrpí úraz, rodičia dostanú určité percento z ročného poistného. V prípade úmrtia rodičov bude dieťaťu vyplácaný vopred dohodnutý dôchodok až do roku, v ktorom dovŕši vek 25 rokov.

Ďalší typ poistných produktov pri poistení *m* -tíc osôb je určený pre firmy a slúži na zabezpečenie zamestnancov a ich blízkych v prípade smrti alebo úrazov a ich následkov.

¹⁰ Zdroj: <https://www.csob.sk/zivotne-poistenie/vital>

¹¹ Zdroj: <https://www.generali.sk/pre-obcanov/zivotne-poistenie/detske-poistenie/>

¹² Zdroj: <https://www.metlife.sk/individualne-poistenie/zivotne-poistenie/trendlife/>

¹³ Zdroj: <https://poistovna.nn.sk/nn-partner/#tab1>

¹⁴ Zdroj: https://www.allianzsp.sk/sk_SK/sukromne-osoby/zivot/moja-farbicka.html#

Poist'ovňa **MetLife Europe Insurance d.a.c** ponúka poistenie, ktoré je určené ako doplnok k zákonnému poisteniu zamestnávateľa za škodu pri pracovnom úraze a chorobe z povolania. Poist'ovňa ponúka výrazne nižšie poistné za tento typ poistenia ako u individuálneho poistenia a platnosť poistenia na celom svete. Poistné plnenie pre takéto poistenia je stanovené jednotne pre všetky poistené osoby, alebo poistník môže rozdeliť poistených do viacerých kategórií s odlišnými poistnými plneniami, a to podľa pozície zamestnanca v spoločnosti¹⁵.

1.3 Programovacie jazyky používané v poistnej praxi

Programovacie jazyky sú neoddeliteľnou súčasťou podnikateľských a iných subjektov, v ktorých sú prevádzané matematické výpočty, analýzy, ale aj v subjektov, ktorých činnosť je závislá od dobre naprogramovaného softvéru. Nie je tomu ani inak v poisťovníctve, nakoľko poisťovne pracujú s obrovským poistným kmeňom. Najčastejšie používané programovacie jazyky, ktoré používajú poisťovne, sú:

1. Prophet

Prophet je jednou zo špičkových poistno-matematických systémov modelovania, ktorý pomáha poisťovacím a finančným spoločnostiam plniť si svoje vykazovacie povinnosti, zlepšovať riadenie rizík a rýchlejšie zavádzať nové poistné produkty na trh. Softvér Prophet používa poistno-matematické knižnice pre všetky hlavné typy produktov. Prophet Professional je integrovaná sada poistno-matematických systémov, ktoré používajú predovšetkým poisťovacie spoločnosti na vykonávanie poistno-matematických výpočtov vrátane testovania zisku, projekcií, rezervácií, modelovania pasív, stochastického modelovania a riadenia rizík¹⁶.

2. SQL

Medzi často používané a zároveň najľahšie zvládnuteľné programovacie jazyky patrí dotazovací jazyk SQL. Jazyk SQL alebo „structured query language“ je štandardný jazyk pre prístup k relačným databázam. Aj keď mnohé poisťovne zaviedli vlastné programovacie jazyky, väčšina z nich sa drží štandardu SQL. Hlavným dôvodom

¹⁵ Zdroj: <https://www.metlife.sk/skupinove-poistenie/skupinove-zivotne-alebo-urazove-poistenie/>

¹⁶ Zdroj: <https://www.prophet-web.com/about-us/>

využívateľnosti tohto programovacieho jazyka je jednoduchosť a ľahšie spravovania databázy MySQL, Oracle ¹⁷.

3. Visual Basic for Application

Visual Basic for Application (ďalej len „VBA“) je vlastne programovací jazyk programov z balíčka Microsoft Office. VBA po inštalácii balíka Microsoft Office je automaticky dostupný bez nutnosti dodatočnej inštalácie. Syntaxi jazyka sú pre všetky aplikácie, či už Excel, Word alebo Access rovnaké. Líšia sa iba objektovým modelom, s ktorým pracujeme. Jazyk je určený predovšetkým na prispôsobenie a na tvorbu vlastných makier v Microsoft Office¹⁸.

4. JAVA

Programovací jazyk Java je druhým najpopulárnejším jazykom na svete. V Jave sa dajú programovať desktopové aj serverové aplikácie, vytvárať webové stránky, dá sa používať na prístup k dátam a umožňuje realizovať zložité výpočty. Java odvodzuje svoju syntax z jazyka C, teda keď niekto ovláda jazyk Java, ľahšie sa naučí programovať v jazyku C# a C++. Jazyk Java je objektovo orientovaný jazyk, existuje v ňom množstvo rôznych knižníc. Veľkou nevýhodou programovacieho jazyka je dlhá štruktúra príkazov¹⁹.

5. Jazyk C

Programovací jazyk C je jedným z najpoužívanějších programovacích jazykov. Je vhodný na písanie systémového softvéru, hoci sa používa aj na tvorbu aplikačného softvéru. Jazyk C nepatrí medzi jazyky vyznačujúce sa dobrou čitateľnosťou. Je zameraný na systémové programovanie a má pomerne málo jednoduchých pravidiel, pomocou ktorých je možné vytvárať a skladať jednotlivé úseky programov. Základnou programovou jednotkou v jazyku C je funkcia. Veľké množstvo štandardných funkcií je už pripravených v knižniciach²⁰.

6. Python

Python je interaktívny, voľne dostupný programovací jazyk. Namiesto toho, aby nútil programátora používať určitý štýl programovania, umožňuje používanie viacerých

¹⁷ Zdroj: <https://ivit.sk/blog/styri-hlavne-dovody-preco-sa-naucit-jazyk-sql/>

¹⁸ KRÁL, M. Excel VBA. Brno: Vydavatel'stvo Computer Press, 2012. 504 s.

¹⁹ Zdroj: <https://www.learn2code.sk/blog/studijny-plan-pre-buduceho-java-programatora>

²⁰ Zdroj: [https://sk.wikipedia.org/wiki/C_\(programovac%C3%AD_jazyk\)](https://sk.wikipedia.org/wiki/C_(programovac%C3%AD_jazyk))

programovacích jazykov. Dôležitou vlastnosťou Pythonu je, že sa dá jednoducho rozširovať novými modulmi napísanými v jazykoch C alebo C++²¹.

7. SAS

Jazyk SAS ("Statistical Analysis System") je počítačový programovací jazyk používaný na štatistickú analýzu. Dokáže čítať údaje z bežných tabuliek (napr. z excelovských tabuliek) a databáz a podľa nich vytvárať grafy použité na štatistické analýzy²².

8. Štatistický softvér R

Open source (voľne dostupný) softvér R je určený hlavne pre štatistické analýzy dát a pre grafickú vizualizáciu. Programovací jazyk R vytvorili Ross Ihaka a Robert Gentleman v roku 1991, ktorí vtedy pôsobili na Auckland Univerzity na Katedre štatistiky. Zo svojich začiatkových písmen mien dostal programovací jazyk názov R. Napriek tomu, že ide o implementáciu jazyka S (programovací jazyk, ktorý bol používaný na štatistické výpočty), sa programovací jazyk R stal praktickým štandardom v mnohých oblastiach štatistiky.

Súčasťou inštalácie R je sada základných balíkov a ďalších viac než 16 000 sád je voľne k dispozícii. Funkcie prostredia R je možné rozšíriť pomocou knižníc a je tiež možnosť vytvoriť si vlastné knižnice s vlastnými funkciami²³.

1.4 Balík lifecontingencies programovacieho jazyka R

Pre potreby poisťnej praxe sa používa knižnica `lifecontingencies`. Tento balík nie je súčasťou základných balíkov softvéru R, ale okrem databáz týkajúcich sa životného poistenia obsahuje aj rôzne funkcie z oblasti finančnej matematiky. Balík je najdôležitejšia knižnica oblasti životného poistenia, ktorý obsahuje príkazy na konštrukciu úmrtnostných tabuliek, príkazy na výpočet súčasných hodnôt poisťných produktov, ako aj príkazy slúžiace na výpočet pravdepodobnosti prežitia a úmrtia. Čiastočne sa ním rieši aj problematika poistenia m -tíc osôb²⁴.

²¹ Zdroj: [https://sk.wikipedia.org/wiki/Python_\(programovac%C3%AD_jazyk\)](https://sk.wikipedia.org/wiki/Python_(programovac%C3%AD_jazyk))

²² Zdroj: https://en.wikipedia.org/wiki/SAS_language

²³ Zdroj: [https://sk.wikipedia.org/wiki/R_\(programovac%C3%AD_jazyk\)](https://sk.wikipedia.org/wiki/R_(programovac%C3%AD_jazyk))

²⁴ SPEDICATO, G. A., KAINHOFER, R., OWENS, J. K., DUTANG CHRISTOPHE, D., SCHIRMACHER, E., CLEMENTE G. P., WILLIAMS I. 2021. Package 'lifecontingencies'

Tab 1: Prehľadná tabuľka poistno-matematických funkcií v balíku lifecontingencies

1 osoba		m -tice osôb	
príkaz	produkt	príkaz	produkt
axn	dôchodok	$axyn$	dôchodok pre dvojice osôb
$Iaxn$	lineárne rastúci dôchodok		
Axn	poistenie na úmrtie	$Axyn$	poistenie na úmrtie pre dvojice osôb
$DAxn$	poistenie na úmrtie klesajúce lineárne		
$IAxn$	poistenie na úmrtie rastúce lineárne	$axyzn$	dôchodok pre m -tice osôb
Exn	poistenie na dožitie	$Axyzn$	poistenie na úmrtie pre m -tice osôb
$AExn$	zmiešané poistenie		

Zdroj: vlastné spracovanie

Pri prvom používaní daného balíka je nevyhnutné najprv nainštalovať balík pomocou príkazu `install.packages("lifecontingencies")` a pri použití vždy je potrebné načítať balík do softvéru pomocou príkazu `library(lifecontingencies)`²⁵.

Okrem finančných a poistno- matematických funkcií obsahuje databázy rôznych krajín zamerané na ukazovatele úmrtnostných tabuliek. Existujú štyri typy takých databáz²⁶:

1. Databázy s ukazovateľom pravdepodobnosti úmrtia – patria sem databázy:

- `demoCanada` (dáta z roku 2015 a 2020 pre obidve pohlavia zvlášť),
- `demoChina` (obsahuje ročné pravdepodobnosti úmrtia),
- `demoGermany` (dáta z roku 1994 pre obidve pohlavia zvlášť),
- `demoJapan` (dáta z roku 2012),

²⁵ R Core Team (2020). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>

²⁶ SPEDICATO, G. A., KAINHOFER, R., OWENS, J. K., DUTANG CHRISTOPHE, D., SCHIRMACHER, E., CLEMENTE G. P., WILLIAMS I. 2021. Package 'lifecontingencies'

- demoUk (dáta z roku 1992).
2. Databázy s ukazovateľom l_x - patria sem databázy:
 - demoFrance (dáta z rokov 1988 – 1990 a 2000-2002),
 - demoIta (dáta z rokov 1931, 1951, 1961, 1971, 1981, 1992, 2000 a 2002 pre obidve pohlavia zvlášť),
 - demoUsa (dáta z rokov 1990, 2000 a 2007 pre obidve pohlavia zvlášť),
 - soaLt (dáta, ktoré sú aplikované na riešenie ilustračných príkladov v knihe: Actuarial Mathematics).
 3. Databáza s úmrtnostnou tabuľkou – obsahuje kompletnú úmrtnostnú tabuľku, patrí sem databáza soa08 (úmrtnostná tabuľka použitá na riešenie ilustračných príkladov v knihe: Actuarial Mathematics).
 4. Databáza s komutačnými číslami – patrí sem databáza soa08Act (tabuľka konštruovaná z databázy soa08 pri úrokovej miere 6 %).

2 Ciele práce

Cieľom diplomovej práce je poskytnúť efektívne možnosti oceňovania poistných produktov v prípade poistenia viacerých osôb pomocou štatistického softvéru R.

Na splnenie hlavného cieľa budeme musieť spracovať čiastkové ciele. Ako prvý čiastkový cieľ je nutné zmapovať jednotlivé príkazy už zabudované v open source softvéri R. Vzhľadom na to, že softvér R obsahuje funkcie pre oceňovanie poistných produktov len pre spojený život a pre stav do posledného úmrtia, naším cieľom je naprogramovanie určitých funkcií v prostredí R na umožnenie výpočtov aj pre ďalšie stavy v m -tici osôb.

V praxi je konštrukcia úmrtnostných tabuliek pre viaceré osoby veľmi zdĺhavá a kvôli veľkému množstvu kombinácií by vznikli neprehľadné tabuľky. Preto sa v minulosti vyvíjali rôzne aproximácie pri zostavení úmrtnostných tabuliek, aby boli ľahšie použiteľné a prehľadné. Aplikovanie softvéru nám umožní získať presnejšie a rýchlejšie výpočty pri oceňovaní poistných produktov pre m -tice osôb. Vzhľadom na to, že v jazyku R je zabudovaná konštrukcia úmrtnostných tabuliek len pre jednu osobu, jedným z našich čiastkových cieľov je vytvoriť vhodný zdrojový kód na konštrukciu úmrtnostných tabuliek pre dve osoby.

Posledným čiastkovým cieľom je výpočet ilustračných príkladov v oblasti poistenia m -tíc osôb, a to aplikovaním príkazov a vlastných funkcií v prostredí jazyka R.

3 Metodika práce a metódy skúmania

V tretej kapitole tejto diplomovej práce zadefinujeme základné princípy a predpoklady v životnom poistení, ukazovatele úmrtnostných tabuliek a základné poistné produkty v prípade poistenia m -tíc osôb.

Aktuári sa často v praxi spoliehajú na rôzne softvérové produkty, pretože sa často dostávajú k výpočtom, ktoré sú náročné na numerické spracovanie. V ďalšej časti tejto diplomovej práce sa zameriame na používanie aktuárskych balíkov, ktoré riešia problematiku poistenia m -tíc osôb. Najprv sa budeme zaoberať poistením a pravdepodobnosťami pre stav spojený život a stav do posledného úmrtia, potom sa postupne dostaneme k stavom, keď je v m -tici osôb práve r osôb nažive a aspoň r osôb nažive. Pri výpočtoch a pri algoritmoch budeme aplikovať tabuľkový počet žijúcich z roku 2019. Dáta potrebné k výpočtom sú získané z webovej stránky Štatistického úradu Slovenskej republiky²⁷.

3.1 Základné princípy poistenia osôb

Skôr, ako prejdeme k zadefinovaniu jednotlivých pravdepodobností, ukazovateľov a poistení pre m -tícu osôb, zavedieme si základné princípy poistenia osôb. Uvedieme ich pre jednoduchý život, t. j. pre jednu osobu dnes vo veku x .

Pri kalkulácii netto poistného sa používajú dva princípy²⁸, a to princíp fiktívneho súboru a princíp ekvivalencie. Pre prvý z nich je potrebné uviesť jeden z ukazovateľov pre úmrtnostnú tabuľku pre jednoduchý život. Nech teda l_x vyjadruje tabuľkový počet dožívajúcich sa veku x , pričom platí, že $x \in \{0, 1, \dots, \omega\}$ a ω vyjadruje predpokladaný najvyšší vek v tabuľke, ktorý môže dosiahnuť osoba sledovaného súboru. Špeciálne sa začiatočná hodnota l_0 nazýva koreň úmrtnostnej tabuľky a najčastejšie je to 100 000 osôb a tiež platí, že $l_{\omega+1} = 0$. Teraz tieto dva princípy uvedieme, pretože na základe týchto princíпов oceníme aj produkty životného poistenia pre jednotlivé stavy v poistení m -tice osôb.

²⁷ Štatistický úrad Slovenskej republiky. Úmrtnostné tabuľky (roky 1996 - 2019): Slovenská republika. Obyvateľstvo a migrácia: Tabuľky života.

²⁸ SEKEROVÁ, M. – BILÍKOVÁ, M. Poistná matematika, s.127. Bratislava: Vydavateľstvo EKONÓM, 2005

Princíp fiktívneho súboru spočíva v predpoklade, že počet uzatvárajúcich nových poistných zmlúv toho istého poistného produktu je l_x z úmrtnostnej tabuľky (teda poistnú zmluvu uzatvárajú všetky osoby, ktoré sú nažive vo veku x) pričom sa predpokladá, že všetky osoby modelového súboru sa narodili 1.1 a zomreli 31.12.

Princíp ekvivalencie znamená, že pri uzatváraní novej poistnej zmluvy toho istého typu poistného produktu sú príjmy a výdavky poisťovne v rovnováhe, keď sú diskontované k rovnakej časovej základni.

3.2 Aktuárska báza

Nemenej dôležité je uviesť pojem aktuárskej bázy. Aktuársku bázu tvoria výpočtové predpoklady, ktoré významne ovplyvnia všetky výpočty súvisiace s poistením, a preto je aktuárska báza pre poistnú prax veľmi dôležitá. Medzi základné prvky aktuárskej bázy patria: úroková miera, úmrtnosť, inflácia, náklady poisťovne, odkupy (odkupné hodnoty). Jednotlivé predpoklady veľmi stručne popíšeme, hoci nie všetky budeme v praktickej časti práce používať.

Úroková miera

Úroková miera je najdôležitejší predpoklad poistnej bázy. Predstavuje priemernú návratnosť všetkých investícií poisťovne. Určenie výšky úrokovej miery má najväčší vplyv na zisk poisťovne. Ak poisťovňa predpokladá optimistický vývoj úrokovej miery a napriek tomu v budúcnosti reálna návratnosť bude nižšia ako predpokladaná, tak aj zisk poisťovne bude nižší ako predpokladaný. Preto poisťovne používajú pesimistický predpoklad pri určovaní výšky úrokovej miery. Existujú rôzne faktory, ktoré ovplyvňujú výber úrokovej miery pri oceňovaní produktov životného poistenia ako napríklad druh poistného produktu (poistnej zmluvy), citlivosť tohto produktu na úrokovú mieru, doba trvania poistenia, spôsob platenia poistného a miera reinvestícií v budúcnosti²⁹.

Úmrtnosť

Úmrtnosť udáva podiel zomrelých z určitej skupiny za jeden rok³⁰. Význam predpokladu o mortalite (úmrtnosti) závisí predovšetkým od druhu poistného produktu, a to najmä pre dočasné poistenie na úmrtie. Miera úmrtnosti, ktorú poisťovňa používa, by

²⁹ SAKÁLOVÁ, K. Oceňovanie produktov v životnom poistení s. 46. Bratislava: Vydavateľstvo EKONÓM, 2001

³⁰ <http://www.hpi.sk/2009/08/mortalita/>

mala reprezentovať očakávanú úmrtnosť tých poistencov, ktorí uzavrú daný poistný produkt. Poistovňa používa úmrtnostné tabuľky celej populácie alebo úmrtnostné tabuľky týkajúce sa príslušného poistného produktu. Veľké poisťovne z dlhoročných skúseností používajú vlastné selekčné tabuľky na určenie mortality pre daný poistný produkt.

Náklady

Poisťovňa ako každý podnikateľský subjekt vytvára náklady počas svojej činnosti. Klient okrem toho, že platí určitú sumu aby kryl svoje riziko, podieľa sa aj na splatení nákladov prostredníctvom platenia brutto poistného. Na určenie nákladov poisťovňa analyzuje svoje náklady v určitom období pre istý druh poistného produktu. Podľa toho, v ktorom časovom období vzniknú počas trvania poistenia, členíme náklady na:

- začiatkové náklady – patria sem: provízia, marketingové náklady, administratívne náklady,
- priebežné (správne) náklady – tieto náklady slúžia na udržiavanie a prevádzku poistenia,
- inkasné náklady – náklady na inkasovanie poistného (v prípade, ak sa poistné platí bežne),
- záverečné (terminálne) náklady – náklady súvisia s výplatou poistného plnenia.

Výška nákladov môže byť určená ako časť alebo percento poistného, alebo poistného plnenia.

Inflácia

Inflácia znamená všeobecný nárast cien ekonomických statkov, čo má za následok zníženie kúpnej sily peňazí³¹. Rastúca inflácia zvyšuje náklady poisťovne, čo má vplyv aj na výšku poistného. Spôsobuje aj znižovanie reálnej hodnoty poistného plnenia. Poisťovňa infláciu zahrňuje do ocenení produktov buď predpokladom pesimistickej úrokovej miery alebo explicitným spôsobom, t. j. prostredníctvom modifikovanej úrokovej miery, ktorý sa určí podľa vzorca

$$i' = \frac{i - f}{1 + f}$$

³¹ <https://www.ecb.europa.eu/ecb/educational/hicp/html/index.sk.html>

Odkup poistenia

Odkup poistenia znamená predčasné ukončenie alebo stornovanie kontraktu, pri ktorom je vyplatená odkupná hodnota. Je nepredvídateľným faktorom, lebo o ukončení rozhoduje poistenec.

V ostatných častiach tejto diplomovej práce sa nebudeme zaoberať s nákladmi a infláciou a nebudeme rátať s možnosťou odkupu poistenia.

3.3 Ukazovatele úmrtnostných tabuliek pre m -tice osôb

Poistné riziko v životnom poistení sa posudzuje na základe úmrtnostných tabuliek. Podľa sledovaného vekového intervalu poznáme skrátené (s vekovým intervalom 5 rokov) a úplne úmrtnostné tabuľky (s vekovým intervalom 1 rok). Úmrtnostné tabuľky vydáva Štatistický úrad Slovenskej republiky každoročne, zvlášť pre ženy a mužov ako aj spoločne pre obidve pohlavia. Poistovne používajú vlastnú úmrtnostnú tabuľku, zostavenú na základe dlhoročných skúseností.

V ďalších častiach tejto kapitoly budeme predpokladať, že jednotlivé osoby m -tíc osôb majú veku $x_1, x_2, \dots, x_m$. Životy týchto osôb sú nezávislé a všetky osoby sa riadia tým istým úmrtnostným zákonom, teda „používajú“ tú istú úmrtnostnú tabuľku.

3.3.1 Funkcia prežitia a pravdepodobnosti pre spojený život³²

Funkcia prežitia pre spojený život reprezentuje počet neporušených m -tíc osôb, teda všetky osoby z m -tice sú nažive, označí sa symbolom $l_{x_1, x_2, \dots, x_m}$. Z predpokladu nezávislého správania veku m -tíc osôb vyplýva tento vzťah:

$$l_{x_1, x_2, \dots, x_m} = l_{x_1} \cdot l_{x_2} \cdot \dots \cdot l_{x_m} \quad (1)$$

Počet m -tíc porušených úmrtím v priebehu jedného roka definujeme vzťahom:

$$d_{x_1, x_2, \dots, x_m} = l_{x_1, x_2, \dots, x_m} - l_{x_1+1, x_2+1, \dots, x_m+1} \quad (2)$$

Pravdepodobnosť prežitia pre spojený život reprezentuje pravdepodobnosť toho, že všetky osoby v skupine prežijú nasledujúcich n rokov. Označí sa symbolom ${}_n p_{x_1, x_2, \dots, x_m}$ a podobne ako pre funkciu prežitia pre spojený život platí vzťah:

³² SEKEROVÁ, M. – BILÍKOVÁ, M. Poistná matematika, s.127. Bratislava: Vydavateľstvo EKONÓM, 2005

$${}_n P_{x_1, x_2, \dots, x_m} = {}_n P_{x_1} \cdot {}_n P_{x_2} \cdot \dots \cdot {}_n P_{x_m} \quad (3)$$

kde ${}_n p_{x_i}$, pre $i = 1, 2, \dots, m$ sú pravdepodobnosti pre jednoduchý život, čiže pravdepodobnosť, že osoba vo veku x_i prežije nasledujúci rok.

Pravdepodobnosť porušenia spojeného života úmrtím vyjadruje pravdepodobnosť, že v m -tici aspoň jedna osoba zomrela v priebehu n rokov. Označí sa symbolom ${}_n q_{x_1, x_2, \dots, x_m}$ a platí vzťah:

$${}_n q_{x_1, x_2, \dots, x_m} = 1 - {}_n P_{x_1, x_2, \dots, x_m} \quad (4)$$

Pravdepodobnosť, že spojený život bude porušený úmrtím práve v priebehu $(n+1)$ -vého roka sa označuje symbolom ${}_n | q_{x_1, x_2, \dots, x_m}$ a platí tento vzťah:

$${}_n | q_{x_1, x_2, \dots, x_m} = {}_n P_{x_1, x_2, \dots, x_m} - {}_{n+1} P_{x_1, x_2, \dots, x_m} = {}_n P_{x_1, x_2, \dots, x_m} \cdot q_{x_1+n, x_2+n, \dots, x_m+n} \quad (5)$$

3.4 Komutačné čísla pre spojený život

V životnom poistení sa často opakujú niektoré súčty a súčiny, a práve preto sa zaviedli komutačné čísla. Tieto matematické funkcie uľahčia a urýchľujú určenie súčasných hodnôt poistných produktov. Vo všeobecnosti pre m -tice osôb rozlišujeme komutačné čísla počtu neporušených m -tíc a komutačné čísla počtu porušených m -tíc³³.

Komutačné čísla počtu neporušených m -tíc

$$D_{x_1, x_2, \dots, x_m} = l_{x_1, x_2, \dots, x_m} \cdot v^{\frac{x_1 + x_2 + \dots + x_m}{m}} \quad (6)$$

predstavuje diskontovaný počet neporušených m -tíc osôb diskontované v priemere k dátumu ich narodenia, kde $v = (1+i)^{-1}$ a i je úroková sadzba, ktorou poisťovňa je schopná ďalej investovať do určitých investičných aktív. Pre ďalšie počty budeme potrebovať aj komutačné čísla $N_{x_1, x_2, \dots, x_m}$ a $S_{x_1, x_2, \dots, x_m}$, ktoré dostaneme postupnou sumarizáciou³⁴:

³³ TOKÁROVÁ, D. 2012. Tabuľky komutačných čísel pre dvojice a trojice osôb a ich použitie. Bakalárska práca. Fakulta hospodárskej informatiky, Ekonomická univerzita v Bratislave.

TOKÁROVÁ, D. 2012. Tabuľky komutačných čísel pre dvojice a trojice osôb a ich použitie. Bakalárska práca. Fakulta hospodárskej informatiky, Ekonomická univerzita v Bratislave.

$$N_{x_1, x_2, \dots, x_m} = \sum_{t=0}^{\infty} D_{x_1+t, x_2+t, \dots, x_m+t} \quad (7)$$

$$S_{x_1, x_2, \dots, x_m} = \sum_{t=0}^{\infty} N_{x_1+t, x_2+t, \dots, x_m+t} \quad (8)$$

Komutačné čísla počtu porušených m -tíc

$$C_{x_1, x_2, \dots, x_m} = d_{x_1, x_2, \dots, x_m} \cdot v^{\frac{x_1+x_2+\dots+x_m}{m}+1} \quad (9)$$

Aj v tomto prípade postupnou sumarizáciou dostaneme ďalšie komutačné čísla počtu prerušených m -tíc, a to

$$M_{x_1, x_2, \dots, x_m} = \sum_{t=0}^{\infty} C_{x_1+t, x_2+t, \dots, x_m+t} \quad (10)$$

$$R_{x_1, x_2, \dots, x_m} = \sum_{t=0}^{\infty} M_{x_1+t, x_2+t, \dots, x_m+t} \quad (11)$$

Špeciálny prípad poistenia m -tíc osôb je poistenie dvojice. Pri takýchto poisteniach boli definované rôzne typy komutačných čísel:

1. Dawisove komutačné čísla - umožňujú používať jednodekrementné úmrtnostné tabuľky, napríklad pri poistení x -ročnej a y -ročnej osoby komutačné číslo sa definuje nasledovne:

$${}^x D_{xy} = D_x \cdot l_y$$

2. De Morganova definícia – využíva všeobecnú definíciu komutačných čísel pre m -tícu osôb, teda berie do úvahy priemerný vek dvojicu osôb:

$$D_{xy} = l_{xy} \cdot v^{\frac{x+y}{2}}$$

$$C_{xy} = d_{xy} \cdot v^{\frac{x+y}{2}+1}$$

3. Definícia, ktorá využíva maximálny vek dvojice osôb³⁵:

$$D_{xy} = l_{xy} \cdot v^{\max\{x, y\}}$$

³⁵ TOKÁROVÁ, D. 2012. Tabuľky komutačných čísel pre dvojice a trojice osôb a ich použitie. Bakalárska práca. Fakulta hospodárskej informatiky, Ekonomická univerzita v Bratislave.

3.5 Úmrtnostné tabuľky pre m -tícu osôb

Konstruktúra úmrtnostných tabuliek pre m -tícu osôb je komplikovanejšia ako pri poistení jednej osoby. Ak by sme zostrojili úmrtnostné tabuľky pre dvojicu osôb alebo pre trojicu osôb, dostali by sme zložité neprehľadné tabuľky s obrovskými množstvami riadkov. Preto sa v praxi zaviedli viaceré spôsoby aproximácie úmrtnostných tabuliek na jednodekrementné.

„V praxi sa väčšinou postupuje takto:

- Používajú sa Davisove komutačné čísla, ktoré umožňujú používať jednodnotové úmrtnostné tabuľky.
- Používajú sa tzv. odstupňované tabuľky, napr. po 5 rokoch, t. j. vekový rozdiel medzi osobami x a y je práve 5 rokov.
- Ku každej dvojici (x, y) sa nájde (vždy za podmienky splnenia istých predpokladov o úmrtnosti) taká hodnota $\bar{x}$, aby platilo $p_{xy} \doteq p_{x\bar{x}}$.

Potom už úmrtnostné tabuľky a tabuľky komutačných čísel majú toľko riadkov ako zodpovedajúce tabuľky pre jednu osobu“³⁶.

Aby sme mohli pokračovať v charakteristike špeciálnych prípadov poistenia m -tíc osôb, je potrebné určiť základné poistné produkty, ich podstatu a vzorce pomocou ktorých určíme súčasné hodnoty.

3.6 Základné poistné produkty

Skôr, ako sa budeme venovať poisteniu „konkrétnych“ osôb v tejto podkapitole, definujeme a charakterizujeme základné životné poistné produkty pre m -tícu osôb pre rôzne stavy, ako sú: poistenie spojeného života a poistenie do posledného úmrtia.

3.6.1 Poistenie spojeného života na dožitie

Tento stav je charakteristický tým, že skupina osôb sa poistí tak, že bude im vyplatená 1 peňažná jednotka v prípade, ak sa všetky osoby v skupine dožijú n rokov. Označuje sa analogicky ako v prípade poistení jednej osoby, t. j. ${}_nE_{x_1, x_2, \dots, x_m}$. Prostredníctvom základných princípov uvedených v 3.1 by sme zostavili rovnicu

³⁶ SEKEROVÁ, M. – BILÍKOVÁ, M. Poistná matematika, s.127. Bratislava: Vydavateľstvo EKONÓM, 2005.

ekvivalencie a z nej by sme odvodili súčasnú hodnotu tohto poistenia. Súčasnú hodnotu nakoniec dostávame ako podiel dvoch komutačných čísel

$${}_nE_{x_1, x_2, \dots, x_m} = \frac{D_{x_1+n, x_2+n, \dots, x_m+n}}{D_{x_1, x_2, \dots, x_m}} \quad (12)$$

alebo môžeme toto poistenie vyjadriť pomocou pravdepodobnosti prežitia pre spojitý život

$${}_nE_{x_1, x_2, \dots, x_m} = {}_nP_{x_1+n, x_2+n, \dots, x_m+n} \cdot v^n \quad (13)$$

3.6.2 Poistenie dôchodkov pre spojený život

V tejto časti práce zadefinujeme predlehotné resp. polehotné dôchodky pre spojený život.

- **Predlehotný dôchodok** – skupina m osôb sa poistí tak, že na začiatku každého roka dostanú 1 peňažnú jednotku, pokiaľ budú všetky osoby v skupine nažive, teda až do prvého úmrtia v m -tici. Takéto dôchodky označujeme $\ddot{a}_{x_1, x_2, \dots, x_m}$ a vypočítame ich pomocou vzťahu

$$\ddot{a}_{x_1, x_2, \dots, x_m} = \frac{N_{x_1, x_2, \dots, x_m}}{D_{x_1, x_2, \dots, x_m}} \quad (14)$$

alebo pomocou pravdepodobnosti prežitia pre spojený život:

$$\ddot{a}_{x_1, x_2, \dots, x_m} = \sum_{t=0}^{\infty} v^t \cdot {}_tP_{x_1, x_2, \dots, x_m} \quad (15)$$

V prípade predlehotného odloženého dočasného dôchodku sa skupina osôb poistí tak, že dávky sa začnú vyplácať po uplynutí k rokov, ak sú všetky poistené osoby nažive, a výplata sa skončí uplynutím n rokov alebo prvým úmrtím v skupine. Súčasnú hodnotu takého poistenia označujeme ${}_k|\ddot{a}_{x_1, x_2, \dots, x_m:\overline{n}|}$, vypočítame ju podľa vzťahu

$${}_k|\ddot{a}_{x_1, x_2, \dots, x_m:\overline{n}|} = \frac{N_{x_1+k, x_2+k, \dots, x_m+k} - N_{x_1+k+n, x_2+k+n, \dots, x_m+k+n}}{D_{x_1, x_2, \dots, x_m}} \quad (16)$$

- **Polehotný dôchodok** – v tomto prípade sa m -tica osôb poistí tak, aby na konci každého roka im bola vyplatená 1 peňažná jednotka, pokiaľ budú všetky poistené osoby nažive. Označujeme $a_{x_1, x_2, \dots, x_m}$ a vypočítame podľa vzorca

$$a_{x_1, x_2, \dots, x_m} = \frac{N_{x_1+1, x_2+1, \dots, x_m+1}}{D_{x_1, x_2, \dots, x_m}} \quad (17)$$

alebo pomocou pravdepodobnosti prežitia pre spojený život:

$$a_{x_1, x_2, \dots, x_m} = \sum_{t=1}^{\infty} v^t \cdot {}_t p_{x_1, x_2, \dots, x_m} \quad (18)$$

V prípade odloženého dočasného polehotného dôchodku sa m -tice osôb poistí tak, že na konci každého roka medzi vekmi $x_1 + (k+1)$ a $x_1 + (k+1+n), \dots, x_m + (k+1)$ a $x_m + (k+1+n)$ dostanú 1 peňažnú jednotku v prípade, že všetky osoby v skupine sú nažive. Označuje sa ${}_k | a_{x_1, x_2, \dots, x_m; \overline{n}}$ a vypočítame podľa vzťahu

$${}_k | a_{x_1, x_2, \dots, x_m; \overline{n}} = \frac{N_{x_1+k+1, x_2+k+1, \dots, x_m+k+1} - N_{x_1+k+1+n, x_2+k+1+n, \dots, x_m+k+1+n}}{D_{x_1, x_2, \dots, x_m}} \quad (19)$$

3.6.3 Poistenie na prvé úmrtie

Na rozdiel od dôchodkov, pri ktorých dochádza k výplate v prípade neporušeného stavu, v poisteniach na úmrtie výplata nastáva práve jeho porušením a výplata je jednorazová. V tomto prípade sa m -tica osôb poistí tak, že ak zomrie prvá osoba z m -tice, bude na konci roka vyplatená 1 peňažná jednotka jej pozostalým, alebo dopredu určeným osobám. Označenie je $A_{x_1, x_2, \dots, x_m}$ a na výpočet sa používa vzťah

$$A_{x_1, x_2, \dots, x_m} = \frac{M_{x_1, x_2, \dots, x_m}}{D_{x_1, x_2, \dots, x_m}} \quad (20)$$

V prípade odloženého dočasného poistenia na prvé úmrtie sa m -tica osôb poistí tak, že ak niekto zo skupiny ľudí zomrie medzi vekmi $x_1 + k, x_2 + k, \dots, x_m + k$ a $x_1 + (k+n), x_2 + (k+n), \dots, x_m + (k+n)$, bude na konci roka vyplatená 1 peňažná jednotka jej pozostalým. Súčasnú hodnotu takéhoto poistenia, označenie ${}_k | A_{x_1, x_2, \dots, x_m; \overline{n}}^1$, určíme pomocou vzťahu

$${}_k|A_{x_1, x_2, \dots, x_m: \overline{n}}^1 = \frac{M_{x_1+k, x_2+k, \dots, x_m+k} - M_{x_1+k+n, x_2+k+n, \dots, x_m+k+n}}{D_{x_1, x_2, \dots, x_m}} \quad (21)$$

3.6.4 Zmiešané poistenie pre spojený život

Zmiešané poistenie je jedným z najčastejších poistení, pretože kryje dve riziká, ktoré môžu nastať v životnom poistení, a to úmrtie a dožitie. Týmto sa zabezpečia určité finančné prostriedky pre poisteného, resp. poistených alebo pozostalých. Pri zmiešanom poistení pre m -tícu osôb sa poistí tak, že ak prvé úmrtie nastane v priebehu n rokov, tak pozostali, resp. vopred určená osoba alebo osoby dostanú 1 peňažnú jednotku na konci roka, keď nastalo prvé úmrtie, alebo bude vyplatená jedna peňažná jednotka v prípade, ak sa všetci poistení dožijú ďalších n rokov. Súčasnú hodnotu dostaneme ako súčet súčasných hodnôt poistenia spojeného života na dožitie a poistenia na prvé úmrtie. Teda súčasnú hodnotu označíme $A_{x:\overline{n}}$ vypočítame pomocou vzťahu

$$A_{x:\overline{n}} = \frac{M_{x_1, x_2, \dots, x_m} - M_{x_1+n, x_2+n, \dots, x_m+n}}{D_{x_1, x_2, \dots, x_m}} + \frac{D_{x_1+n, x_2+n, \dots, x_m+n}}{D_{x_1, x_2, \dots, x_m}} \quad (22)$$

3.7 Ďalšie stavy v poistení m -tíc osôb

Okrem stavu spojeného života existujú ďalšie stavy v poistení m -tíc osôb. Doteraz sme uviedli situáciu, kedy sa stav poruší prvým úmrtím v m -tici. V tejto časti sa zaoberáme ďalšími stavmi v poistení m -tíc osôb.

3.7.1 Stav do posledného úmrtia v m -tici osôb

Tento stav existuje, pokiaľ aspoň jedna osoba z danej skupiny osôb je nažive a stav sa poruší v prípade, keď všetci poistenci v m -tice osôb zomrú.

Pravdepodobnosť existencie stavu do posledného úmrtia

Podobne, ako sme uviedli pre spojený život, zavedieme najprv pravdepodobnosť existencie stavu do posledného úmrtia. Označujeme ju symbolom ${}_n p_{x_1, x_2, \dots, x_m}$ a vyjadruje pravdepodobnosť toho, že po n rokoch bude aspoň jedna osoba nažive z m -tice osôb. Vypočítame podľa vzťahu

$${}_n p_{x_1, x_2, \dots, x_m} = 1 - {}_n q_{x_1} \cdot {}_n q_{x_2} \cdot \dots \cdot {}_n q_{x_m} \quad (23)$$

teda od 1 odčítame súčin pravdepodobností, že osoby vo veku $x_1, x_2, \dots, x_m$ sú po n rokoch nažive. Ak vieme, že ${}_nq_{x_i} = 1 - {}_np_{x_i}$ pre $i = 1, 2, \dots, n$ tak

$${}_np_{\overline{x_1, x_2, \dots, x_m}} = 1 - (1 - {}_np_{x_1}) \cdot (1 - {}_np_{x_2}) \cdot \dots \cdot (1 - {}_np_{x_m}) \quad (24)$$

Posledný vzťah môžeme interpretovať nasledovne: Pravdepodobnosť existencie stavu do posledného úmrtia je vlastne doplnok pravdepodobnosti, že v m -tici osôb všetky osoby zomreli. Výraz $(1 - {}_np_{x_1}) \cdot (1 - {}_np_{x_2}) \cdot \dots \cdot (1 - {}_np_{x_m})$ vyjadruje pravdepodobnosť, že zomrela prvá osoba a súčasne druhá, tretia atď. a aj m -tá osoba v m -tici osôb, teda vyjadruje stav, kedy v skupine všetky osoby zomreli v priebehu n rokov³⁷.

Po roznásobení vzťahu (24) a úprave dostaneme vzťah

$${}_np_{\overline{x_1, x_2, \dots, x_m}} = \sum_{i=1}^m {}_np_{x_i} - \sum_{i \neq j} {}_np_{x_i, x_j} + \sum_{i \neq j \neq k} {}_np_{x_i, x_j, x_k} - \dots + (-1)^{m+1} \cdot {}_np_{x_1, x_2, \dots, x_m} \quad (25)$$

Aj v tomto prípade uvedieme príklady nejakých poistení.

Doživotný predlehotný dôchodok

Na vyjadrenie súčasnej hodnoty základných poistných produktov aplikujeme horeuvedený vzťah. Napríklad v prípade doživotného dôchodku, kedy výplata poistnej sumy trvá až do posledného úmrtia v m -tici osôb. Označujeme ju $\ddot{a}_{\overline{x_1, x_2, \dots, x_m}}$ a súčasnú hodnotu dostaneme pomocou vzťahu

$$\ddot{a}_{\overline{x_1, x_2, \dots, x_m}} = \sum_{i=1}^m \ddot{a}_{x_i} - \sum_{i \neq j} \ddot{a}_{x_i, x_j} + \sum_{i \neq j \neq k} \ddot{a}_{x_i, x_j, x_k} - \dots + (-1)^{m+1} \cdot \ddot{a}_{x_1, x_2, \dots, x_m} \quad (26)$$

Poistenie na úmrtie

Podobne ako pri výpočte súčasnej hodnoty doživotného predlehotného dôchodku v stave do posledného úmrtia v m -tici osôb súčasnú hodnotu označujeme $A_{\overline{x_1, x_2, \dots, x_m}}$ a vyjadríme pomocou vzťahu (26):

³⁷ SEKEROVÁ, M. – BILÍKOVÁ, M. Poistná matematika. Bratislava: Vydavateľstvo EKONÓM, 2005.

$$A_{x_1, x_2, \dots, x_m} = \sum_{i=1}^m A_{x_i} - \sum_{i \neq j} A_{x_i, x_j} + \sum_{i \neq j \neq k} A_{x_i, x_j, x_k} - \dots + (-1)^{m+1} \cdot A_{x_i, x_j, \dots, x_k} \quad (27)$$

3.7.2 Stav, keď práve r osôb je nažive v m -tici osôb

Ďalším stavom v m -tici osôb je stav, keď v skupine je práve r osôb nažive. Teda tento stav sa začína $(m-r)$ -tým úmrtím a končí sa $(m-r+1)$ -tým úmrtím v m -tici osôb.

Označenie: $\frac{[r]}{x_1, x_2, \dots, x_m}$.

Podľa predchádzajúcich úvah o stavoch v m -tici osôb vyplýva, že ak $r=m$, tak dostaneme spojený život.

Na určenie pravdepodobnosti toho, že v skupine je práve r osôb je po n rokoch nažive, používame takzvanú Z -metódu. Pravdepodobnosť existencie daného stavu po n rokoch vypočítame podľa vzťahu

$${}_n p_{x_1, x_2, \dots, x_m}^{[r]} = Z_r - \binom{r+1}{1} \cdot Z_{r+1} + \binom{r+2}{2} \cdot Z_{r+2} - \dots + (-1)^{m-r} \cdot \binom{m}{m-r} \cdot Z_m \quad (28)$$

$$\text{kde } Z_1 = \sum_{i=1}^m {}_n p_{x_i}, Z_2 = \sum_{i \neq j} {}_n p_{x_i, x_j}, Z_3 = \sum_{i \neq j \neq k} {}_n p_{x_i, x_j, x_k}, \dots, Z_m = \sum_{i \neq j \neq k \neq \dots \neq m} {}_n p_{x_i, x_j, x_k, \dots, x_m}$$

3.7.3 Stav, keď aspoň r osôb je nažive v m -tici osôb

Tento stav sa začína, keď v danej skupine sú všetky osoby nažive a končí sa $(m-r+1)$ -tým úmrtím.

Určenie pravdepodobnosti toho, že po n rokoch v skupine je aspoň r osoba nažive, určíme pomocou upravenej Z -metódy:

$${}_n p_{x_1, x_2, \dots, x_m}^{[r]} = Z_r - \binom{r}{1} \cdot Z_{r+1} + \binom{r+1}{2} \cdot Z_{r+2} - \dots - (-1)^{m-r} \cdot \binom{m-1}{m-r} \cdot Z_m \quad (29)$$

3.7.4 Zložené stavy m -tice osôb

Zložené stavy predstavujú spojenie dvoch alebo viac stavov jednoduchého alebo spojeného života do jedného vyjadrenia. Napríklad symbol $\ddot{a}_{\overline{xy:zu}}$ označuje súčasnú hodnotu predlehotného dôchodku vo výške 1 peňažnej jednotky, ktorý bude vyplácaný pokiaľ aspoň jeden zo stavov $\overline{xy}$ alebo zu existuje, teda pokiaľ osoby z dvojice osôb zu žijú a aspoň jedna osoba z dvojíc xy žije.

Ak používame substitúciu $\overline{xy} = v$ a $zu = w$, potom pre

$\overline{vw} = v + w - vw = x + y - xy + zu - zu \cdot (x + y - xy) = x + y - xy + uz - xuz - yuz + xyuz$ a odkiaľ dostaneme vyjadrenie $\ddot{a}_{\overline{xy:zu}} = \ddot{a}_x + \ddot{a}_y - \ddot{a}_{xy} + \ddot{a}_{zu} - \ddot{a}_{xzu} - \ddot{a}_{yzu} + \ddot{a}_{xyzu}$ ³⁸.

3.8 Jednostranné dôchodky

Hlavnou úlohou celého poisťovníctva je príprava klientov na nepredvídateľné životné udalosti. V životnom poistení je to predovšetkým riziko úmrtia. Krytie poistného rizika môže spočívať aj v tom, že poistenec chce zabezpečiť finančne svoju rodinu v prípade jeho neočakávaného úmrtia. Na krytie takéhoto rizika slúžia jednostranné dôchodky. Začiatok výplaty dôchodku sa začína po úmrtí konkrétnej osoby. Najčastejšia formy sú vdovský, vdovecký a sirotský dôchodok, keď sa výplata dôchodku začne po smrti jedného partnera v prospech druhého, resp. po smrti rodičov v prospech dieťaťa, resp. detí.

Predpokladajme, že existuje dva nezávislé stavy v skupine ľudí, ktoré označujeme u a v . Dôchodok sa začne vyplácať po porušení stavu u a vyplácanie pokračuje, pokiaľ stav v nebude porušený. Taký dôchodok označujeme a vypočítame podľa vzťahu:

$$a_{u|v} = a_v - a_{uv} \quad (30)$$

Špeciálny prípad, ako sme už vyššie spomínali, je poistenie pre dvojicu osôb x, y . Označenie: $a_{x|y}$ - aktuársky symbol označuje súčasnú hodnotu jednostranného dôchodku na 1 peňažnú jednotku, ktorý sa začne vyplácať po úmrtí osoby x , pokiaľ osoba y je nažive.

³⁸ SEKEROVÁ, M. – BILÍKOVÁ, M. Poistná matematika. Bratislava: Vydavateľstvo EKONÓM, 2005.

Vypočítame podľa vzorca:

$$a_{x|y} = \sum_{t=1}^{\infty} v^t \cdot (1 - {}_t p_x) \cdot {}_t p_y \quad (31)$$

alebo pomocou rozdielu poistných produktov:

$$a_{x|y} = a_y - a_{xy} \quad (32)$$

3.9 Spojité poistenia

Spojité prístup je založený na úvahe, že porušenie spojeného života m -tice môže nastať v ľubovoľnom okamihu uvažovanej poistnej doby³⁹.

Predpokladáme, že funkcia prežitia m -tice osôb $l_{x_1+t, x_2+t, \dots, x_m+t}$ je spojitá a diferencovateľná funkcia premennej t , je definovanej na intervale $\langle 0, \infty \rangle$. Potom úplná intenzita porušenia spojeného života, ktorú označujeme $\mu_{x_1+t, x_2+t, \dots, x_m+t}$ a je definovaná nasledovne

$$\mu_{x_1+t, x_2+t, \dots, x_m+t} = -\frac{d}{dt} \left[\ln l_{x_1+t, x_2+t, \dots, x_m+t} \right] \quad (33)$$

Na základe predpokladu o nezávislosti úmrtí všetkých členov m -tice možno

$$\mu_{x_1+t, x_2+t, \dots, x_m+t} \text{ vyjadriť v tvare } \mu_{x_1+t, x_2+t, \dots, x_m+t} = \sum_{i=1}^m \mu_{x_i+t}.$$

Teraz uvedieme niektoré spojité poistenia m -tice osôb.

Spojité dôchodok

Súčasnú hodnotu spojitého doživotného dôchodku veľkosti 1 peňažnej jednotky ročne, splatnej počas existencie spojeného života vypočítame podľa vzťahu

$$\bar{a}_{x_1, x_2, \dots, x_m} = \int_0^{\infty} {}_t p_{x_1, x_2, \dots, x_m} \cdot v^t dt \quad (34)$$

Súčasnú hodnotu doživotného poistenia na úmrtie s poistnou sumou 1 peňažnej jednotky, splatnou bezprostredne po prvom úmrtí z m -tice vypočítame podľa vzorca:

³⁹ BILÍKOVÁ, M. – JOHANESOVÁ, M. Aktuárske výpočty pre rôzne druhy poistenia m -tice osôb.

Bratislava: Vydavateľstvo EKONÓM, 2008. 86 s.

$$\bar{A}_{x_1, x_2, \dots, x_m} = \int_0^{\infty} {}_t p_{x_1, x_2, \dots, x_m} \mu_{x_1+t, x_2+t, \dots, x_m+t} dt \quad (35)$$

Aproximácia spojitého poistenia na úmrtie diskretným prístupom

Vďaka svojej jednoduchosti sa v praxi často používa aproximácia spojitých poistení, ktorá predpokladá rovnomerné rozdelenie úmrtí v roku úmrtia. Aproximácia sa uskutočňuje pomocou úrokových faktorov:

- faktor jednoduchého úrokovania: $f = 1 + \frac{i}{2}$, (36)

- faktor zloženého úrokovania: $f = (1+i)^{\frac{1}{2}}$, (37)

- faktor spojitého úrokovania: $f = \frac{i}{\delta}$, kde $\delta = \ln(1+i)$. (38)

3.10 Zostavenie úmrtnostných tabuliek v prostredí R

Teraz popíšeme príkazy balíka `lifecontingencies` týkajúce sa životného poistenia. Pri používaní príkazov je potrebné najprv zostaviť úmrtnostné tabuľky. Pre m -ticu osôb musíme zostaviť tabuľky zvlášť pre každú osobu. V prvej časti tejto kapitoly sme predpokladali, že všetky osoby v skupine spĺňajú ten istý zákon úmrtnosti, a práve preto nám stačí vo výpočtoch konštruovať len jednu úmrtnostnú tabuľku a to príkazom:

```
UmrtnostnaTabulka <- new("lifetable", x, lx, name)
```

Zdrojový kód a postup konštrukcie úmrtnostných tabuliek:

```
# Pred prvým použitím nižšie uvedených programových kódov je
# potrebné

# nainštalovať balík lifecontingencies v softvéri R:

install.packages("lifecontingencies")

# načítame balík lifecontingencies

library(lifecontingencies)

# zadáme vekové rozpätie

x <- seq(0, 105, by=1)

# načítame vektor o modelovom počte žijúcich
```

```

lx<-
read.csv("https://www.dropbox.com/s/ie7j8iggc0u6k38/lx2019.csv?dl=1",header=FALSE)

lx<- as.matrix(lx)

# vytvoríme úmrtnostnú tabuľku

UT2019 <- new("lifetable", x = x, lx = lx, name = "UT2019")

# vypíšeme tabuľku

UT2019

```

Ďalšie metódy konštrukcie úmrtnostných tabuliek je možné nájsť v [2].

Aktuárske úmrtnostné tabuľky v prostredí R

Aktuári okrem štandardných úmrtnostných tabuliek pracujú s úmrtnostnými tabuľkami obsahujúce komutačné čísla.

1. Metóda – konštrukcia pomocou vzorcov

```

# úroková sadzba

i <- 0.007

# vytvoríme úmrtnostnú tabuľku

x <- seq(0,105,by=1)

lx<-
read.csv("https://www.dropbox.com/s/ie7j8iggc0u6k38/lx2019.csv?dl=1",header=FALSE)

lx<- as.matrix(lx)

# uložíme dĺžku vektora lx do premennej

l<-length(lx)

# výpočet ukazovateľa dx

dx1 <- lx[]-lx[1:l+1]

```

```

dx <- c(dx1[1:105], tail(lx, n=1))

# výpočet komutačných čísel

Dx <- lx*(1+i)^(-x[])
Cx <- dx*(1+i)^(-x[]-1)

Nx <- rev(cumsum(rev(Dx)))
Sx <- rev(cumsum(rev(Nx)))
Mx <- rev(cumsum(rev(Cx)))
Rx <- rev(cumsum(rev(Mx)))

# vytvoríme tabuľku

UT2019KC<-data.frame(x=x, lx=lx, Dx=Dx, Nx=Nx, Sx=Sx, Cx=Cx, Mx=Mx,
Rx=Rx)

# vypíšeme tabuľku

UT2019KC

```

V prípade, ak potrebujeme z tabuľky zistiť hodnotu komutačného čísla, napr. C_{50} , postup je jednoduchý:

```
C50 <- UT2019KC$Cx [UT2019KC$x==50];C50
```

Nevýhoda tejto metódy je, že tabuľka nebude mať typ `actuarialtable` a v ďalších príkazoch je to nevyhnutné, aby sme zákon úmrtnosti zadali takým typom objektu.

2. Metóda – konštrukcia pomocou príkazu `actuarialtable`⁴⁰

```

# načítame balík lifecontingencies

library(lifecontingencies)

# zadáme vekové rozpätie

```

⁴⁰ HORVÁTHOVÁ, S. 2019. Používanie aktuárskych balíkov v softvéri R. Bakalárska práca. Fakulta matematiky, fyziky a informatiky, Univerzita Komenského, Bratislava.

```

x <- seq(0,105,by=1)

# úroková sadzba

i <- 0.007

# načítame vektor lx

lx<-
read.csv("https://www.dropbox.com/s/ie7j8iggc0u6k38/lx2019.csv?dl=1",header=FALSE)

lx<- as.matrix(lx)

# zavoláme funkciu actuarialtable

UT2019KC2 <- new("actuarialtable", interest=i, x=x,

lx=lx, name="UT2019KC2")

# vypíšeme tabuľku

UT2019KC2

```

Dostali sme rovnakú tabuľku typu actuarialtable ako pri prvej metóde, len táto tabuľka neobsahuje stĺpec S_x .

3.11 Výpočet súčasných hodnôt poistných produktov v prostredí R

V tejto časti záverečnej práce uvedieme príkazy zabudované v softvéri R, ktoré slúžia na oceňovanie poistných produktov pre m -ticu osôb. Charakterizujeme parametre príkazov a aplikujeme ich na ilustračných príkladoch.

3.11.1 Pravdepodobnosti existencie daného stavu

Na výpočet pravdepodobnosti prežitia v m -tici osôb používame príkaz `pxyzt` a v prípade úmrtia `qxyt`. V balíku existujú aj samotné príkazy na výpočet pravdepodobnosti prežitia a úmrtia pre dvojicu osôb `pxyt` a `qxyt`. Príkazy majú rovnaké parametre.

Príkazy `pxyt` a `qxyt` majú nasledujúce tvary⁴¹:

```
pxyt(objectx, objecty, x, y, t, status = "joint")
```

```
qxyt(objectx, objecty, x, y, t, status = "joint")
```

Parametre príkazov:

- `objectx` – úmrtnostná tabuľka pre osobu x ,
- `objecty` – úmrtnostná tabuľka pre osobu y ,
- `x` – vek osoby x ,
- `y` – vek osoby y ,
- `t` – doba pre ktorú chceme vypočítať pravdepodobnosť,
- `status` – stav v dvojice osôb, v prípade "joint" počítame pravdepodobnosť pre spojený život, v prípade "last" počítame pravdepodobnosť pre stav do posledného úmrtia.

Obr. 1: výpočet pravdepodobnosti toho, že 35 a 37 ročné osoby budú nažive po 5 rokov

```
> pxyt(UT2019KC2, UT2019KC2, 35, 37, 5, status = "joint")  
[1] 0.9874773
```

Zdroj: vlastné spracovanie

Príkazy `pxyzt` a `qxyzt` majú nasledujúce tvary:

```
pxyzt(tablesList, x, t, status="joint", fractional=rep("linear",  
length(tablesList)), ...)
```

```
qxyzt(tablesList, x, t, status="joint", fractional=rep("linear",  
length(tablesList)), ...)
```

Parametre sú podobné ako pri predchádzajúcich príkazoch, rozdiel spočíva v tom, že parametre majú iný typ objektu.

Parametre príkazov:

⁴¹ SPEDICATO, G. A., KAINHOFER, R., OWENS, J. K., DUTANG CHRISTOPHE, D., SCHIRMACHER, E., CLEMENTE G. P., WILLIAMS I. 2021. Package 'lifecontingencies'

- `tablesList` – objekt typu zoznam, ktorého prvkom sú úmrtnostné tabuľky alebo úmrtnostné tabuľky obsahujúce komutačné čísla,
- `x` – vektor, ktorý obsahuje veky osôb v skupinovom prvku, rozmer musí byť rovnaký s parametrom `tablesList`,
- `t` – doba, pre ktorú chceme vypočítať pravdepodobnosť,
- `fractional` – rozloženie úmrtnosti počas roka, môže byť "linear" – lineárna, "hyperbolic" – hyperbolická, "constant force" – konštantná.

Obr. 2: výpočet pravdepod. toho, že osoby vo veku 35,37 a 38 budú po 5 rokoch nažive

```
> lifetables <- c(UT2019KC2, UT2019KC2, UT2019KC2)
> x <- c(35,37,38)
> pxyz(lifetables,x,5,status="joint",fractional="linear")
[1] 0.9804034
```

Zdroj: vlastné spracovanie

3.11.2 Poistenie na úmrtie

Skupina ľudí sa poistí tak, že v prípade porušenia daného stavu bude vyplatená pozostalým 1 peňažná jednotka na konci roka, kedy došlo k porušeniu stavu.

Príkaz, ak $m = 2$, má nasledujúci tvar:

```
Axyn(tablex, x, tabley, y, n, i, m, k = 1, status = "joint",
type = "EV")
```

Príkaz má nasledujúce parametre:

- `tablex` – úmrtnostná tabuľka s komutačnými číslami (typ `actuarialtable`) pre osobu x ,
- `x` – vek osoby x ,
- `tabley` – úmrtnostná tabuľka s komutačnými číslami (typ `actuarialtable`) pre osobu y ,
- `y` – vek osoby y ,
- `n` – poistná doba,
- `i` – úroková miera, ak sa v príkaze neuvádza, tak sa automaticky používa úroková sadzba z úmrtnostnej tabuľky,
- `m` – doba odkladu,

- *k* – frekvencia platieb, t. j. počet možných termínov vyplácania poistnej sumy za jednu časovú jednotku,
- *status* – stav v dvojici osôb, v prípade "joint" počítame pravdepodobnosť pre spojený život, v prípade "last" počítame pravdepodobnosť pre stav do posledného úmrtia,
- *type* – označuje, či ide o tradičný výpočet očakávanej hodnoty poistného produktu (expected value, EV) alebo o výpočet pomocou stochastickej simulácie (v prípade ST) súčasnej hodnoty poistného produktu.

Obr. 3: výpočet súčasnej hodnoty poistenia na úmrtie pre dvojice osôb

```
> Axyn(UT2019KC2,35,UT2019KC2,37,n=10,i=0.007,status="joint",type="EV")
[1] 0.03029279
```

Zdroj: vlastné spracovanie

Príkaz, ak sa poistí *m* -tica osôb, má tvar

```
Axyzn(tablesList, x, n, i, m, k = 1, status = "joint",
type = "EV", power=1)
```

Príkaz má nasledujúce parametre:

- *tablesList* – objekt typu zoznam, ktorého prvkom sú úmrtnostné tabuľky alebo úmrtnostné tabuľky obsahujúce komutačné čísla,
- *x* – vektor, ktorý obsahuje veky osôb v skupinovom prvku, rozmer musí byť rovnaký s parametrom *tablesList*,
- *n* – poistná doba,
- *m* – doba odkladu,
- *k* – frekvencia platieb, t. j. počet možných termínov vyplácania poistnej sumy za jednu časovú jednotku,
- *status* – stav v dvojici osôb, v prípade "joint" počítame pravdepodobnosť pre spojený život, v prípade "last" počítame pravdepodobnosť pre stav do posledného,
- *type* – označuje, či ide o tradičný výpočet očakávanej hodnoty poistného produktu (expected value, EV) alebo o výpočet pomocou stochastickej simulácie (v prípade ST) súčasnej hodnoty poistného produktu,
- *power* – rád počiatočného momentu bázičného poistenia na úmrtie.

Obr. 4: výpočet súčasnej hodnoty poistenia na úmrtie pre trojicu osôb

```
> lifetables <- c(UT2019KC2,UT2019KC2,UT2019KC2)
> x <- c(35,37,38)
> Axyzn(lifetables,x,n=10,i=0.007,status="joint",type="EV",power=1)
[1] 0.04796336
```

Zdroj: vlastné spracovanie

3.11.3 Dôchodky pre m -tice osôb

Príkazy pre výpočet súčasnej hodnoty predlehotného a polehotného dôchodku sú rovnaké, keď parameter `payment = "due"` príkaz počíta súčasnú hodnotu predlehotného dôchodku, ak `payment = "immediate"` počíta súčasnú hodnotu polehotného dôchodku.

Príkaz pre dvojicu osôb má tvar⁴²

```
axyn(tablex, tabley, x, y, n, i, m, k = 1, status = "joint",
type = "EV", payment="advance")
```

Príkaz má nasledujúce parametre:

- `tablex` – úmrtnostná tabuľka s komutačnými číslami (typ `actuarialtable`) pre osobu x ,
- `tabley` – úmrtnostná tabuľka s komutačnými číslami (typ `actuarialtable`) pre osobu y ,
- x – vek osoby x ,
- y – vek osoby y ,
- n – poistná doba,
- i – úroková miera, ak sa v príkaze neuvádza, tak sa automaticky používa úroková sadzba z úmrtnostnej tabuľky,
- m – doba odkladu,
- k – frekvencia platieb, t. j. počet možných termínov vyplácania poistnej sumy za jednu časovú jednotku,

⁴² SPEDICATO, G. A., KAINHOFER, R., OWENS, J. K., DUTANG CHRISTOPHE, D., SCHIRMACHER, E., CLEMENTE G. P., WILLIAMS I. 2021. Package 'lifecontingencies'

- `status` – stav v dvojici osôb, v prípade "joint" počítame pravdepodobnosť pre spojený život, v prípade "last" počítame pravdepodobnosť pre stav do posledného úmrtia,
- `type` – označuje, či ide o tradičný výpočet očakávanej hodnoty poistného produktu (expected value, EV) alebo o výpočet pomocou stochastickej simulácie (v prípade ST) súčasnej hodnoty poistného produktu,
- `payment` – označuje, či ide o polehotný alebo predlehotný dôchodok; v prípade predlehotného dôchodku používame nastavenie "due", v prípade polehotného dôchodku používame nastavenie "immediate".

Obr. 5: výpočet súčasnej hodnoty dôchodku pre dvojice osôb

```
> axyn(UT2019KC2,UT2019KC2,35,37,n=10,i=0.007,status="joint",type="EV")
[1] 9.577736
```

Zdroj: vlastné spracovanie

Príkaz pre m -ticu osôb má nasledujúci tvar:

```
axyzn(tablesList, x, n, i, m, k = 1, status = "joint", type =
"EV", power=1, payment="advance")
```

Príkaz má parametre:

- `tablesList` – objekt typu zoznam, ktorého prvkom sú úmrtnostné tabuľky alebo úmrtnostné tabuľky obsahujúce komutačné čísla,
- `x` – vektor, ktorý obsahuje vektory osôb v skupinovom prvku, rozmer musí byť rovnaký s parametrom `tablesList`,
- `n` – poistná doba,
- `m` – doba odkladu,
- `k` – frekvencia platieb, t. j. počet možných termínov vyplácania poistnej sumy za jednu časovú jednotku,
- `status` – stav v dvojice osôb, v prípade "joint" počítame pravdepodobnosť pre spojený život, v prípade "last" počítame pravdepodobnosť pre stav do posledného,
- `type` – označuje, či ide o tradičný výpočet očakávanej hodnoty poistného produktu (expected value, EV) alebo o výpočet pomocou stochastickej simulácie (v prípade ST) súčasnej hodnoty poistného produktu,

- `power` – rád počiatočného momentu bazického poistenia na úmrtie,
- `payment` – označuje, či ide o polehotný alebo predlehotný dôchodok; v prípade predlehotného dôchodku používame nastavenie "due", v prípade polehotného dôchodku používame nastavenie "immediate".

Obr. 6: výpočet súčasnej hodnoty dôchodku pre trojicu osôb

```
> lifetables <- c(UT2019KC2,UT2019KC2,UT2019KC2)
> x <- c(35,37,38)
> axyzn(lifetables,x,n=10,i=0.007,status="joint",type="EV",power=1,payment="advance")
[1] 9.51185
```

Zdroj: vlastné spracovanie

Poznámka: Ak spustíme príkazy `pxyt`, `qxyt`, `Axyn`, `axyn`, tak softvér vypíše upozornenie, že daný výpočet môžeme realizovať aj pomocou ekvivalentných príkazov (`pxyzt`, `qxyzt`, `Axyzn`, `axyzn`).

4 Výsledky práce a diskusia

Posledná časť diplomovej práce sa zaoberá vlastnými funkciami. Táto kapitola je určená ako rozšírenie balíka `lifecontingencies`, teda obsahuje algoritmy, ktoré chýbajú z tohto balíka. Príkazy sú stručne charakterizované a aplikované v rámci ilustračných príkladov. Zdrojové kódy sú výsledkom vlastného programovania, a keďže sú rozsiahle, sú dostupné v prílohách tejto diplomovej práce.

4.1 Aktuárske úmrtnostné tabuľky pre dvojicu osôb pomocou vlastnej funkcie

Knižnica `lifecontingencies` neobsahuje príkaz na konštrukciu úmrtnostných tabuliek pre viac osôb, ale vytvorenie vlastnej funkcie na takýto účel v prostredí R je možné. Naša funkcia `LTxy` vytvorí úmrtnostnú tabuľku pre dvojicu osôb a z tabuľky môžeme vybrať komutačné číslo, ktoré potrebujeme. Funkcia má nasledujúce parametre:

- `i` – úroková miera,
- `x` – vek osoby x ,
- `y` – vek osoby y ,
- `number` – typ komutačného čísla.

Aby sme mohli používať náš príkaz, je potrebné najprv skopírovať funkciu do softvéru R.

```
LTxy <- function(i, x, y, number) {  
  
  i <- i  
  
  diff <- y-x  
  
  Dxy <- NULL  
  
  Cxy <- NULL  
  
  lx<-as.double(as.vector(read.table  
  
    ("https://www.dropbox.com/s/ie7j8iggc0u6k38/lx2019.csv?dl=1",  
    header=FALSE)$V1))  
  
  lx<- as.matrix(lx)
```

```

for(j in 1:(105-diff)){
Dxy[j] <- c(lx[j]*lx[j+diff]*(1+i)^(-(j+j+diff-2)/2))}

for(h in 1:(105-diff)){
Cxy[h] <- c((lx[h]*lx[h+diff]-lx[h+1]*lx[h+diff+1])*(1+i)^(-(h+h+diff-2)/2+1)))}

Nxy <- rev(cumsum(rev(Dxy)))
Sxy <- rev(cumsum(rev(Nxy)))
Mxy <- rev(cumsum(rev(Cxy)))
Rxy <- rev(cumsum(rev(Mxy)))

if(number=="Dxy"){
numb<-Dxy[x+1]}

if(number=="Nxy"){
numb<-Nxy[x+1]}

if(number=="Sxy"){
numb<-Sxy[x+1]}

if(number=="Cxy"){
numb<-Cxy[x+1]}

if(number=="Mxy"){
numb<-Mxy[x+1]}

if(number=="Rxy"){
numb<-Rxy[x+1]}

numb}

```

Ilustračný príklad 1:

Výpočet súčasnej hodnoty doživotného predlehotného dôchodku osoby x a y , ktorý sa vypláca, pokiaľ obidve osoby sú nažive: $x = 35$; $y = 37$.

Riešenie:

A) Riešenie pomocou príkazu `axyn`:

Obr. 7: výpočet doživotného predlehotného dôchodku pre 35 a 37 ročné osoby pomocou príkazu `axyn`

```
> axyn(UT2019KC2,UT2019KC2,35,37,i=0.007,status="joint")  
[1] 31.77235
```

Zdroj: vlastné spracovanie

B) Riešenie pomocou vzťahu (14) a pomocou vlastnej funkcie v programovacom jazyku R:

$$\ddot{a}_{35:37} = \frac{N_{35:37}}{D_{35:37}}$$

Obr. 8: výpočet doživotného predlehotného dôchodku pre 35 a 37 ročné osoby

```
> LTxy(i=0.007,x=35,y=37,number="Nxy")/LTxy(i=0.007,x=35,y=37,number="Dxy")  
[1] 31.77235
```

Zdroj: vlastné spracovanie

Súčasná hodnota predlehotného dôchodku, ktorý vypláca 1 peňažnú jednotku, kým obidve osoby sú nažive, je 31,77 p. j.

4.2 Jednostranné dôchodky

Začiatok výplaty dôchodku sa začína po úmrtí osoby uvedenej v poisťnej zmluve. Najčastejšia formy sú vdovský, vdovecký a sirotsky dôchodok, keď sa výplata dôchodku začne po smrti jedného partnera v prospech druhého, resp. po smrti rodičov v prospech dieťaťa.

Na výpočet súčasnej hodnoty jednostranného dôchodku používame vlastný zdrojový kód s parametrami:

- `tableslist1` – úmrtnostná tabuľka, ktorou sa riadi zákon úmrtnosti osoby, ktorej smrť musí nastať, aby sa mohla začať výplata jednostranného dôchodku (môže byť závislá úmrtím viac osôb, preto sa zadáva vo forme vektora),
- `tableslist2` – úmrtnostná tabuľka, ktorou sa riadi zákon úmrtnosti osoby, ktorej sa vypláca dôchodok (dôchodok môže byť vyplatený viacerým osobám, preto sa zadáva vo forme vektora),
- `vek1` – vek osoby, ktorej smrť musí nastať, aby sa výplata jednostranného dôchodku mohla začať (zadáva sa ako vektor),
- `vek2` – vek osoby, ktorej sa vypláca dôchodok, pokiaľ bude nažive (zadáva sa ako vektor),
- `status1` – stav v poistení m -tíc osôb, ktorých smrť musí nastať, aby sa výplata dôchodku mohla začať (v prípade, ak ide o jednu osobu, parameter nemusíme zadávať),
- `status2` – stav v poistení m -tíc osôb, ktorým bude vyplatený dôchodok (v prípade, ak ide o jednu osobu, parameter nemusíme zadávať),
- m – frekvencia platenia poistného.

Pred použitím funkcie je potrebné skopírovať zdrojový kód z prílohy A.

Ilustračný príklad 2:

Rodičia chcú zabezpečiť formou jednostranného dôchodku svoje dieťa v prípade, ak obidvaja zomrú. Matka (x) má 30 rokov, otec (y) má 32 rokov, dieťa (z) má 10 rokov.

Riešenie:

Označenie dôchodku $a_{\overline{xy}|z}$.

A) Podľa vzťahov (25) a (30) platí $a_{\overline{xy}|z} = a_z - a_{\overline{xyz}} = a_z - a_{xz} - a_{yz} + a_{xyz}$.

Vieme, že $a_{10} = 54,35029$, $a_{10,30} = 40,16181$, $a_{10,32} = 38,94777$, $a_{10,30,32} = 34,74315$.

Potom po dosadení do vyššie uvedeného vzťahu dostaneme: $a_{\overline{xy}|z} = 9,98386$ p. j.

B) Riešenie pomocou vlastnej funkcie v programovacom jazyku R:

Obr. 9: výpočet súčasnej hodnoty jednostranného dôchodku

```
> jednostranne(c(UT2019KC2,UT2019KC2),c(UT2019KC2),c(30,32),c(10),i=0.007,"last","joint")  
[1] 9.983862
```

Zdroj: vlastné spracovanie

Súčasná hodnota jednostranného dôchodku, ktorý sa vypláca 1 p. j. dieťaťu v prípade, ak zomrú jeho rodičia, je 9,98 p. j.

4.3 Spojité poistenie na úmrtie pre m -tice osôb

Skupina ľudí sa poistí tak, že bezprostredne po úmrtí jednej osoby z m -tice bude vyplatená 1 peňažná jednotka.

Na výpočet súčasnej hodnoty spojitého poistenia na úmrtie používame vlastný zdrojový kód:

```
Axyncon <-  
function(tableslist,x,n,i,m,k=1,status,interest=c("simple","c  
ompound","continuous"))  
  
{library(lifecontingencies)  
  
if(interest=="simple")  
  
{f <- 1 + i/2}  
  
if(interest=="compound")  
  
{f <- (1+i)^0.5}  
  
if(interest=="continuous")  
  
{f <- i/(log(1+i))}  
  
Axyzn <- f*Axyzn(tablesList=tableslist,  
  
x=x,n=n,i=i,m=m,k=k,status=status,type="EV",power=1)  
  
Axyzn}
```

Parametre príkazu:

- `tableslist`, `x`, `n`, `i`, `m`, `k`, `status` – sú rovnaké ako pri príkazoch `pxyz`, `qxyz`, `axyz`, `Axyz`,
- `interest` – typ aproximácie, v prípade, ak používame nastavenie "simple", príkaz používa jednoduché úrokovanie, v prípade "compound" zložené úrokovanie, v prípade "continuous" spojitú úrokovanie.

Ilustračný príklad 3:

Vypočítajme súčasnú hodnotu spojitého poistenia na úmrtie pre 30 a 35 ročné osoby trvajúce 10 rokov. Výpočet uskutočníme pomocou faktora jednoduchého úrokovania.

Riešenie:

A) Výpočet pomocou vzťahu (36): $\bar{A}_{xy:\overline{10}|} = \left(1 + \frac{i}{2}\right) A_{xy:\overline{10}|}$. Ak $A_{30;35:\overline{10}|} = 0,02314368$ p. j.

Po dosadení dostaneme $\bar{A}_{30,35:\overline{10}|} = \left(1 + \frac{0,007}{2}\right) \cdot 0,02314368 = 0,023224682$ p. j.

B) Pomocou vlastnej funkcie v programovacom jazyku R

Obr. 10: výpočet súčasnej hodnoty spojitého poistenia na úmrtie

```
> Axyncon(c(UT2019KC2,UT2019KC2),c(30,35),n=10,i=0.007,interest = "simple",status="joint")  
[1] 0.02322469
```

Zdroj: vlastné spracovanie

Súčasná hodnota spojitého poistenia na úmrtie pre 30 a 35 ročné osoby trvajúce 10 rokov pri aproximácii jednoduchého úrokovania je 0,023 p. j.

4.4 Z-metóda a upravená Z-metóda

Na výpočet pravdepodobnosti, že v m -tici je aspoň r osôb alebo práve r osôb je nažive používame vlastnú funkciu v programovacom jazyku R `npxyzr`.

Parametre príkazu:

- `tableslist` – objekt typu zoznam, ktorého prvkom sú úmrtnostné tabuľky alebo úmrtnostné tabuľky obsahujúce komutačné čísla,

- `ages` – vektor, ktorý obsahuje veku osôb v skupinovom prvku, rozmer musí byť rovnaký s parametrom `tableslist`,
- `n` – doba, pre ktorú chceme vypočítať pravdepodobnosť,
- `r` – počet osôb, ktoré sú práve nažive v `m` - tici alebo minimálny počet osôb, ktoré sú nažive v `m` - tici,
- `probability` – v prípade "exactly" vypočíta pravdepodobnosť toho, že po `n` rokoch bude nažive práve `r` osoba, v prípade "atleast" vypočíta pravdepodobnosť toho, že po `n` rokoch bude nažive aspoň `r` osôb.

Pred použitím funkcie je potrebné skopírovať zdrojový kód z prílohy B.

Ilustračný príklad 4:

Vypočítajme pravdepodobnosť toho, že z trojice (20,21,22) budú po 5 rokoch práve 2 nažive.

Riešenie:

A) Výpočet pomocou vzťahu (28):

$${}_5P_{\frac{[2]}{20,21,22}} = {}_5P_{20,21} + {}_5P_{20,22} + {}_5P_{21,22} - 3 \cdot {}_5P_{20,21,22}$$

Vieme, že ${}_5P_{20,21} = 0,99499$, ${}_5P_{20,22} = 0,9949384$, ${}_5P_{21,22} = 0,9949674$,
 ${}_5P_{20,21,22} = 0,9924574$, potom po dosadení do vyššieho uvedeného vzťahu dostaneme

$${}_5P_{\frac{[2]}{20,21,22}} = 0,99499 + 0,9949384 + 0,9949674 - 3 \cdot 0,9924574 = 0,0075236$$

B) Pomocou vlastnej funkcie v programovacom jazyku R

Obr. 11: výpočet pravdepodobnosti toho, že z trojice budú po 45 rokoch práve 2 osoby nažive

```
> npxyzr(c(UT2019KC2,UT2019KC2,UT2019KC2),c(20,21,22),n=5,r=2,probability = "exactly")
[1] 0.007523575
```

Zdroj: vlastné spracovanie

Pravdepodobnosť toho, že z trojice (20,21,22) budú po 5 rokoch práve 2 nažive, je približne 0,007.

4.5 Poistenia so stavmi, keď práve a aspoň r osoba je nažive

Existujú poistenia, pri ktorých k výplate poistného plnenia prichádza pri zrušení, resp. existencii daného stavu.

Parametre príkazov `nxyz` a `nAxyz` sú rovnaké ako pri príkaze `npxyzr`.

Zdrojový kód na výpočet dôchodkov, ktoré sú vyplácané, kým existuje stav, že práve alebo aspoň r osôb je nažive:

```
nxyz <- function(tableslist, ages, n, r, i, probability)
{v <- 1/(1+i)
axyz<-0
for(j in 0:(n-1))
{axyz<- axyz+v^(j)*npxyzr(tableslist=tableslist, ages,
n=j, r=r, probability=probability) }
axyz}
```

Ilustračný príklad 5:

Vypočítajme súčasnú hodnotu dôchodku, ktorý vypláca 1 peňažnú jednotku, kým práve dve osoby sú z trojice (20,21,22) nažive.

Riešenie:

A) Pomocou vzťahu (28): $\ddot{a}_{\overline{[2]}_{xyz}} = \ddot{a}_{xy} + \ddot{a}_{xz} + \ddot{a}_{yz} - 3 \cdot \ddot{a}_{xyz}$. Ak, $\ddot{a}_{20;21} = 42,59639$,

$\ddot{a}_{20,22} = 42,23838$, $\ddot{a}_{21,22} = 41,92902$ a $\ddot{a}_{20,21,22} = 39,14399$. Po dosadení dostaneme:

$$\begin{aligned}\ddot{a}_{\overline{[2]}_{20,21,22}} &= \ddot{a}_{20,21} + \ddot{a}_{20,22} + \ddot{a}_{21,22} - 3 \cdot \ddot{a}_{20,21,22} = 42,59639 + 42,23838 + 41,92902 \\ &\quad - 3 \cdot 39,14399 = 9,33182\end{aligned}$$

B) pomocou vlastnej funkcie v programovacom jazyku R

Obr. 12: výpočet súčasnej hodnoty dôchodku

```
> nxyz(c(UT2019KC2,UT2019KC2,UT2019KC2),c(20,21,22),r=2,i=0.007,probability="exactly",n=84)
[1] 9.331832
```

Zdroj: vlastné spracovanie

Súčasná hodnota dôchodku, ktorý vypláca 1 p. j., kým práve dve osoby sú z trojice (20,21,22) nažive, je 9,33 p. j.

Poistenia na úmrtie

Zdrojový kód na výpočet súčasnej hodnoty poistenia na úmrtie, ktoré vyplatí 1 peňažnú jednotku po úmrtí r -tej osoby v m -tici:

```
nAxyz <- function(tableslist, ages, n, r, i)

{d <- i/(1+i)

v <- 1/(1+i)

axyz<-0

Axyz <- 0

for(j in 0:(n-1))

{axyz<- axyz+v^(j)*npxyzr(tableslist=tableslist,

ages, n=j, r=r, probability="atleast") }

Axyz<-1-d*axyz

Axyz}
```

Ilustračný príklad 6:

Vypočítajme súčasnú hodnotu poistenia na úmrtie, ktoré vypláca 1 peňažnú jednotku po druhom úmrtí v trojici (20,21,22).

Riešenie:

A) pomocou vlastnej funkcie `naxyz` a vzťahu $A_{\frac{2}{x,y,z}} = 1 - d \cdot \ddot{a}_{\frac{2}{x,y,z}}$, kde $d = \frac{i}{1+i}$.

Obr. 13: výpočet súčasnej hodnoty poistenia na úmrtie

```
> 1-(0.007/1.007)*naxyz(c(UT2019KC2,UT2019KC2,UT2019KC2),c(20,21,22),r=2,i=0.007,probability="atleast",n=84)
[1] 0.6630281
```

Zdroj: vlastné spracovanie

B) pomocou vlastnej funkcie v programovacom jazyku R

Obr. 14: výpočet súčasnej hodnoty poistenia na úmrtie

```
> nAxyz(c(UT2019KC2,UT2019KC2,UT2019KC2),c(20,21,22),n=84,r=2,i=0.007)
[1] 0.6630281
```

Zdroj: vlastné spracovanie

Súčasná hodnota poistenia na úmrtie, ktoré vypláca 1 p. j. po druhom úmrtí v trojici je 0,66 p. j.

Záver

Oceňovanie je pri tvorbe poistných produktov kľúčovým postupom. Analýza senzitivnosti produktu na jednotlivé predpoklady je nevyhnutná na uchovanie ziskovosti poisťovne. V poisťovnej praxi sa často aktuári stretávajú s komplikovanými výpočtami, preto v dnešnej dobe poisťovníctvo úzko súvisí s používaním softvérových aplikácií a programovacích jazykov. Cieľom práce bolo zmapovať súčasný stav poistenia m -tíc osôb a možnosti využitia programovacieho jazyka R pre určenie súčasných hodnôt takýchto poistení.

V prvej kapitole sme skonštatovali poistné produkty, ktoré ponúkajú poisťovne na Slovensku. Ďalej sme charakterizovali programovacie jazyky používané v poisťovnej praxi. Kapitola analyzuje balík `lifecontingencies` ako hlavnú rozšírenú knižnicu životného poistenia v štatistickom softvéri R.

V druhej časti práce sme uviedli stručný cieľ práce a konkretizovali čiastočné ciele. Naším cieľom bolo, aby sme charakterizovali balík `lifecontingencies` a príkazy, ktoré sa používajú pri výpočtoch poistenia m -tíc osôb. Hlavným cieľom bolo vytvorenie vlastných funkcií, ktoré nie sú zabudované do balíka `lifecontingencies`.

Tretia kapitola obsahovala metodické východiská. Stručne opisovala základné princípy poistenia osôb, ako aj predpoklady výpočtov použitých v životnom poistení. Charakterizovala poistné produkty poistenia m -tíc osôb. Posledná časť tejto kapitoly sa zaoberala konštrukciou úmrtnostných tabuliek pomocou programovacieho jazyka R a príkazmi, ktoré sú dostupné v balíku `lifecontingencies` pre oceňovanie poistných produktov pre m -ticu osôb.

Posledná kapitola diplomovej práce sa zaoberala tvorbou vlastných algoritmov. Súčasťou práce v našom prípade sú aj prílohy, nakoľko obsahujú zdrojové kódy, ktoré kvôli rozsiahlosti nie sú súčasťou poslednej kapitoly.

Výsledkom práce okrem charakteristík príkazov zabudovaných v programovacom jazyku R sú aj vlastné algoritmy, ktoré sú užitočnými aparátmi pre životné poisťovne. Okrem praxe vlastné príkazy sú prínosné aj pre študentov, ktorí študujú predmet Matematiku pre životné poistenie, resp. študijný program Aktuárstvo. Pomocou vlastných zdrojových kódov sme dopĺňali balík `lifecontingencies` a, tým sme rozšírili možnosti využitia programovacieho jazyka R pre poistenia m -tíc osôb.

Ako odporúčanie pre ďalšie programovanie v rámci štatistického softvéru R a v rámci problematiky poistenia m -tíc osôb je možné naprogramovať aj algoritmy pre výpočet súčasnej hodnoty kontingenčných poistení, kde výplata poistného plnenia závisí od poradia úmrtí v m -tice osôb.

Použitá literatúra

- [1] BILÍKOVÁ, M. – JOHANESOVÁ, M. Aktuárske výpočty pre rôzne druhy poistenia *m* -tice osôb. Bratislava: Vydavateľstvo EKONÓM, 2008. 86 s. ISBN 978-80-225-2530-5.
- [2] HORVÁTHOVÁ, S. Používanie aktuárskych balíkov v softvéri R. Bakalárska práca. Fakulta matematiky, fyziky a informatiky, Univerzita Komenského v Bratislave, 2019.
- [3] KRÁL, M. Excel VBA. Brno: Vydavateľstvo Computer Press, 2012. 504 s. ISBN 978 - 80-251-2358-4.
- [4] R Core Team (2020). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.
- [5] SAKÁLOVÁ, K. Oceňovanie produktov v životnom poistení s. 46. Bratislava: Vydavateľstvo EKONÓM, 2001. ISBN 80-225-1350-4.
- [6] SEKEROVÁ, M. – BILÍKOVÁ, M. Poistná matematika. Bratislava: Vydavateľstvo: EKONÓM, 2005. 180 s. ISBN 80-225-2001-2.
- [7] TOKÁROVÁ, D. Tabuľky komutačných čísel pre dvojice a trojice osôb a ich použitie. Bakalárska práca. Fakulta hospodárskej informatiky, Ekonomická univerzita v Bratislave, 2012.

Internetové zdroje:

- [7] SPEDICATO, G. A., KAINHOFER, R., OWENS, J. K., DUTANG CHRISTOPHE, D., SCHIRMACHER, E., CLEMENTE G. P. , WILLIAMS I. 2021. Package 'lifecontingencies'. [cit. 26.02.2021] Dostupné na adrese: <https://cran.r-project.org/web/packages/lifecontingencies/lifecontingencies.pdf>.
- [8] Štatistický úrad Slovenskej republiky. Úmrtnostné tabuľky (roky 1996 - 2019): Slovenská republika. Obyvateľstvo a migrácia: Tabuľky života. [cit. 14.03.2021] Dostupné na adrese: <https://slovak.statistics.sk/wps/portal/ext/themes/demography/>

population/indicators/!ut/p/z1/jZHLboMwEEW_pQu29tjmYbozVHHcoLaQkhBv
KhJRQsUjIjT8fmnUTaXWYVYz0rkzo3uxxhnWbX6pynyujavp3mn3bfYUzwIiAA
ebCgo7ykhqziWyqd4awJWqYf1HH0oxdL2IgAeSQeUWKaJHzMGgs3Twz8lYJ7e
AGjz- i3WphNXB67AYv1ChW_L8CF5XoB6DSlfRy4F6d4AgPwAJpNuvfmIdbV
v0HhoECBOwPNdwp9dQ75Dlm0e8ZLrPviveiLHn32U_bHYTid7y2wYBxHVVHZd
WRfo0DUW_CU5ducBZ79JfGrSNM2gUh9OfYnE3Rf4pmuV/dz/d5/L2dBISEvZ0F
BIS9nQSEh/

- [9] <https://openiazoch.zozn.am.sk/zivotne-poistenie/co-je-to> [cit. 08.07.2021]
- [10] <https://www.slov-lex.sk/pravne-predpisy/SK/ZZ/2015/39/> [cit. 08.07.2021]
- [11] <https://www.financnykompas.sk/clanok/priipoistenie> [cit. 11.07.2021]
- [12] <https://poistovne.sk/poistny-kmen> [cit. 11.07.2021]
- [13] https://www.allianzsp.sk/sk_SK/sukromne-osoby/zivot/stastny-zivot.html#
[cit. 11.07.2021]
- [14] https://www.allianzsp.sk/sk_SK/sukromne-osoby/zivot/flexi-zivot.html#
[cit. 11.07.2021]
- [15] <https://www.csob.sk/zivotne-poistenie/vital> [cit. 11.07.2021]
- [16] <https://www.generalisk.sk/pre-obcanov/zivotne-poistenie/detske-poistenie/>
[cit. 11.07.2021]
- [17] <https://www.metlife.sk/individualne-poistenie/zivotne-poistenie/trendlife/>
[cit. 11.07.2021]
- [18] <https://poistovna.nn.sk/nn-partner/#tab1> [cit. 12.07.2021]
- [19] https://www.allianzsp.sk/sk_SK/sukromne-osoby/zivot/moja-farbicka.html#
[cit. 12.07.2021]
- [20] [https://www.metlife.sk/skupinove-poistenie/skupinove-zivotne-alebo-urazove-
poistenie/](https://www.metlife.sk/skupinove-poistenie/skupinove-zivotne-alebo-urazove-poistenie/) [cit. 12.07.2021]
- [21] <https://www.prophet-web.com/about-us/> [cit. 28.07.2021]
- [22] <https://ivit.sk/blog/styri-hlavne-dovody-preco-sa-naucit-jazyk-sql/> [cit. 28.07.2021]

[23] <https://www.learn2code.sk/blog/studijny-plan-pre-buduceho-java-programatora>
[cit. 29.07.2021]

[24] [https://sk.wikipedia.org/wiki/C_\(programovac%C3%AD_jazyk\)](https://sk.wikipedia.org/wiki/C_(programovac%C3%AD_jazyk)) [cit. 29.07.2021]

[25] [https://sk.wikipedia.org/wiki/Python_\(programovac%C3%AD_jazyk\)](https://sk.wikipedia.org/wiki/Python_(programovac%C3%AD_jazyk))
[cit. 31.07.2021]

[26] https://en.wikipedia.org/wiki/SAS_language [cit. 31.07.2021]

[27] [https://sk.wikipedia.org/wiki/R_\(programovac%C3%AD_jazyk\)](https://sk.wikipedia.org/wiki/R_(programovac%C3%AD_jazyk)) [cit. 13.03.2021]

[28] <http://www.hpi.sk/2009/08/mortalita/> [cit. 13.03.2021]

[29] <https://www.ecb.europa.eu/ecb/educational/hicp/html/index.sk.html> [cit. 13.03.2021]

PRÍLOHY

Príloha A

Zdrojový kód na výpočet jednostranných dôchodkov

```
jednostranne<-  
function(tableslist1,tableslist2,vek1,vek2,status1="joint",  
status2="joint",i,m)  
{library(lifecontingencies)  
if(length(vek1)==1 & length(vek2)==1)  
{  
dochodok <- axyzn(tablesList=tableslist2, x=vek2, i=i, m=m,  
k=1,          status=status2,          type="EV",          power=1,  
payment="immediate") -  
axyzn(tablesList=c(tableslist1,tableslist2), x=c(vek1,vek2),  
i=i,    k=1,    status=status1,m=m,    type="EV",    power=1,  
payment="immediate")  
}  
else if (length(vek2)==1 & length(vek1)>1 & status1=="joint")  
{  
dochodok <- axyzn(tablesList=tableslist2, x=vek2, i=i, m=m,  
k=1,          status=status2,          type="EV",          power=1,  
payment="immediate") -  
axyzn(tablesList=c(tableslist1,tableslist2), x=c(vek1,vek2),  
i=i,    k=1,    status=status1,    m=m,    type="EV",    power=1,  
payment="immediate")  
}  
else if (length(vek2)>1 & length(vek1)==1 & status2=="joint")  
{  
dochodok <- axyzn(tablesList=tableslist2, x=vek2, i=i, m=m,  
k=1,          status=status2,          type="EV",          power=1,  
payment="immediate") -
```

```

axyzn(tablesList=c(tableslist1,tableslist2), x=c(vek1,vek2),
i=i, k=1, status=status1, m=m, type="EV", power=1,
payment="immediate")
}
else if (length(vek2)==1 & length(vek1)>1 & status1=="last")
{
if(length(vek1)==2)
{
dochodok <- axyzn(tablesList=tableslist2, x=vek2, i=i, m=m,
k=1, status=status2, type="EV", power=1,
payment="immediate")-
(axyzn(tablesList=c(tableslist1[1],tableslist2),
x=c(vek1[1],vek2), i=i, k=1, status="joint", m=m, type="EV",
power=1, payment="immediate")
+axyzn(tablesList=c(tableslist1[2],tableslist2),
x=c(vek1[2],vek2), i=i, k=1, status="joint", m=m, type="EV",
power=1,payment="immediate")
-axyzn(tablesList=c(tableslist1,tableslist2), x=c(vek1,vek2),
i=i, k=1, status="joint", m=m, type="EV", power=1,
payment="immediate"))
}
else if(length(vek1)==3)
{
dochodok <- axyzn(tablesList=tableslist2, x=vek2, i=i, m=m,
k=1, status=status2, type="EV", power=1, payment="immediate")
-(axyzn(tablesList=c(tableslist1[1],tableslist2),
x=c(vek1[1],vek2), i=i, k=1, status="joint", m=m, type="EV",
power=1, payment="immediate")
+axyzn(tablesList=c(tableslist1[2],tableslist2),
x=c(vek1[2],vek2), i=i, k=1, status="joint", m=m, type="EV",
power=1, payment="immediate")
+axyzn(tablesList=c(tableslist1[3],tableslist2),
x=c(vek1[3],vek2), i=i, k=1, status="joint", m=m, type="EV",
power=1, payment="immediate")

```

```

-axyzn(tablesList=c(tableslist1[1],tableslist1[2],
tableslist2),x=c(vek1[1],vek1[2],vek2),i=i,k=1,
status="joint", m=m, type="EV", power=1, payment="immediate")
-axyzn(tablesList=c(tableslist1[1],tableslist1[3],
tableslist2),x=c(vek1[1],vek1[3],vek2),i=i,k=1,
status="joint", m=m, type="EV", power=1, payment="immediate")
-axyzn(tablesList=c(tableslist1[2],tableslist1[3],
tableslist2),      x=c(vek1[2],vek1[3],vek2),      i=i,      k=1,
status="joint",m=m,type="EV",power=1, payment="immediate")
+axyzn(tablesList=c(tableslist1,tableslist2), x=c(vek1,vek2),
i=i,      k=1,      status="joint",      m=m,      type="EV",      power=1,
payment="immediate"))
}else if(length(vek1)==4)
{
dochodok <- axyzn(tablesList=tableslist2, x=vek2, i=i, m=m,
k=1, status=status2, type="EV", power=1, payment="immediate")
-(axyzn(tablesList=c(tableslist1[1],tableslist2),
x=c(vek1[1],vek2), i=i, k=1, status="joint", m=m, type="EV",
power=1, payment="immediate")
+axyzn(tablesList=c(tableslist1[2],tableslist1),
x=c(vek1[2],vek2), i=i, k=1, status="joint", m=m, type="EV",
power=1, payment="immediate")
+axyzn(tablesList=c(tableslist1[3],tableslist2),
x=c(vek1[3],vek2), i=i, k=1, status="joint", m=m, type="EV",
power=1, payment="immediate")
+axyzn(tablesList=c(tableslist1[4],tableslist2),
x=c(vek1[4],vek2), i=i, k=1, status="joint", m=m, type="EV",
power=1, payment="immediate")
-axyzn(tablesList=c(tableslist1[1],tableslist1[2],
tableslist2),      x=c(vek1[1],vek1[2],vek2),      i=i,      k=1,
status="joint", m=m, type="EV", power=1, payment="immediate")
-axyzn(tablesList=c(tableslist1[1],tableslist1[3],
tableslist2),x=c(vek1[1],vek1[3],vek2),i=i,k=1,
status="joint", m=m, type="EV", power=1, payment="immediate")

```

```

-axyzn(tablesList=c(tableslist1[1],tableslist1[4],
tableslist2),      x=c(vek1[1],vek1[4],vek2),      i=i,      k=1,
status="joint", m=m, type="EV", power=1, payment="immediate")
-axyzn(tablesList=c(tableslist1[2],tableslist1[3],
tableslist2),      x=c(vek1[2],vek1[3],vek2),      i=i,      k=1,
status="joint", m=m, type="EV", power=1, payment="immediate")
-axyzn(tablesList=c(tableslist1[2],tableslist1[4],
tableslist2),      x=c(vek1[2],vek1[4],vek2),      i=i,      k=1,
status="joint", m=m, type="EV", power=1, payment="immediate")
-axyzn(tablesList=c(tableslist1[3],tableslist1[4],
tableslist2),      x=c(vek1[3],vek1[4],vek2),      i=i,      k=1,
status="joint", m=m, type="EV", power=1, payment="immediate")
+axyzn(tablesList=c(tableslist1[1],tableslist1[2],tableslist1
[3],tableslist2),  x=c(vek1[1],vek1[2],vek1[3],vek2),  i=i,
k=1,      status="joint",      m=m,      type="EV",      power=1,
payment="immediate")
+axyzn(tablesList=c(tableslist1[1],tableslist1[3],
tableslist1[4],tableslist2), x=c(vek1[1],vek1[3],vek1[4],
vek2), i=i, k=1, status="joint", m=m, type="EV", power=1,
payment="immediate")
+axyzn(tablesList=c(tableslist1[2],tableslist1[3],tableslist1
[4],tableslist2),  x=c(vek1[2],vek1[3],vek1[4],vek2),  i=i,
k=1,      status="joint",      m=m,      type="EV",      power=1,
payment="immediate")
+axyzn(tablesList=c(tableslist1[1],tableslist1[2],
tableslist1[4],tableslist2),
x=c(vek1[1],vek1[2],vek1[4],vek2), i=i, k=1, status="joint",
m=m, type="EV", power=1, payment="immediate")
-axyzn(tablesList=c(tableslist1,tableslist2), x=c(vek1,vek2),
i=i, k=1, status="joint", m=m, type="EV", power=1,
payment="immediate"))
}
}
else if (length(vek1)==1 & length(vek2)>1 & status2=="last")

```

```

{
if(length(vek1)==2)
{
dochodok <- axyzn(tablesList=tableslist2, x=vek2, i=i, m=m,
k=1, status=status2, type="EV", power=1, payment="immediate")
-(axyzn(tablesList=c(tableslist1,tableslist2[1]),
x=c(vek1,vek2[1]), i=i, k=1, status="joint", m=m, type="EV",
power=1, payment="immediate")
+axyzn(tablesList=c(tableslist1,tableslist2[2]),
x=c(vek1,vek2[2]), i=i, k=1, status="joint", m=m, type="EV",
power=1, payment="immediate")
-axyzn(tablesList=c(tableslist1,tableslist2), x=c(vek1,vek2),
i=i, k=1, status="joint", m=m, type="EV", power=1,
payment="immediate"))
}
else if(length(vek1)==3)
{
dochodok <- axyzn(tablesList=tableslist2, x=vek2, i=i, m=m,
k=1, status=status2, type="EV", power=1, payment="immediate")
-(axyzn(tablesList=c(tableslist1,tableslist2[1]),
x=c(vek1,vek2[1]), i=i, k=1, status="joint", m=m, type="EV",
power=1, payment="immediate")
+axyzn(tablesList=c(tableslist1,tableslist2[2]),
x=c(vek1,vek2[2]), i=i, k=1, status="joint", m=m, type="EV",
power=1, payment="immediate")
+axyzn(tablesList=c(tableslist1,tableslist2[3]),
x=c(vek1,vek2[3]), i=i, k=1, status="joint", m=m, type="EV",
power=1, payment="immediate")
-axyzn(tablesList=c(tableslist2[1],tableslist2[2],
tableslist1), x=c(vek2[1],vek2[2],vek1), i=i, k=1,
status="joint", m=m, type="EV", power=1, payment="immediate")
-
axyzn(tablesList=c(tableslist2[1],tableslist2[3],tableslist2)

```

```

, x=c(vek2[1],vek2[3],vek1), i=i, k=1, status="joint", m=m,
type="EV", power=1, payment="immediate")
-axyzn(tablesList=c(tableslist2[2],tableslist2[3],
tableslist1), x=c(vek2[2],vek2[3],vek1), i=i, k=1,
status="joint", m=m, type="EV", power=1, payment="immediate")
+axyzn(tablesList=c(tableslist1,tableslist2), x=c(vek1,vek2),
i=i, k=1, status="joint", m=m, type="EV", power=1,
payment="immediate"))
}
else if(length(vek1)==4)
{
dochodok <- axyzn(tablesList=tableslist2, x=vek2, i=i, m=m,
k=1, status=status2, type="EV", power=1, payment="immediate")
-(axyzn(tablesList=c(tableslist2[1],tableslist1),
x=c(vek2[1],vek1), i=i, k=1, status="joint", m=m, type="EV",
power=1, payment="immediate")
+axyzn(tablesList=c(tableslist2[2],tableslist1),
x=c(vek2[2],vek1), i=i, k=1, status="joint", m=m, type="EV",
power=1, payment="immediate")
+axyzn(tablesList=c(tableslist2[3],tableslist1),
x=c(vek2[3],vek1), i=i, k=1, status="joint", m=m, type="EV",
power=1, payment="immediate")
+axyzn(tablesList=c(tableslist2[4],tableslist1),
x=c(vek2[4],vek1), i=i, k=1, status="joint", m=m, type="EV",
power=1, payment="immediate")
-
axyzn(tablesList=c(tableslist2[1],tableslist2[2],tableslist1)
, x=c(vek2[1],vek2[2],vek1), i=i, k=1, status="joint", m=m,
type="EV", power=1, payment="immediate")
-axyzn(tablesList=c(tableslist2[1],tableslist2[3],
tableslist1), x=c(vek2[1],vek2[3],vek1), i=i, k=1,
status="joint", m=m, type="EV", power=1, payment="immediate")
-
axyzn(tablesList=c(tableslist2[1],tableslist2[4],tableslist1)

```

```

, x=c(vek2[1],vek2[4],vek1), i=i, k=1, status="joint", m=m,
type="EV", power=1, payment="immediate")
-axyzn(tablesList=c(tableslist2[2],tableslist2[3],
tableslist1), x=c(vek2[2],vek2[3],vek1), i=i, k=1,
status="joint", m=m, type="EV", power=1, payment="immediate")
-axyzn(tablesList=c(tableslist2[2],tableslist2[4],
tableslist1), x=c(vek2[2],vek2[4],vek1), i=i, k=1,
status="joint", m=m, type="EV", power=1, payment="immediate")
-axyzn(tablesList=c(tableslist2[3],tableslist2[4],
tableslist1), x=c(vek2[3],vek2[4],vek1), i=i, k=1,
status="joint", m=m, type="EV", power=1, payment="immediate")
+axyzn(tablesList=c(tableslist2[1],tableslist2[2],tableslist2
[3],tableslist1), x=c(vek2[1],vek2[2],vek2[3],vek1), i=i,
k=1, status="joint", m=m, type="EV", power=1,
payment="immediate")
+axyzn(tablesList=c(tableslist2[1],tableslist2[3],tableslist2
[4],tableslist1), x=c(vek2[1],vek2[3],vek2[4],vek1), i=i,
k=1, status="joint", m=m, type="EV", power=1,
payment="immediate")
+axyzn(tablesList=c(tableslist2[2],tableslist2[3],tableslist2
[4],tableslist1), x=c(vek2[2],vek2[3],vek2[4],vek1), i=i,
k=1, status="joint", m=m, type="EV", power=1,
payment="immediate")
+axyzn(tablesList=c(tableslist2[1],tableslist2[2],tableslist2
[4],tableslist1), x=c(vek2[1],vek2[2],vek2[4],vek1), i=i,
k=1, status="joint", m=m, type="EV", power=1,
payment="immediate")
-axyzn(tablesList=c(tableslist1,tableslist2), x=c(vek1,vek2),
i=i, k=1, status="joint", m=m, type="EV", power=1,
payment="immediate"))
}}
dochodok
}

```

Príloha B

Zdrojový kód na výpočet pravdepodobnosti:

```
npxyzr <- function(tableslist, ages, n, r, probability)

{

library(lifecontingencies)

m <- length(ages)

Z1<-0;Z2<-0;Z3<-0;Z4<-0;Z5<-0;Z6<-0;Z7<-0;Z8<-0;Z9<-0;Z10<-
0;prob<-0;ins<-0;v<-0

if(probability=="exactly")

{

if(length(ages)==2)

{

Z1 <-
pxt(tableslist[[1]],ages[1],t=n)+pxt(tableslist[[2]],ages[2],
t=n)

Z2 <- pxyzr(tableslist,ages,t=n)

Z <- c(Z1,Z2)

}

if(length(ages)==3)

{

Z1 <-
pxt(tableslist[[1]],ages[1],t=n)+pxt(tableslist[[2]],ages[2],
t=n)+pxt(tableslist[[3]],ages[3],t=n)

Z2 <-
pxyzr(c(tableslist[[1]],tableslist[[2]]),c(ages[1],ages[2]),
```

```

t=n)+pxyzt(c(tableslist[[1]],tableslist[[3]]),c(ages[1],
ages[3]),t=n)+pxyzt(c(tableslist[[2]],tableslist[[3]]),
c(ages[2],ages[3]),t=n)
Z3 <- pxyzt(tableslist,ages,t=n)
Z <- c(Z1,Z2,Z3)
}

if(length(ages)==4)
{
Z1 <-
pxt(tableslist[[1]],ages[1],t=n)+pxt(tableslist[[2]],ages[2],
t=n)+pxt(tableslist[[3]],ages[3],t=n)+pxt(tableslist[[4]],
ages[4],t=n)
Z2 <-
pxyzt(c(tableslist[[1]],tableslist[[2]]),c(ages[1],ages[2]),
t=n)+pxyzt(c(tableslist[[1]],tableslist[[3]]),c(ages[1],
ages[3]),t=n)+pxyzt(c(tableslist[[2]],tableslist[[3]]),
c(ages[2],ages[3]),t=n)+pxyzt(c(tableslist[[1]],
tableslist[[4]]),c(ages[1],ages[4]),t=n)
+pxyzt(c(tableslist[[2]],tableslist[[4]]),c(ages[2],ages[4]),
t=n)+pxyzt(c(tableslist[[3]],tableslist[[4]]),c(ages[3],
ages[4]),t=n)
Z3 <-
pxyzt(c(tableslist[[1]],tableslist[[2]],tableslist[[3]]),
c(ages[1],ages[2],ages[3]),t=n)+pxyzt(c(tableslist[[1]],
tableslist[[2]],tableslist[[4]]),c(ages[1],ages[2],ages[4]),

```

```

t=n)+pxyzt(c(tableslist[[1]],tableslist[[3]],tableslist[[4]]),
,c(ages[1],ages[3],ages[4]),t=n)+pxyzt(c(tableslist[[2]],
tableslist[[3]],tableslist[[4]]),c(ages[2],ages[3],ages[4]),
t=n)

Z4 <- pxyzt(tableslist,ages,t=n)

Z <- c(Z1,Z2,Z3,Z4)

}

if(length(ages)==5)

{

Z1 <-
pxt(tableslist[[1]],ages[1],t=n)+pxt(tableslist[[2]],ages[2],
t=n)+pxt(tableslist[[3]],ages[3],t=n)+pxt(tableslist[[4]],
ages[4],t=n)+pxt(tableslist[[5]],ages[5],t=n)

Z2 <-
pxyzt(c(tableslist[[1]],tableslist[[2]]),c(ages[1],ages[2]),
t=n)+pxyzt(c(tableslist[[1]],tableslist[[3]]),c(ages[1],
ages[3]),t=n)+pxyzt(c(tableslist[[2]],tableslist[[3]]),
c(ages[2],ages[3]),t=n)+pxyzt(c(tableslist[[1]],
tableslist[[4]]),c(ages[1],ages[4]),t=n)
+pxyzt(c(tableslist[[2]],tableslist[[4]]),c(ages[2],ages[4]),
t=n)+pxyzt(c(tableslist[[3]],tableslist[[4]]),
c(ages[3],ages[4]),t=n)+pxyzt(c(tableslist[[1]],
tableslist[[5]]),c(ages[1],ages[5]),t=n)
+pxyzt(c(tableslist[[2]],tableslist[[5]]),c(ages[2],ages[5]),
t=n)+pxyzt(c(tableslist[[3]],tableslist[[5]]),c(ages[3],
ages[5]),t=n)+pxyzt(c(tableslist[[4]],tableslist[[5]]),

```

```

c(ages[4],ages[5]),t=n)

Z3 <-
pxyzt(c(tableslist[[1]],tableslist[[2]],tableslist[[3]]),
c(ages[1],ages[2],ages[3]),t=n)+pxyzt(c(tableslist[[1]],
tableslist[[2]],tableslist[[4]]),c(ages[1],ages[2],ages[4]),
t=n)+pxyzt(c(tableslist[[1]],tableslist[[3]],
tableslist[[4]]),c(ages[1],ages[3],ages[4]),t=n)
+pxyzt(c(tableslist[[2]],tableslist[[3]],tableslist[[4]]),
c(ages[2],ages[3],ages[4]),t=n)+pxyzt(c(tableslist[[1]],
tableslist[[2]],tableslist[[5]]),c(ages[1],ages[2],ages[5]),
t=n)+pxyzt(c(tableslist[[1]],tableslist[[3]],
tableslist[[5]]),c(ages[1],ages[3],ages[5]),t=n)
+pxyzt(c(tableslist[[1]],tableslist[[4]],tableslist[[5]]),
c(ages[1],ages[4],ages[5]),t=n) +
pxyzt(c(tableslist[[2]],tableslist[[3]],tableslist[[5]]),
c(ages[2],ages[3],ages[5]),t=n)+pxyzt(c(tableslist[[2]],
tableslist[[4]],tableslist[[5]]),c(ages[2],ages[4],ages[5]),
t=n)+pxyzt(c(tableslist[[3]],tableslist[[4]],
tableslist[[5]]),c(ages[3],ages[4],ages[5]),t=n)

Z4 <-
pxyzt(c(tableslist[[1]],tableslist[[2]],tableslist[[3]],
tableslist[[4]]),c(ages[1],ages[2],ages[3],ages[4]),t=n)
+pxyzt(c(tableslist[[1]],tableslist[[3]],tableslist[[4]],
tableslist[[5]]),c(ages[1],ages[3],ages[4],ages[5]),t=n)
+pxyzt(c(tableslist[[1]],tableslist[[2]],tableslist[[4]],
tableslist[[5]]),c(ages[1],ages[2],ages[4],ages[5]),t=n)

```

```

+pxyzt(c(tableslist[[2]],tableslist[[3]],tableslist[[4]],
tableslist[[5]]),c(ages[2],ages[3],ages[4],ages[5]),t=n)
+pxyzt(c(tableslist[[1]],tableslist[[2]],tableslist[[3]],
tableslist[[5]]),c(ages[1],ages[2],ages[3],ages[5]),t=n)
Z5 <- pxyzt(tableslist,ages,t=n)

Z <- c(Z1,Z2,Z3,Z4,Z5)

}

for (i in 1:(m-r))

{

ins <- ins + ((-1)^(i))*Z[r+i]*choose(r+i,i)

if(length(ages)==r)

{Z[r]<- pxyzt(tableslist, ages, status="joint",t=n)

ins <- 0

}

}

else if(probability=="atleast")

{

if(length(ages)==2)

{

Z1 <-

pxt(tableslist[[1]],ages[1],t=n)+pxt(tableslist[[2]],ages[2],

t=n)

Z2 <- pxyzt(tableslist,ages,t=n)

Z <- c(Z1,Z2) }

if(length(ages)==3)

```

```

{
Z1 <-
pxt(tableslist[[1]],ages[1],t=n)+pxt(tableslist[[2]],ages[2],
t=n)+pxt(tableslist[[3]],ages[3],t=n)

Z2 <-
pxyzt(c(tableslist[[1]],tableslist[[2]]),c(ages[1],ages[2]),
t=n)+pxyzt(c(tableslist[[1]],tableslist[[3]]),c(ages[1],
ages[3]),t=n)+pxyzt(c(tableslist[[2]],tableslist[[3]]),
c(ages[2],ages[3]),t=n)

Z3 <- pxyzt(tableslist,ages,t=n)

Z <- c(Z1,Z2,Z3)

}

if(length(ages)==4)

{

Z1 <-
pxt(tableslist[[1]],ages[1],t=n)+pxt(tableslist[[2]],ages[2],
t=n)+pxt(tableslist[[3]],ages[3],t=n)+pxt(tableslist[[4]],
ages[4],t=n)

Z2 <-
pxyzt(c(tableslist[[1]],tableslist[[2]]),c(ages[1],ages[2]),
t=n)+pxyzt(c(tableslist[[1]],tableslist[[3]]),c(ages[1],
ages[3]),t=n)+pxyzt(c(tableslist[[2]],tableslist[[3]]),
c(ages[2],ages[3]),t=n)+pxyzt(c(tableslist[[1]],
tableslist[[4]]),c(ages[1],ages[4]),t=n)

+pxyzt(c(tableslist[[2]],tableslist[[4]]),c(ages[2],ages[4]),
t=n)+pxyzt(c(tableslist[[3]],tableslist[[4]]),c(ages[3],
ages[4]),t=n)

```

```

Z3 <-
pxyzt(c(tableslist[[1]],tableslist[[2]],tableslist[[3]]),
c(ages[1],ages[2],ages[3]),t=n)+pxyzt(c(tableslist[[1]],
tableslist[[2]],tableslist[[4]]),c(ages[1],ages[2],ages[4]),
t=n)+pxyzt(c(tableslist[[1]],tableslist[[3]],
tableslist[[4]]),c(ages[1],ages[3],ages[4]),t=n)
+pxyzt(c(tableslist[[2]],tableslist[[3]],tableslist[[4]]),
c(ages[2],ages[3],ages[4]),t=n)

Z4 <- pxyzt(tableslist,ages,t=n)

Z <- c(Z1,Z2,Z3,Z4)

}

if(length(ages)==5)

{Z1 <-
pxt(tableslist[[1]],ages[1],t=n)+pxt(tableslist[[2]],ages[2],
t=n)+pxt(tableslist[[3]],ages[3],t=n)+pxt(tableslist[[4]],
ages[4],t=n)+pxt(tableslist[[5]],ages[5],t=n)

Z2 <-
pxyzt(c(tableslist[[1]],tableslist[[2]]),c(ages[1],ages[2]),
t=n)+pxyzt(c(tableslist[[1]],tableslist[[3]]),c(ages[1],
ages[3]),t=n)+pxyzt(c(tableslist[[2]],tableslist[[3]]),
c(ages[2],ages[3]),t=n)+pxyzt(c(tableslist[[1]],
tableslist[[4]]),c(ages[1],ages[4]),t=n)
+pxyzt(c(tableslist[[2]],tableslist[[4]]),c(ages[2],ages[4]),
t=n)+pxyzt(c(tableslist[[3]],tableslist[[4]]),c(ages[3],
ages[4]),t=n)+pxyzt(c(tableslist[[1]],tableslist[[5]]),
c(ages[1],ages[5]),t=n)+pxyzt(c(tableslist[[2]],

```

```

tableslist[[5]]),c(ages[2],ages[5]),t=n)

+pxyzt(c(tableslist[[3]],tableslist[[5]]),c(ages[3],ages[5]),
t=n)+pxyzt(c(tableslist[[4]],tableslist[[5]]),c(ages[4],
ages[5]),t=n)

```

```

Z3 <-
pxyzt(c(tableslist[[1]],tableslist[[2]],tableslist[[3]]),
c(ages[1],ages[2],ages[3]),t=n)+pxyzt(c(tableslist[[1]],
tableslist[[2]],tableslist[[4]]),c(ages[1],ages[2],ages[4]),
t=n)+pxyzt(c(tableslist[[1]],tableslist[[3]],
tableslist[[4]]),c(ages[1],ages[3],ages[4]),t=n)
+pxyzt(c(tableslist[[2]],tableslist[[3]],tableslist[[4]]),
c(ages[2],ages[3],ages[4]),t=n)+pxyzt(c(tableslist[[1]],
tableslist[[2]],tableslist[[5]]),c(ages[1],ages[2],ages[5]),
t=n)+pxyzt(c(tableslist[[1]],tableslist[[3]],
tableslist[[5]]),c(ages[1],ages[3],ages[5]),t=n)
+pxyzt(c(tableslist[[1]],tableslist[[4]],tableslist[[5]]),
c(ages[1],ages[4],ages[5]),t=n) +
pxyzt(c(tableslist[[2]],tableslist[[3]],tableslist[[5]]),
c(ages[2],ages[3],ages[5]),t=n)+pxyzt(c(tableslist[[2]],
tableslist[[4]],tableslist[[5]]),c(ages[2],ages[4],ages[5]),
t=n)+pxyzt(c(tableslist[[3]],tableslist[[4]],
tableslist[[5]]),c(ages[3],ages[4],ages[5]),t=n)

```

```

Z4 <-
pxyzt(c(tableslist[[1]],tableslist[[2]],tableslist[[3]],
tableslist[[4]]),c(ages[1],ages[2],ages[3],ages[4]),t=n)
+pxyzt(c(tableslist[[1]],tableslist[[3]],tableslist[[4]],

```

```

tableslist[[5]]),c(ages[1],ages[3],ages[4],ages[5]),t=n)
+pxyzt(c(tableslist[[1]],tableslist[[2]],tableslist[[4]],
tableslist[[5]]),c(ages[1],ages[2],ages[4],ages[5]),t=n)
+pxyzt(c(tableslist[[2]],tableslist[[3]],tableslist[[4]],
tableslist[[5]]),c(ages[2],ages[3],ages[4],ages[5]),t=n)
+pxyzt(c(tableslist[[1]],tableslist[[2]],tableslist[[3]],
tableslist[[5]]),c(ages[1],ages[2],ages[3],ages[5]),t=n)
Z5 <- pxyzt(tableslist,ages,t=n)

Z <- c(Z1,Z2,Z3,Z4,Z5)

}

for (i in 0:(m-r-1))
{
ins <- ins + ((-1)^(i+1))*Z[r+i+1]*choose(r+i,i+1)}

if(length(ages)==r)
{
Z[r]<- pxyzt(tableslist, ages, status="joint",t=n)

ins <- 0

}

if(r==1)
{
Z[r]<- pxyzt(tableslist, ages, status="last",t=n)

ins <- 0

}

}

prob <- Z[r] + ins

```

prob

}