Journal of
Actuarial Research
(JAR)

On the Automated Generation of Life Contingencies Problems and Solutions

Journal of Actuarial Science
ISSN: 2958-0595 (Online)
Vol. 2, Issue No. 2, pp 1 - 20, 2024 www.carijournals.org

On the Automated Generation of Life Contingencies Problems and Solutions

[1]Chris Groendyke
[1]Professor, Department of Mathematics, School of Data Intelligence and Technology
Robert Morris University
https://orcid.org/0000-0001-9635-8397

Accepted: 14th June, 2024, Received in Revised Form: 2nd July, 2024, Published: 16th July, 2024

Abstract

Purpose: The purpose of this project is to provide an open source solution to this problem that is customizable, efficient, and easy to use.

Methodology: We introduce an open-source software package named **AutoSULT** to address this problem. This package utilizes the computing abilities of R and the typesetting abilities of R Markdown, LaTeX, and other existing R packages.

Findings: Our software efficiently generates and typesets customized questions with step-by-step solutions for several types of problems that are commonly used in introductory life contingencies courses. We highlight some of the advantages of this tool and demonstrate its uses.

Unique contribution to theory, practice and policy: We believe that this software can serve as a valuable resource for both instructors and students of introductory life contingencies courses.

Keywords: *Life Contingencies; Life Tables; Automatic Problem Generation*

Journal of Actuarial Science
ISSN: 2958-0595 (Online)
Vol. 2, Issue No. 2, pp 1 - 20, 2024 www.carijournals.org

1. INTRODUCTION

As the capabilities of artificial intelligence and other related technologies have greatly increased in recent years, many aspects of education have benefitted from greater automation. These span a vast spectrum of fields, from analytical chemistry (Cocovi-Solberg & Miró, 2015) to computer science (Combéfis, 2022), to name just a couple. Kurdi et al. (2020) provides a review of recent literature on the topic. There are some areas, however, that have not seen as great of a transformation due to this increase in automation, one of these being in the area of basic life contingencies education.

A key component of many introductory-level life contingencies classes involves computing various quantities from the values given in some standard life table. These tables typically have some fixed set of calculated values and rely on the user to be able to calculate other required values using a set of basic relationships between the quantities. Because these relationships are fundamental to learning life contingencies, it is important that students in these courses have a firm understanding of them, which is most typically gained through the completion of a large number of problems, ideally with detailed solutions against which the students can check their work.

The actuarial profession has developed a complex system of notation for many of its fundamental concepts, including survival probabilities and expected present values of life insurance and annuity products. While this system has its own internal logic, it often follows somewhat different rules than those seen in related fields, in terms of how symbols (especially subscripts and superscripts, but also groupings of subsymbols) are placed and grouped. As a result, standard mainstream software applications, as well as even many mathematically and scientifically oriented packages, often struggle to typeset these symbols efficiently, or sometimes even at all.

Given the two considerations above, there is a clear need for an open-source software package that can automatically generate an arbitrary number of basic life contingencies problems (including step-by-step solutions), typeset in an organized, nicely formatted manner. The R package **AutoSULT** fills this niche with a series of functions that use R Markdown (Allaire et al., 2023) to render PDF files with randomly generated problems (and accompanying solutions) of types selected by the user. The target audience for this software is instructors of university courses covering life contingencies concepts. Instructors could generate (perhaps individualized) homework assignments, quizzes, or problem sets for use in problem-solving, recitation, or tutorial sessions, without having to

Journal of Actuarial Science
ISSN: 2958-0595 (Online)
Vol. 2, Issue No. 2, pp 1 - 20, 2024 www.carijournals.org

invest the time to learn all of the corresponding LaTeX code for life contingencies symbols, or to manually typeset each solution.

This is not intended to be a tool for exam preparation for the actuarial professional credentialing exams; there are many high-quality commercial products filling that role. Rather, the goal is to provide an open-source solution to generate and typeset problems and solutions covering basic types of life table calculations, as a time-saving tool for life contingencies instructors.

The remainder of this paper is organized as follows: Section 2 discusses some selected components of actuarial notation; Section 3 describes some standard actuarial tables and associated considerations for teaching introductory life contingencies courses; Section 4 introduces a tool to automate and simplify some of the tasks faced by life contingencies instructors; and finally, Section 5 offers some concluding remarks.

2. ACTUARIAL NOTATION

Like many disciplines, actuarial science has developed its own unique set of notations to describe frequently used concepts and quantities. Because our focus is on life contingencies courses, we restrict our attention here to notation dealing with quantities commonly calculated in such courses; notation for quantities associated with casualty and other types of short-term insurances is beyond the scope of this paper. In particular, these introductory life contingencies courses often deal with survival and mortality probabilities, life expectancies, and expected present values of life-contingent instruments such as life insurance and annuities, premiums, and reserves.

The history of this actuarial notation dates back at least 180 years with David Jones introducing life insurance notation (Jones, 1843), and George King later expanding it (King, 1889). Attempts at standardizing the notation started soon afterwards -- and continue to the present, see Louca (2020) -- with the Second International Actuarial Congress adopting in 1898 a “Universal Actuarial Notation” that would become known as “International Actuarial Notation” (Layton and Layton, 1899).

Appendix 4 of Bowers et al. (1997, pp. 693 - 700) contains a thorough description of the architecture of actuarial symbols. The symbols we utilize are constructed by starting with a given principal symbol, which usually specifies the nature of the probability or quantity being calculated. Then accompanying symbols, embellishments, or accents are added in various positions around the principal symbol. Both the nature and position of the accompaniments dictate their meaning; they may relate to, for example, product term lengths, age of the insured(s), and payment frequency and timing. Figure 1, based on a similar figure in Bowers et al. (1997) shows the relative positions of the principal symbol (box I in the center) and the accompanying markings (boxes II – VI).

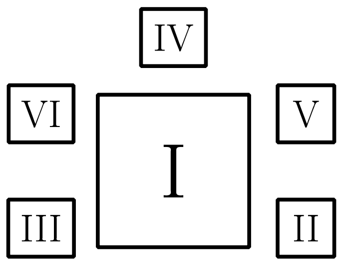

Figure 1: Diagram of actuarial symbol construction.

A few examples to illustrate such constructions are given in Figure 2. We see here that within some of these positions, it is possible for the accompanying symbols to also have further adornments. Here, the number 1 in the rightmost symbol indicates a specific order of events that must occur to trigger an insurance payment.

$${}_{10}q_{xy} \qquad \ddot{a}_{65:\overline{n}|} \qquad {}^{2}A_{45}^{(m)} \qquad A_{\overset{1}{\overline{30:40}}:\overline{20}|}$$

Figure 2: Examples of common actuarial symbols.

As the insurance industry has advanced and the products sold by insurance companies have become more complex, actuarial notation has also evolved in multiple ways. First, some notations have fallen out of favor and are no longer as widely used as they once were. One example of this is the set of special notations for net premiums and net reserves, which are more complex variations on some of the symbols above. While these were commonly found in older, traditional life contingencies textbooks, some newer texts (Dickson et al., 2019) drop the more complex notation for these quantities, not finding it useful in practice. On the other hand, as insurance companies have begun to offer increasingly complex products, the models needed to price and provision for them have also increased in complexity, and the notation associated with these models has then reflected this complexity. For example, many long-term health products are modelled by Markov or semi-Markov multiple state models. The notation for these models must additionally incorporate information regarding current and future model states. While the notation for these models is not as standardized as it is for the more traditional products, Dickson et al.

Journal of Actuarial Science
ISSN: 2958-0595 (Online)
Vol. 2, Issue No. 2, pp 1 - 20, 2024 www.carijournals.org

(2019) adopts a notation that generalizes parts of International Actuarial Notation. Figure 3 gives some examples of this notation; in all cases the two numbers in position V denote beginning and ending state numbers.

$$ {}_{10}p_{xy}^{\overline{11}} \qquad \ddot{a}_{65:\overline{n}|}^{00} \qquad {}^{2}A_{45}^{02} $$

Figure 3: Examples of common multiple state actuarial symbols.

While producing the notation described above, especially in large volumes, has traditionally been challenging, there have been recent developments in software (in particular, LaTeX packages) that have made the typesetting of this notation somewhat less cumbersome. In particular, the package **actuarialangle** (Goulet, 2017) allows for the production of both the "angle" notation often found in actuarial symbols such as the middle symbol in Figure 3 as well as the "roof" notation sometimes used in symbols related to joint insurance products, such as the rightmost symbol in Figure 2. More recently, the package **actuarialsymbol** (Beauchemin and Goulet, 2019) provides a wide array of commonly used actuarial symbols. However, producing a high volume of material that utilizes this notation (e.g., a large set of problems and solutions for a life contingencies course) can still be rather tedious.

3. LIFE TABLES AND THEIR USAGE IN INTRODUCTORY LIFE CONTINGENCIES COURSES

Life tables, also known as mortality tables or survival tables, are very commonly used in actuarial science to describe survival models. They track a hypothetical cohort of individuals over time, giving at regular discrete intervals the expected number remaining from the original group. Using this information, the probability of surviving from a given age to some later age can be calculated, as can other quantities such as life expectancy. For convenience, other auxiliary information is also often included in life tables, such as annual mortality probabilities, and expected present values of various types of life insurances and annuities.

Life tables with this same basic structure can be traced date back as far as the 17th century (Graunt, 1662; Halley, 1693). These types of life tables remain pervasive in actuarial practice today; they are still used in many common actuarial tasks such as pricing and reserving. As discussed in Section 3.3, life tables also remain a staple of life contingencies courses. Their tabular format makes them very easy to manipulate in spreadsheets, which is a clear benefit in both practical and pedagogical realms.

Journal of Actuarial Science
ISSN: 2958-0595 (Online)
Vol. 2, Issue No. 2, pp 1 - 20, 2024 www.carijournals.org

Similar types of tables are also commonly used in other areas of actuarial practice. For example, tables that describe frequencies and durations of sicknesses and disabilities have been used to price various types of health and disability insurance. And service tables are commonly used by actuaries to perform valuations related to pension and benefit plans.

One drawback of life tables is that they typically only provide information at discrete (often annual) intervals. For calculations requiring more granular information (such as the probability of surviving some portion of a year), the table must be supplemented by additional assumptions. In this context, we consider two of the more commonly made assumptions regarding the survival of individuals within a given year of age, namely the Uniform Distribution of Deaths (UDD) and Constant Force of Mortality (CF) assumptions. Alternatively, for expected present values (EPVs) of annuities with payments occurring more frequently than yearly, a frequently used approximation is Woolhouse's formula (Woolhouse, 1869), which is based on the Euler-Maclaurin formula. These assumptions and approximations, as well as their resulting formulas, are discussed in many introductory life contingencies texts (Bowers et al., 1997; Dickson et al., 2019; Camilli et al., 2014).

3.1 *Makeham's Law of Mortality and the Standard Ultimate Life Table*

One specific mortality model that is commonly used for educational and testing purposes is the **Standard Ultimate Survival Model (SUSM)** or **Standard Ultimate Life Table (SULT)**. This survival model is defined in Dickson et al. (2019) on p. 82 and is used extensively throughout this text in many of its examples and exercises. This life table is also currently (as of Fall 2024) used as one of the standard tables in the Society of Actuaries (SOA) Fundamentals of Actuarial Mathematics (FAM) and Advanced Long Term Actuarial Mathematics (ALTAM) exams (Society of Actuaries, 2024).

The survival model underlying the mortality table is based on the Gompertz-Makeham law of mortality (Makeham, 1860). Under this model, the force of mortality (also sometimes known as the hazard rate) at attained age x, denoted by μ_x, is given by the formula $\mu_x = A + Bc^x$. This model is flexible and useful enough to have been utilized in a variety of contexts, and has been applied to both human and non-human species (Castellanos et al., 2022). The SUSM uses the parameter values $A = 0.000222$, $B = 2.7(10^{-6})$, and $c = 1.124$.

Where an interest rate is needed, the default is an annual effective interest rate of $i = 5\%$. This interest rate is used for all of the insurance and annuity EPVs in the table. The joint life portion of the table additionally assumes independent future lifetimes of the individuals.

The layout of the individual life portion of the table is typical of many life tables, where each row represents an age x, with $20 \leq x \leq 100$, and the columns include:

- Annual survival model information indicating expected remaining size of the hypothetical cohort of lives and annual mortality rate;
- EPVs of annuities-due, for 10- and 20-year terms and whole life;
- EPVs of whole life and 10- and 20-year endowment insurances, with the former also calculated at i = 10.25%;
- Pure endowment factors for 5, 10, and 20 years.

The layout of the joint life table is similar, with the following EPVs calculated for joint lives both age x, or ages x and x + 10, with $30 \leq x \leq 80$:

- EPVs of annuities-due, for 10-year term and whole life annuities;
- EPVs of whole life insurances, calculated at both i = 5% and i = 10.25%.

Excerpts of the single life and joint life SULT tables are shown in Tables 1 and 2, respectively. These tables can be found in the **AutoSULT** package as the R data objects SULT.rda and SULTjt.rda, respectively.

Table 1: Excerpt of individual life SULT table.

x	ℓ_x	q_x	$\ddot{a}_x$	A_x	${}^2A_x$	$\ddot{a}_{x:\overline{10}\rvert}$	$A_{x:\overline{10}\rvert}$	$\ddot{a}_{x:\overline{20}\rvert}$	$A_{x:\overline{20}\rvert}$	${}_5E_x$	${}_{10}E_x$	${}_{20}E_x$
20	100000.0	0.000250	19.9664	0.04922	0.00580	8.0991	0.61433	13.0559	0.37830	0.78252	0.61224	0.37440
21	99975.0	0.000253	19.9197	0.05144	0.00614	8.0990	0.61434	13.0551	0.37833	0.78250	0.61220	0.37429
22	99949.7	0.000257	19.8707	0.05378	0.00652	8.0988	0.61434	13.0541	0.37837	0.78248	0.61215	0.37417
23	99924.0	0.000262	19.8193	0.05622	0.00694	8.0986	0.61435	13.0531	0.37842	0.78245	0.61210	0.37404
24	99897.8	0.000267	19.7655	0.05879	0.00739	8.0983	0.61437	13.0519	0.37848	0.78243	0.61205	0.37390
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
100	6248.2	0.289584	2.7156	0.87068	0.76427	2.7137	0.87078	2.7156	0.87068	0.08777	0.00136	0.00000

Table 2: Excerpt of joint life SULT table.

x	$\ddot{a}_{xx}$	A_{xx}	${}^2A_{xx}$	$\ddot{a}_{x:x:\overline{10}\rceil}$	$\ddot{a}_{x:x+10}$	$A_{x:x+10}$	${}^2A_{x:x+10}$	$\ddot{a}_{x:x+10:\overline{10}\rceil}$
30	18.8224	0.10369	0.01917	8.0844	18.1212	0.13709	0.03001	8.0747
31	18.7253	0.10832	0.02052	8.0833	17.9924	0.14322	0.03227	8.0724
32	18.6238	0.11315	0.02198	8.0821	17.8579	0.14962	0.03472	8.0698
33	18.5176	0.11821	0.02357	8.0807	17.7176	0.15630	0.03736	8.0669
34	18.4066	0.12350	0.02529	8.0792	17.5713	0.16327	0.04022	8.0636
35	18.2905	0.12902	0.02716	8.0774	17.4187	0.17054	0.04331	8.0600
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
80	6.4794	0.69146	0.50165	5.8082	4.5071	0.78538	0.63165	4.3896

3.2 . *Standard Sickness-Death Model*

The **Standard Sickness-Death (SSD) Model** (Dickson et al., 2019, pp. 289-290) is a model commonly used for teaching and examination purposes with respect to continuous-time multi-state Markov models. It can be used to model various types of long-term health coverages, such as disability income (DI) insurance.

A diagram of this three-state Markov model is shown in Figure 4, and like the SULT, it also uses an effective interest rate of $i = 5\%$ to calculate EPVs.

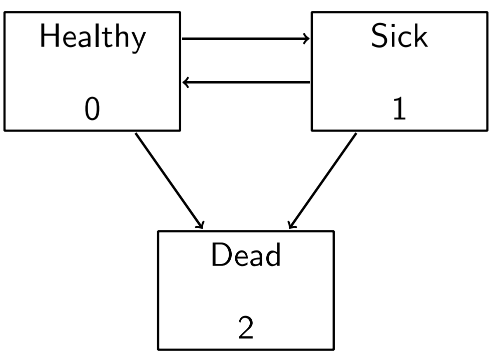

Figure 4: State Diagram for SSD Model.

The forces of transition between states for this model for an individual age x, with $50 \leq x \leq 80$, are given by:

$$\mu_x^{01} = a_1 + b_1 e^{c_1 x}, \quad a_1 = 0.0004,\ b_1 = 3.47(10^{-6}),\ c_1 = 0.138; \qquad (1)$$

$$\mu_x^{02} = a_2 + b_2 e^{c_2 x}, \quad a_2 = 0.0005,\ b_2 = 7.58(10^{-5}),\ c_2 = 0.087; \qquad (2)$$

$$\mu_x^{10} = b_1 e^{c_1(110-x)}; \quad \mu_x^{12} = 1.4\mu_x^{02}. \qquad (3)$$

This model allows an individual to move back and forth between the Healthy and Sick states, allowing for multiple sojourns in the Sick state. The table corresponding to this model (**SSD Table**) is arranged similarly to those described above, with the following values for each integer age x for $50 \leq x \leq 80$:

- EPVs of continuous state-contingent payment streams for an individual age *x;*
- EPVs of state-contingent lump sum payments;
- Probabilities of the individual being in a particular state in 10 years, given their current state at age *x*.

An excerpt of the SSD table is shown in Table 3. This table can be found in the **AutoSULT** package as the R data object SSD.rda.

Table 3: Excerpt of SSD table.

x	$\bar{a}_x^{00}$	$\bar{a}_x^{01}$	$\bar{a}_x^{11}$	$\bar{a}_x^{10}$	$\bar{A}_x^{01}$	$\bar{A}_x^{02}$	$\bar{A}_x^{10}$	$\bar{A}_x^{12}$	${}_{10}p_x^{00}$	${}_{10}p_x^{01}$	${}_{10}p_x^{11}$	${}_{10}p_x^{10}$
50	11.7454	1.9621	12.3919	0.6675	0.24144	0.33126	0.06550	0.36288	0.83936	0.06554	0.81210	0.06063
51	11.4326	2.0306	12.2393	0.5626	0.25196	0.34318	0.05702	0.37544	0.82316	0.07379	0.81016	0.05215
52	11.1135	2.0994	12.0672	0.4731	0.26284	0.35539	0.04958	0.38820	0.80533	0.08298	0.80636	0.04473
53	10.7886	2.1684	11.8777	0.3969	0.27410	0.36787	0.04307	0.40116	0.78577	0.09318	0.80078	0.03827
54	10.4582	2.2373	11.6727	0.3321	0.28574	0.38063	0.03737	0.41432	0.76433	0.10444	0.79346	0.03264
55	10.1228	2.3057	11.4542	0.2772	0.29774	0.39366	0.03240	0.42766	0.74091	0.11682	0.78447	0.02774
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

3.3. *Applications to Introductory Life Contingencies Courses*

The types of tables described above are integral to most introductory courses in life contingencies. The most fundamental concepts in these courses, such as survival probabilities and means and variances of present value random variables, relate directly to information presented in these tables. As a result, most topics, concepts, and problem types covered in these types of courses make use of these types of tables.

Indeed, one of the hallmarks of introductory life contingent courses is learning how to take the information given in a life table and use it to derive some other specified value. These types of problems are often highly standardized and their solutions are typically formulaic, making them excellent candidates for automation. The solutions also tend to build on themselves, in that problems covering more advanced topics often reference earlier material as sub-problems. In addition, because the notation -- and indeed many of the concepts -- presented in these courses are unfamiliar to the students, these students tend to benefit from a large volume of practice problems and examples. Thus, a tool to help automate the production of such problems and solutions has the potential to save a considerable amount of time and effort for the instructors of these courses.

4. A PROPOSED SOLUTION: THE R PACKAGE AUTOSULT

The R package **AutoSULT** harnesses the computing and data management facilities of R and combines them with the typesetting prowess of RMarkdown and LaTeX, resulting in a tool that we believe will prove to be a robust and useful solution to the problems described above. We describe here various aspects of this package including the user interface and inputs and outputs, demonstrating the tool with a few examples.

The **AutoSULT** package is available for download (open-source and free to use) on GitHub and can be installed using the install_github() function[1], which requires the **devtools** package (Wickham et al., 2022).

devtools::install_github("ChrisGroendyke/AutoSULT", build_vignettes = TRUE)

Note that this package does require an installation of a LaTeX compiler. The package contains one introductory vignette, which walks the user through the use of the package.

4.1 *Using AutoSULT: Inputs and Outputs*

There are five main user-level functions in the package, corresponding to the five chapters in Dickson et al. (2019) that the problems are associated with. These functions are named ChXQA(), where X is 3, 4, 5, 8, or 10. See Tables 5 and 6 in Section 4.3 for a listing of problem types for each chapter. Each problem consists of the calculation of one specific probability or EPV, using entries from the corresponding table. These functions all have identical parameters and default parameter values, with the exception of the default names of the problem and solution files:

ChXQA(probspertype = 5, probspec = NULL, randomize = FALSE, probsummary = FALSE, qfile = "qfilechX", afile = "afilechX", keeptex = FALSE)

Each of these functions produces two PDFs, one containing the problems generated, and one containing the solutions to the problems. The functions don't return any values, per se, and are solely used for the side effect of generating these two PDFs.

The types of problems to be generated can be specified in two different ways. The first

[1] Per convention, R code, variables, and commands are given in `typewriter` font.

Journal of Actuarial Science
ISSN: 2958-0595 (Online)
Vol. 2, Issue No. 2, pp 1 - 20, 2024 www.carijournals.org

method is by using the scalar argument probspertype. This will simply generate a fixed, specified number of problems for each of the problem types in the chapter. The default value of this parameter is 5. Thus, for example, to generate 2 problems (and associated solutions) for each of the 16 types of Chapter 4 problems, we could use:

Ch4QA(probspertype = 2)

The other method of specifying the problem types is the vector argument probspec, which enables more fine tuning by specifying how many problems of each type are to be generated. For example, to generate 15 Chapter 3 problems total, consisting of 1 problem of type 1, 4 problems of type 2, 2 problems of type 3, 3 problems of type 5, and 5 problems of type 6, we could use:

Ch3QA(probspec = c(1, 4, 2, 0, 3, 5))

It is also possible to generate fixed number of problems, randomly chosen from the various problem types. For example, to generate 10 Chapter 5 problems chosen randomly from the 13 problem types from that chapter, we could use:

Ch5QA(probspec = tabulate(sample(1:13, 10, TRUE), nbins = 13))

Several other parameters to these functions are available to help customize the resulting problem and solution files; see Table 4.

Table 4: Additional parameters for user-level AutoSULT functions.

Paramete	Description
randomize	Boolean parameter controlling whether the problems will be randomized, or appear in problem type order.
probsummary	Boolean parameter controlling whether a summary of the problem types generated will be placed at the end of the question file.
qfile	Name (and path) to write the question PDF file.
afile	Name (and path) to write the solution PDF file.
keeptex	Boolean parameter controlling whether the .tex files compiled to produce the PDFs are kept or deleted after the PDF is rendered.

When the ChXQA() function is called, after generating the question parameters and problem types, the package builds two .Rmd files. When these .Rmd files are rendered, the

Journal of Actuarial Science
ISSN: 2958-0595 (Online)
Vol. 2, Issue No. 2, pp 1 - 20, 2024 www.carijournals.org

user has the option of retaining the intermediate .tex files, allowing them to later modify and recompile these files if desired.

4.2 *Examples*

We use three examples to demonstrate the usage of the **AutoSULT** package and to illustrate some of its key features. The first example is a small problem set from Chapter 5, consisting of four annuity EPV calculations and using the individual life SULT table:

```
set.seed(1)

myprobs <- c(0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0)

Ch5QA(probspec = myprobs)
```

Excerpts of the resulting output files are given in Figures 5 and 6:

Chapter 5 SULT Problems

Notes

- $i = 5\%$
- UDD refers to the Uniform Distribution of Deaths fractional age assumption
- $W2$ refers to the 2-term Woolhouse formula

Problems

1. $\ddot{a}_{35:\overline{12}|}$
2. $\ddot{a}^{(2)}_{42}$ under $W2$
3. $\ddot{a}^{(2)}_{54:\overline{37}|}$ under UDD
4. ${}_{2|}\ddot{a}_{74}$

Figure 5: Excerpt of Chapter 5 question output file.

The ages and (if applicable) product term lengths, payment frequencies, fractional age assumptions, and deferral periods are randomly generated so that a large number of distinct problems can be created quickly.

Consider the solution to problem 3 in Figure 6, a standard term life annuity EPV calculation. The SULT table contains whole life annuity EPVs, which are used to derive the required term annuity EPV. Further, the EPV requested is for that of an annuity making semi-annual payments; here the UDD fractional age assumption is applied to each of the whole life annuities. As there are many steps at which a student might make a conceptual or

Journal of Actuarial Science
ISSN: 2958-0595 (Online)
Vol. 2, Issue No. 2, pp 1 - 20, 2024 www.carijournals.org

calculational error, it is helpful to see each step in the solution carried out explicitly.

Solutions to Chapter 5 SULT Problems

1. $\ddot{a}_{\overline{35:\overline{12}|}} = \ddot{a}_{\overline{12}|} + {}_{12}E_{35} \cdot \ddot{a}_{47} = \frac{1-0.55683}{0.04762} + 0.55302 \cdot 17.5189 = \boxed{18.99467}$, where
   $${}_{12}E_{35} = {}_{12}p_{35} \cdot v^{12} = \frac{\ell_{47}}{\ell_{35}} \cdot v^{12} = \frac{98874.5}{99556.7} \cdot \frac{1}{(1.05)^{12}} = (0.99315)(0.55684) = 0.55302$$

2. $\ddot{a}^{(2)}_{42} \overset{W2}{=} \ddot{a}_{42} - \frac{2-1}{2 \cdot 2} = 18.2176 - 0.25 = \boxed{17.9676}$

3. $\ddot{a}^{(2)}_{\overline{54:\overline{37}|}} = \ddot{a}^{(2)}_{54} - {}_{37}E_{54} \cdot \ddot{a}^{(2)}_{91} \overset{UDD}{=} 16.01387 - 0.06311 \cdot 4.63026 = \boxed{15.72165}$, where
   $$\ddot{a}^{(2)}_{54} \overset{UDD}{=} \alpha(2) \cdot \ddot{a}_{54} - \beta(2) = 1.00015 \cdot 16.2676 - 0.25617 = 16.01387, \text{ and}$$
   $$\ddot{a}^{(2)}_{91} \overset{UDD}{=} \alpha(2) \cdot \ddot{a}_{91} - \beta(2) = 1.00015 \cdot 4.8857 - 0.25617 = 4.63026, \text{ and}$$
   $${}_{37}E_{54} = {}_{37}p_{54} \cdot v^{37} = \frac{\ell_{91}}{\ell_{54}} \cdot v^{37} = \frac{37618.6}{98022.4} \cdot \frac{1}{(1.05)^{37}} = (0.38378)(0.16444) = 0.06311$$

4. ${}_{2|}\ddot{a}_{74} = {}_{2}E_{74} \cdot \ddot{a}_{76} = 0.87567 \cdot 6.1308 = \boxed{8.72815}$, where
   $${}_{2}E_{74} = {}_{2}p_{74} \cdot v^{2} = \frac{\ell_{76}}{\ell_{74}} \cdot v^{2} = \frac{83632.9}{86627.6} \cdot \frac{1}{(1.05)^{2}} = (0.96543)(0.90703) = 0.87567$$

Figure 6: Excerpt of Chapter 5 solution output file.

The second example contains a single problem of each type from Chapter 8 and uses the SSD table. The following commands produce output which is excerpted in Figures 7 and 8:

```
set.seed(1)
Ch8QA(probspertype = 1, probsummary = TRUE)
```

Chapter 8 SSD Problems

Notes

- $i = 5\%$
- Problems refer to the Standard Sickness-Death Model

Problems

1. ${}_{20}p^{00}_{52}$
2. ${}_{20}p^{01}_{53}$
3. ${}_{20}p^{02}_{55}$
4. ${}_{20}p^{10}_{58}$
5. ${}_{20}p^{11}_{51}$

Figure 7: Excerpt of Chapter 8 question output file.

Note the solution for question 3, which is the probability of an individual, who is currently age 55 in State 0, being in State 2 in 20 years. Based on the information in the table, this

Journal of Actuarial Science
ISSN: 2958-0595 (Online)
Vol. 2, Issue No. 2, pp 1 - 20, 2024 www.carijournals.org

probability must be derived from the probabilities of being in each of the other two states in 20 years, which themselves must be calculated from multiple 10-year probabilities.

Solutions to Chapter 8 SSD Problems

1. ${}_{20}p_{52}^{00} = {}_{10}p_{52}^{00} \cdot {}_{10}p_{62}^{00} + {}_{10}p_{52}^{01} \cdot {}_{10}p_{62}^{10} = 0.80533 \cdot 0.51586 + 0.08298 \cdot 0.00786 = \boxed{0.41609}$

2. ${}_{20}p_{53}^{01} = {}_{10}p_{53}^{00} \cdot {}_{10}p_{63}^{01} + {}_{10}p_{53}^{01} \cdot {}_{10}p_{63}^{11} = 0.78577 \cdot 0.25397 + 0.09318 \cdot 0.65579 = \boxed{0.26067}$

3. ${}_{20}p_{55}^{02} = 1 - {}_{20}p_{55}^{00} - {}_{20}p_{55}^{01} = 1 - 0.28972 - 0.28812 = \boxed{0.42216}$, where
   ${}_{20}p_{55}^{00} = {}_{10}p_{55}^{00} \cdot {}_{10}p_{65}^{00} + {}_{10}p_{55}^{01} \cdot {}_{10}p_{65}^{10} = 0.74091 \cdot 0.39038 + 0.11682 \cdot 0.00415 = 0.28972$, and
   ${}_{20}p_{55}^{01} = {}_{10}p_{55}^{00} \cdot {}_{10}p_{65}^{01} + {}_{10}p_{55}^{01} \cdot {}_{10}p_{65}^{11} = 0.74091 \cdot 0.29288 + 0.11682 \cdot 0.60878 = 0.28812$

Figure 8: Excerpt of Chapter 8 solution output file.

The third and final example consists of only two problems from Chapter 10, using the joint life SULT table. For this example, the output file for the questions is not shown, while the output file for the solutions in given in Figure 9.

set.seed(1)

myprobs <- c(rep(0, times = 6), 1, 1, rep(0, times = 6))

Ch10QA(probspec = myprobs, randomize = TRUE)

Both of these problems involve term insurance EPVs, the former for a first-to-die insurance and the latter for a second-to-die insurance. The second problem in particular has several steps, the calculations for which are nested, making use of earlier problem types in this chapter.

Solutions to Chapter 10 SULT Problems

1. $A_{\overset{1}{\overline{40:50}}:\overline{9}|} = A_{40:50} - {}_{9}E_{40:50} \cdot A_{49:59} = 0.21163 - 0.62967 \cdot 0.3079 = \boxed{0.01775}$, where
   ${}_{9}E_{40:50} = {}_{9}p_{40:50} \cdot v^{9} = 0.97682 \cdot 0.64461 = 0.62967$, where
   ${}_{9}p_{40:50} = \frac{\ell_{49}}{\ell_{40}} \cdot \frac{\ell_{59}}{\ell_{50}} = \frac{98684.9}{99338.3} \cdot \frac{96929.6}{98576.4} = 0.97682$

2. $A_{\overset{1}{\overline{44:44}}:\overline{10}|} = A^{1}_{44:\overline{10}|} + A^{1}_{44:\overline{10}|} - A_{\overset{1}{\overline{44:44}}:\overline{10}|} = 0.00813 + 0.00813 - 0.01618 = \boxed{0.00008}$, where
   $A_{\overset{1}{\overline{44:44}}:\overline{10}|} = A_{44:44} - {}_{10}E_{44:44} \cdot A_{54:54} = 0.19101 - 0.60058 \cdot 0.29111 = 0.01618$, where
   ${}_{10}E_{44:44} = {}_{10}p_{44:44} \cdot v^{10} = 0.97828 \cdot 0.61391 = 0.60058$, where
   ${}_{10}p_{44:44} = \frac{\ell_{54}}{\ell_{44}} \cdot \frac{\ell_{54}}{\ell_{44}} = \frac{98022.4}{99104.3} \cdot \frac{98022.4}{99104.3} = 0.97828$

Figure 9: Excerpt of Chapter 10 solution output file.

4.3 *Further Details of AutoSULT*

The typesetting functionality embedded in these functions uses the **actuarialangle** (Goulet, 2017), **actuarialsymbol** (Beauchemin & Goulet, 2019), and **caption** (Sommerfeldt, 2023) packages in LaTeX. On the R side, the **kableextra** (Zhu, 2021), **magrittr** (Bache & Wickham, 2022), **rmarkdown** (Allaire et al., 2023), and **knitR** (Xie, 2023) packages are used.

Note that the necessary tables are stored as separate R data objects in the package. Thus, these tables are available for use independently of the functions in the package, allowing for even greater flexibility and functionality by, say, applying mortality improvements to the tables, or adjusting them to account for extra risks. As many operations in R can be vectorized, this also provides a means for very efficient calculations involving these tables. Also of potential importance to instructors is the set.seed() function in R, which allows for problem sets which are both personalized and reproducible by, for example, having each student set their individual seed value to their own particular ID number. Finally, we note that both the calculation and typesetting functions in this package were designed to be modular, taking computational advantage of the recursive nature of many of these types of problems. Tables 5 and 6 give a listing of problem types for each chapter.

Journal of Actuarial Science
ISSN: 2958-0595 (Online)
Vol. 2, Issue No. 2, pp 1 - 20, 2024 www.carijournals.org

Table 5: AutoSULT problem types for Chapters 3, 4, and 5. These chapters make use of the individual life SULT table.

Chapter	Problem Type	Problem Description
3	1	${}_tp_x$
3	2	${}_tq_x$
3	3	${}_{k\|}{}_tq_x$
3	4	${}_{t+s}p_{x+r}$ under UDD or CF
3	5	${}_{t+s}q_{x+r}$ under UDD or CF
3	6	${}_{k+s\|}{}_{t+w}q_{x+r}$ under UDD or CF
4	1	${}_nE_x$
4	2	$A^{1}_{x:\overline{n}\|}$
4	3	$\bar{A}^{1}_{x:\overline{n}\|}$
4	4	$A^{1\,(m)}_{x:\overline{n}\|}$
4	5	$A^{1}_{x:\overline{2}\|}$ or $A^{1}_{x:\overline{3}\|}$ for $i \neq 5\%$
4	6	$\bar{A}^{1}_{x:\overline{2}\|}$ or $\bar{A}^{1}_{x:\overline{3}\|}$ for $i \neq 5\%$
4	7	$A^{1\,(m)}_{x:\overline{2}\|}$ or $A^{1\,(m)}_{x:\overline{3}\|}$ for $i \neq 5\%$
4	8	$A_{x:\overline{n}\|}$
4	9	$\bar{A}_{x:\overline{n}\|}$
4	10	$A^{(m)}_{x:\overline{n}\|}$
4	11	${}_{u\|}A_x$
4	12	${}_{u\|}\bar{A}_x$
4	13	${}_{u\|}A^{(m)}_x$
4	14	${}_{u\|}A^{1}_{x:\overline{n}\|}$
4	15	${}_{u\|}\bar{A}^{1}_{x:\overline{n}\|}$
4	16	${}_{u\|}A^{1\,(m)}_{x:\overline{n}\|}$
5	1	$\ddot{a}_{x:\overline{n}\|}$
5	2	$\ddot{a}_{\overline{x:\overline{n}\|}}$
5	3	$\bar{a}_x$ under UDD or $W2$
5	4	$\ddot{a}^{(m)}_x$ under UDD or $W2$
5	5	$\bar{a}_{x:\overline{n}\|}$ under UDD or $W2$
5	6	$\ddot{a}^{(m)}_{x:\overline{n}\|}$ under UDD or $W2$
5	7	${}_{u\|}\ddot{a}_x$
5	8	${}_{u\|}\ddot{a}_{x:\overline{n}\|}$
5	9	${}_{u\|}\ddot{a}_{\overline{x:\overline{n}\|}}$
5	10	${}_{u\|}\bar{a}_x$ under UDD or $W2$
5	11	${}_{u\|}\ddot{a}^{(m)}_x$ under UDD or $W2$
5	12	${}_{u\|}\bar{a}_{x:\overline{n}\|}$ under UDD or $W2$
5	13	${}_{u\|}\ddot{a}^{(m)}_{x:\overline{n}\|}$ under UDD or $W2$

Journal of Actuarial Science
ISSN: 2958-0595 (Online)
Vol. 2, Issue No. 2, pp 1 - 20, 2024 www.carijournals.org

Table 6. AutoSULT problem types for Chapters 8 and 10. These chapters make use of the SSD and joint life SULT table, respectively.

Chapter	Problem Type	Problem Description
8	1	${}_{20}p_x^{00}$
8	2	${}_{20}p_x^{01}$
8	3	${}_{20}p_x^{02}$
8	4	${}_{20}p_x^{10}$
8	5	${}_{20}p_x^{11}$
8	6	${}_{20}p_x^{12}$
8	7	$\bar{a}^{00}_{x:\overline{10}\rvert}$
8	8	$\bar{a}^{01}_{x:\overline{10}\rvert}$
8	9	$\bar{a}^{10}_{x:\overline{10}\rvert}$
8	10	$\bar{a}^{11}_{x:\overline{10}\rvert}$
8	11	$\bar{a}^{00}_{x:\overline{20}\rvert}$
8	12	$\bar{a}^{01}_{x:\overline{20}\rvert}$
8	13	$\bar{a}^{10}_{x:\overline{20}\rvert}$
8	14	$\bar{a}^{11}_{x:\overline{20}\rvert}$
8	15	$\bar{A}^{02}_{x:\overline{10}\rvert}$
8	16	$\bar{A}^{12}_{x:\overline{10}\rvert}$
8	17	$\bar{A}^{02}_{x:\overline{20}\rvert}$
8	18	$\bar{A}^{12}_{x:\overline{20}\rvert}$
10	1	${}_np_{xy}$
10	2	${}_nq_{xy}$
10	3	${}_np_{\overline{xy}}$
10	4	${}_nq_{\overline{xy}}$
10	5	${}_nE_{xy}$
10	6	${}_nE_{\overline{xy}}$
10	7	$A^{\;1}_{xy:\overline{n}\rvert}$
10	8	$A^{\;1}_{\overline{xy}:\overline{n}\rvert}$
10	9	$A_{\overline{xy}}$
10	10	$\ddot{a}_{x:y:\overline{20}\rvert}$
10	11	$\ddot{a}_{x\vert y}$
10	12	$\ddot{a}_{\overline{xy}}$
10	13	$\ddot{a}_{\overline{x:y}:\overline{10}\rvert}$
10	14	$\ddot{a}_{\overline{x:y}:\overline{20}\rvert}$

5. CONCLUSION AND RECOMMENDATIONS

5.1 *Conclusion*

In this paper, we have described an issue common to introductory life contingencies courses that has frustrated many instructors: the need to produce a high volume of specific types of problems containing actuarial symbols that are notoriously difficult to typseset, especially in an efficient manner. To address this issue, we have introduced an open-source software tool called **AutoSULT** that automatically generates and typesets an arbitrary number of life

Journal of Actuarial Science
ISSN: 2958-0595 (Online)
Vol. 2, Issue No. 2, pp 1 - 20, 2024 www.carijournals.org

contingencies problems (and step-by-step solutions) of types specified by the user. These problems represent some of the most common calculations done in first and second semester life contingencies courses and utilize standard tables. The interface is intended to be flexible and easy to use, resulting in significant time savings for the end-user. This tool could be used by a life contingencies instructor for a variety of purposes, including homework, quizzes, or recitation sessions, with the possibility of generating personalized problem sets for each individual. It could also be used by students to generate specific types of problems on topics they need more practice with.

5.2 *Recommendations*

There are various recommendations that we may consider going forward, both in terms of continuing to automate the process of creating these materials as well as reducing the need for them. In the former category, these might include an increased use of automation and advances in software (preferably open source). The **AutoSULT** tool is flexible and adaptable, and will continue to be maintained and expanded going forward; thus, it can help in this regard. For the latter category, as actuarial science continues to progress and develop, it may be possible to move away from some of the actuarial notation that necessitates complex typesetting. Dickson et al. (2019) has already taken some steps in this direction, and it is anticipated that others will continue this trend moving forward.

References

Allaire, J., Xie, Y., Dervieux, C., McPherson, J., Luraschi, J., Ushey, K., Atkins, A., Wickham, H., Cheng, J., Chang, W., & Iannone, R. (2023). rmarkdown: dynamic documents for R. *R Package Version 2.22,* https://github.com/rstudio/rmarkdown

Bache, S., & Wickham, H. (2022). magrittr: a forward-pipe operator for R. *R Package Version 2.0.3.* https://CRAN.R-project.org/package=magrittr

Beauchemin, D., & Goulet, V. (2019). Actuarial symbols of life contingencies and financial mathematics. http://ctan.org/pkg/actuarialsymbol

Bowers, N. L., Gerber, H. U., Hickman, J. C., Jones, D. A., & Nesbitt, C. J. (1997). *Actuarial Mathematics*, (2nd ed.). Society of Actuaries.

Camilli, S. J., Duncan, I., & London, R. L. (2014). *Models for Quantifying Risk*, (6th ed). ACTEX Publishing.

Castellares, F., Patricio, S., & Lemonte, A. (2022). On the Gompertz-Makeham law: a useful mortality model to deal with human mortality. *Brazilian Journal of Probability and Statistics.* 36. 613-639. http://doi.org/10.1214/22-BJPS545

Cocovi-Solberg, D.J., & Miró, M. (2015). CocoSoft: educational software for automation in the analytical chemistry laboratory. *Analytical and Bioanalytical Chemistry.* 407, 6227–6233. https://doi.org/10.1007/s00216-015-8834-8

Combéfis S. (2022). Automated Code Assessment for Education: Review, Classification and Perspectives on Techniques and Tools. *Software*. 1(1):3-30. https://doi.org/10.3390/software1010002

Dickson, D. C., Hardy, M. R., & Waters, H. R. (2019). *Actuarial Mathematics for Life Contingent risks.* (3rd ed.). Cambridge University Press. https://doi.org/10.1017/9781108784184

Goulet, V. (2017). Actuarial angle symbol for life contingencies and financial mathematics. https://www.ctan.org/pkg/actuarialangle

Graunt, J. (1662). Natural and political observations made upon the bills of mortality.

Halley, E. (1693). An estimate of the degrees of the mortality of mankind, drawn from curious tables of the births and funerals at the city of Breslaw; with an attempt to ascertain the price of annuities upon lives. *Philosophical Transactions*.

Jones, D. (1843). *On the Value of Annuities and Reversionary Payments: with Numerous Tables (Vol. 2).* Baldwin and Cradock.

King, G. (1889). *Institute of Actuaries' Text-book–Part II (with List of Errata).* Journal of the Institute of Actuaries, 28(2), 160-166.

Kurdi, G., Leo, J., Parsia, B., Sattler, U., & Al-Emari, S. (2020). A systematic review of automatic question generation for educational purposes. *International Journal of Artificial Intelligence in Education*, *30*, 121-204.

Layton, C., & Layton, E. (1899). Discussion and resolutions on a universal actuarial notation. *Transactions of the Second International Actuarial Congress.*

Louca, A. (2020). A proposal for universal actuarial notation [working paper]. *Casualty Actuarial Society.* https://www.casact.org/sites/default/files/2021-02/working-paper-louca-2020-07.pdf

Makeham, W. (1860). On the law of mortality and the construction of annuity tables. *Journal of the Institute of Actuaries*, 8(6), 301-310.

R Core Team (2023). R: A language and environment for statistical computing. *R Foundation for Statistical Computing*, Vienna, Austria. https://www.R-project.org.

Society of Actuaries (2024, June 26). Exams and Requirements. https://www.soa.org/education/exam-req/default/

Sommerfeldt, A. (2023). Customizing captions of floating environments. https://ctan.org/pkg/caption

Wickham, H., Hester, J., Chang, W., & Bryan, J. (2022). devtools: tools to make developing

R packages easier. *R Package Version 2.4.5*, https://CRAN.R-project.org/package=devtools

Woolhouse, W. (1869). On an improved theory of annuities and assurances. *Journal of the Institute of Actuaries and Assurance Magazine*. 15(2), 95-125

Xie, Y. (2023). knitr: A general-purpose package for dynamic report generation in R. *R Package Version 1.43.* https://yihui.org/knitr/

Zhu, H. (2021). kableExtra: construct complex table with 'kable' and pipe syntax. *R Package Version 1.3.4.* https://CRAN.R-project.org/package=kableExtra