USING SPREADSHEET TO SIMULATE BASIC STOCHASTIC MODELS

Khosrow Moshirvaziri, Department of Information Systems, California State University, Long Beach, Long Beach, CA 90840, 562-985-7965, Khosrow.Moshirvaziri@csulb.edu

Mahyar A. Amouzegar, Department of Economics and Finance, University of New Orleans, New Orleans, LA 70148, 504-280-6595, mahyar@uno.edu

ABSTRACT

This paper presents a basic approach for illustrating complex probability models using spreadsheets as a simulation tool. We will illustrate this approach by using three popular probability examples. Our targeted audience is practitioners in business professions and those with less rigorous training in computational mathematics. Modeling and Simulation are well known as efficient and effective alternatives to analytical methods in solving many businesses, engineering, and scientific problems. These models are easily understood and can be tweaked for further exploration, and more importantly, they are risk-free, and cost-effective in terms of experimentations of "What-If" analysis. The capability of finding reasonable estimates via simulation, where no analytical solution is known for the problem, may shed light on our understanding and insights about the problem and eventually lead us to find an analytical solution.

Keywords: *Modeling, Simulation, Non-probability methods, Spreadsheet Modeling.*

INTRODUCTION

The recent efforts to prepare the workforce for today's digital economy have spurred an increased interest in developing and introducing *Analytics* into business schools' curriculums. This promulgation has pointed to the import of retooling teaching methods to enhance the learning process. However, there must be a careful balance between the simplicity of these tools and their functional sophistication. This is particularly salient for users with a less rigorous background in mathematical training and underlying statistical modeling. In this paper, we concentrate on using spreadsheets (e.g., MS-Excel) since they are an essential business tool in finance and accounting, and they are readily available on most systems used by our target audience; students and those who use spreadsheets as part of their routine activities as business professionals.

We will use classic examples in popular probability games (models), familiar to average practitioners, to show how one computes the probability of certain events and outcomes without using sophisticated statistical theory. Furthermore, we will illustrate how to use spreadsheets' capability and functionality to generate, organize, and manipulate virtual data to arrive at reasonable estimates of those particular outcomes of interest. For some of the examples we discuss, we have the luxury of verifying the accuracy of the simulated answers with the true answers of probabilistic outcomes obtained according to the rigor of mathematical derivations and closed-form solutions via statistical theory. This will demonstrate how one must structure the underlying model, and define input parameters, variables, and algorithmic layout steps in the random generation of artificial data as inputs to those models, to mimic realistic situations. Before going into the discussion of our examples, we need to briefly describe the model elements and terminologies used in our discussion.

Model - Generally, models are abstractions, computer-based representations of real systems under study. By design, simulated results will be generated through a model via a spreadsheets environment, and thus we need to have a meaningful representation of variables prior to random generation of the data required by our models. This initial step is crucial. One may need to change initial input values for increased accuracy and better results from these models.

Variables – These are memory locations, essentially a placeholder for any mathematical object, designed to hold a numerical representation of random quantities defined on a connected (continuous) or disconnected (discrete) set of values. Variables (along with any other pertinent assumptions the modeler has imposed on the model) must clearly be defined and quantified.

BIRTHDAY PARADOX

This example is discussed in more detail in earlier articles [see Moshirvaziri, Amouzegar, and Rezayat, 2017, Gleich, 2010, and Sun, 2011]; thus, we briefly introduce it again for clarity. This example deals with calculating the probability of finding at least two (or any number) people with the same birthday (matching only the months and day of the year but may be on different years) among a sample of size n people. The problem is a variation of a much more important class of problems known as "Birthday Attack" with applications in information security.

Generating a random sample - Let x denote a randomly selected individual's birthday. We assume birthdays occur uniformly throughout the year from January 1 to December 31. This is a reasonable assumption (See Figure 1) based on decades of data from the Social Security Administration (FiveThirtyEight.com). Furthermore, we assume there are 365 days in a given year. Thus, variable x picks on values from 1 to 365 at random. Given this representation, we can simply generate a random sample of any size using embedded functions on a spreadsheet. For example, an Excel function "RANDBETWEEN (1,365)" will generate an instance of the random variable x in a cell of our spreadsheet. Note that the value of x, in turn, may be converted to the exact month and day of that month by a reverse operation, if desired. This is intuitively true because any data point uniquely represents only one day of the year.

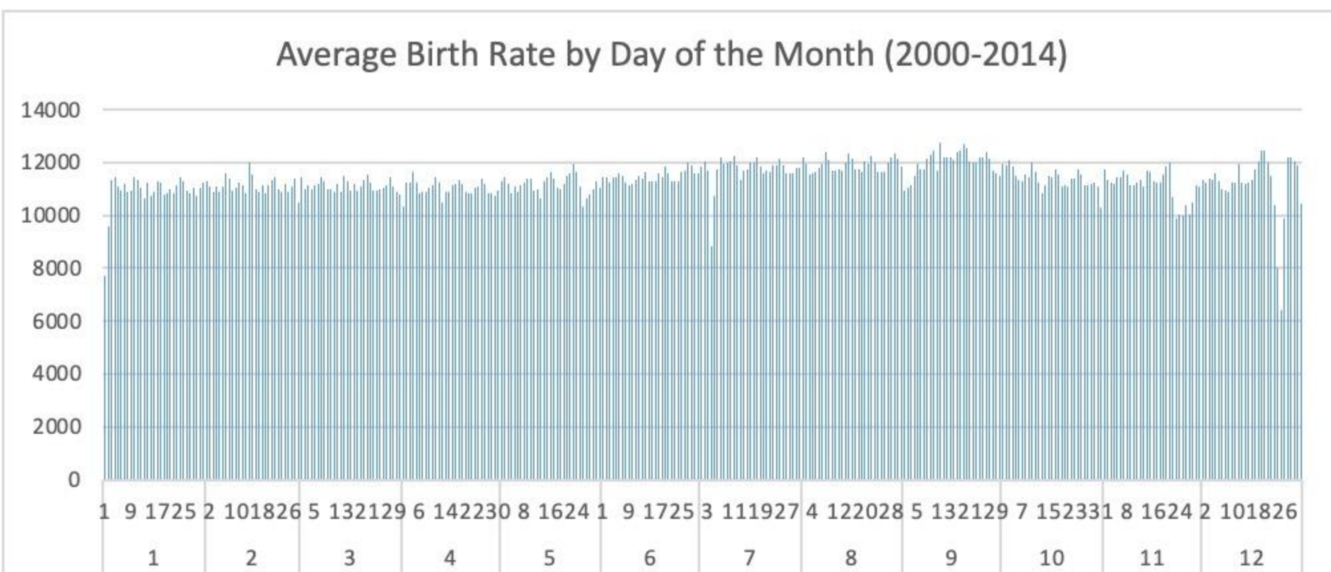


Figure 1: Average Birthrates by day of the month

If two instances of variable x are assigned equal values, this implies the detection of a match, which is also referred to as a *collision* in the terminology of the *Birthday Attack* problem. Therefore, a simple detection method can be used to find two generated values with the same magnitude (e.g., “Frequency” function, such as {=FREQUENCY (D2:D94, D2:D94)} applied to a vector of size n of generated birthdays, we can find unique frequencies of the vector’s elements). Then other embedded Excel functions such as “IF” and “COUNTIF” s (e.g., s =IF (COUNTIF (G2:G94,">=3") > 0, 1, 0) can help in determining the “Collision” or “Matches” points. (i.e., precisely one, at least one, exactly two, at least two, exactly three, at least three, and so on). We then find the mean of the number of “true” outputs from the last function output, representing the simulated probability of the case under study. For this example, we generated 100 samples of various sizes of birthdays and found results that were within an acceptable threshold from the closed-form solutions, where available. For example, for the case of at least two birthday matches in the sample size of 30, we found that in 70 of the samples (out of 100), we had at least one match.[1]

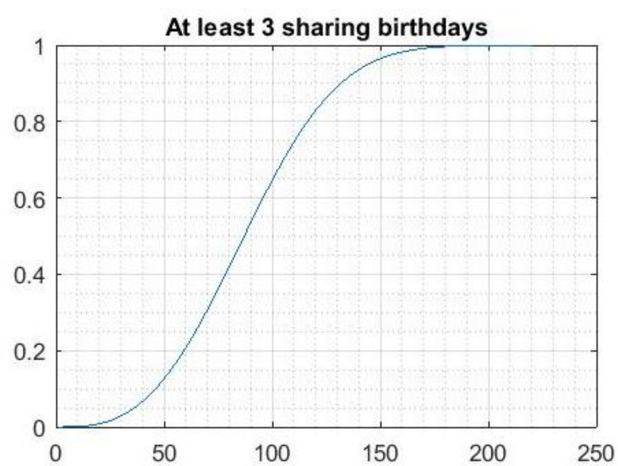
Hence, the P(at least one match) = 0.70000, closely matching the result obtained via analytical solution [Moshirvaziri, Amouzegar, 2017].

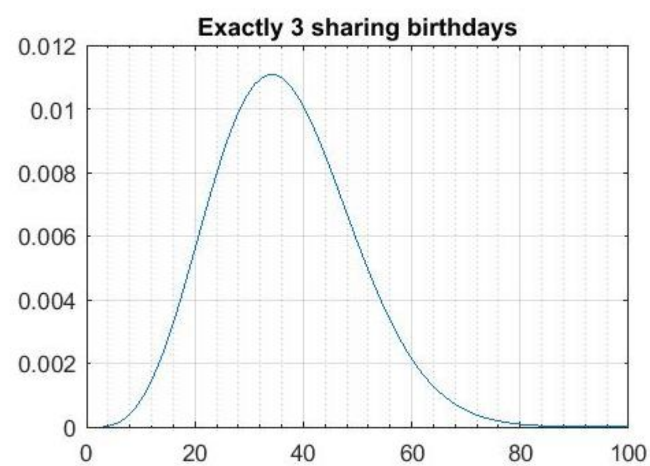
$$P(X_2 \geq 1) = 1 - F(X_2) = 1 - p(n; H) = 1 - \frac{P_n^H}{H^n}$$

For H = 365 and $n = 30$,

Where, as before, n is the sample size, X_2 denotes the collision of two birthdays, H is the number of periods (for example, the number of days in a year) considered, and P_n^H is permutation function. The result therefore is $P(X_2 \geq 1) = 1 - 0.293683757 = 0.706316243$

If we let H = 365 and $n = 93$, $P(X_3 \geq 1) = 0.5430$ for at least one trio collision, as shown in the graph below. In the calculation via spreadsheet, we obtained 0.5400, which is still within an acceptable range of theoretical results.





[1] The generated datasets are available upon request and will be provided during the conference presentation

The complete discussions of density functions, as well as distribution functions for a variety of different birthday collision scenarios, are presented in the authors' prior publication listed.

PROBABILITY GAMES

Preliminary model setup - below, we lay out a data structure for the virtual generation of a random drawing of cards from a deck of 52 playing cards. Let $x_i, for\ i = 0, 1, 2, \cdots, 51,$ denote a uniformly generated random number in the set of nonnegative integers $\{0, 1, 2, \cdots, 51\}$. We can implement this using the Excel function $x_i = Randbetween$ (0,51).

The Suit designation - The suit designation of the card is found by the integer quotient of the division of $x_i/13$. For example, suppose $x_i = 48$. Then, INT (48/13), which is 3 designates the suit to be "Spade". Generally, if the quotient of the integer division is 0, or 1, or 2, or 3, we assign the suite of Club, Diamond, Heart, and Spade, respectively.

The rank designation - The modulus or remainder of the corresponding decimal division of $x_i/13$. is 48 – 3*13 = 9 designates the card's rank, also referred to as value, label, or name. Hence, the randomly generated card in this example has a rank of 9 and a suit of Spade. Generally, if the modulus or remainder of the decimal division (rank) is any member of the set: $\{0, 1, 2, \cdots, 12\}$, we assign the label of Ace, King, 2, 3, …, 10, Jack, and Queen, respectively. This is illustrated in the data structure shown in the table below.

Cards labels →		**Ace**	**King**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Jack**	**Queen**
Card Rank →		0	1	2	3	4	5	6	7	8	9	10	11	12
Club	0	0	1	2	3	4	5	6	7	8	9	10	11	12
Diamond	1	13	14	15	16	17	18	19	20	21	22	23	24	25
Heart	2	26	27	28	29	30	31	32	33	34	35	36	37	38
Spade	3	39	40	41	42	43	44	45	46	47	48	49	50	51
Suits	Card Suit													

With this model set up, we can simulate a random hand of Poker or Blackjack and identify the hand's name according to the rules of these popular games. For example, in a poker game, by creating a bin of size 10 as illustrated below, we can keep track of the frequencies of each bin value, which corresponds to the occurrence of a particular hand of Poker.

This data structure also tends to be very useful for deriving analytical formulas for each possible hand of the card game. For example, in the next two sections we will illustrate how to derive those solutions using the ranks (the columns of the matrix above) and the suits (the corresponding rows). Moreover, the derivations are not unique. For example, we will illustrate the derivation of the density function for the game of 5-card Poker in one way. However, we can derive the same solution using a different selection scheme from the structured data above.

A Bin construction for Spreadsheet simulation – We have built the following Bin of size ten in order to collect data on the frequency of random hands of the simulated game. At the end of the simulation, each Bin cell will have the total frequency of the designated hand, which is used in the calculation of the simulated probabilities.

Bin Value	Hand Type
0	Reject
1	Straight flush
2	Four of a kind
3	Full house
4	Flush
5	Straight
6	Three of a kind
7	Two Pairs
8	One Pair
9	Nothing

Bin zero - Note that the Bin value of zero (0) will be necessary since the rules may be violated during the random generation of 5 cards to simulate a hand of Poker. For example, two cards of the same rank and suit may be generated (such as two Aces of clubs) or five cards of the same suit and rank may be generated, which is in violation of the game rules. The counts of those "illegal" hands would be enumerated at a Bin value of zero. Bin zero frequency will help us to estimate the simulated probability of those outcomes more accurately. This is true regardless of the platform on which the simulation is performed, on a spreadsheet or via commercially available simulation tools.

Bin 9 – This cell accounts for all hands that are not accounted for in a lower number of cells. According to the rules of the game, if the highest ranked card in a nothing hand, is higher in rank than that of another nothing hand's highest card, then it beats that hand. For example, on one Nothing hand, there may be Ace high, and on the other King high. The first hand beats the second.

Let us now look into the theoretical calculations of possible outcomes of this game and find the empirical probability density function for the game of Poker. Finally, we use these results to validate how good are the simulated results through the spreadsheet approach versus those of theoretical (closed form solutions).

The density function - Clearly, we can calculate the frequency of any particular outcome out of the total of $C_5^{52} = 2{,}598{,}960$ hands of a 5-card poker game that may be dealt from a deck of 52 cards. The data structure given earlier greatly facilitates this process—the density function for various hands is given in the table below. For the continuity of our discussion, we also include the derivation of various hands of this game in the next section.

Hand	Frequency	Probability
Royal flush	4	0.00000154
Straight flush	36	0.00001385
Four of a kind	624	0.00024010
Full house	3744	0.00144058
Flush	5108	0.00196540
Straight	10200	0.00392465
Three of a kind	54912	0.02112845
Two Pairs	123552	0.04753902
One Pair	1098240	0.42256903
Nothing	1302540	0.50117739
	2598960	1.00000000

Frequency of various hands in Poker –

In this section, we only derive the number of hands of a particular outcome for the game. Let F denote the frequency of the hand. In order to find the probability of each hand, we can divide this frequency by the total number of possible hands, C_5^{52} or 2,598,960. In addition, we order the equation numbers to be consistent with those of the Bin numbers for an easy cross reference.

Royal Flush hand - There are four Royal Flushes of the highest rank, one in each suit designation.

$$F(Royal\ Flush) = C_1^4 = 4$$

Straight Flush hand - There are ten choices for the minimum rank of the straight and four designations for the suit, deducted by four since four of which are Royal Flushes.

$$F(Straight\ Flush) = C_1^{10} C_1^4 - 4 \quad = 36 \tag{1}$$

Four of a Kind hand – This implies one rank designation for the cards of the same suit and one rank designation for the single card, which may be of any of the four suits.

$$F(four\ of\ a\ kind) = \ C_1^{13}\, C_4^4\ \ C_1^{12}\, C_1^4 \quad = 624 \tag{2}$$

Full House Hand – Similar to the strategy implemented for the previous hand, we have one rank designation and a choice of three from the four suits for the first part and one rank designation (different from the first designation) with a choice of two suits from the possible four suits. Hence, we derive,

$$F(full\ House) = \ C_1^{13}\, C_3^4\ \ C_1^{12}\, C_2^4\ = 3{,}744 \tag{3}$$

Flush Hand - One designation of the four suits with a choice of five from the 13 ranks, deducted by the frequency of the Royal and Straight flushes.

$$F(Flush) = C_1^4\, C_5^{13} - 40\ \ =\ \ 5{,}108 \tag{4}$$

Ordinary Straight hand - The frequency of straight is deducted by the frequencies of straight flushes and Royal flushes to find the frequency of ordinary straights

$$F(Straight) = \ C_1^{10}\, (C_1^4)^5 - \ C_1^4 - C_1^{10} C_1^4 + 4 \quad = 10{,}200 \tag{5}$$

Three of a kind hand – We have one rank designation with the choice of three suits and two more rank designations (different from the first and each other) each with a choice of one suit from the four possible suits, hence the derivation below.

$$F(Three\ of\ a\ kind) = \ C_1^{13}\, C_3^4\ \ C_2^{12}\, (C_1^4\,)^2 \quad = 54{,}912 \tag{6}$$

Two Pair hand – We have two rank designations with the choice of two suits for each, and another rank designation with the choice of one from the four suits to derive,

$$F(Two\ pair) = \ C_2^{13}\, (C_2^4)^2\, C_1^{11}\, C_1^4 \quad = \ 123{,}552 \tag{7}$$

One Pair hand – Similar to the above, we have one rank designation with choice of two suits and three more rank designations, each with the choice of one suit only from the four possible suits; hence we have,

$$F(One\ Pair) = \ C_1^{13}\, C_2^4\ \ C_3^{12}\, (C_1^4\,)^3 \quad = \ \ 1{,}098{,}240 \tag{8}$$

Nothing hand - This implies five rank designations from the deck (five different cards) reduced by ten possible straight hands. Then for each rank, we have the choice of one from four suits, deducted by four possible "flash" hands (five cards of the same suit). Hence, the following derivation.

$$F(Nothing) = (C_5^{13} - 10)(\,(C_1^4)^5 - 4) = \;\; 1{,}302{,}540 \tag{9}$$

Simulated Probability density function (pdf) for the game - Using the functionality of the spreadsheet as per our model description, we generate a great number of simulated hands of the game of 5-card poker. The counter for each Bin will keep track of different possible outcomes. Below, is a table of simulated outcomes for a given run.

Bin Value	Frequency	Simulated Probability
0	17889	
1	3	0.00003654
2	17	0.00020704
3	116	0.00141272
4	155	0.00188769
5	286	0.00348309
6	1777	0.02164144
7	3882	0.04727747
8	34692	0.42250125
9	41183	0.50155278
	100000	1.00000000

Finally, the table below illustrates how good the simulated results are compared to the true answers found by the density function.

Bin Value	Frequency	Simulated Probability	True Probability	Difference
0	17889		0.00000154[2]	
1	3	0.00003654	0.00001385	0.00002268
2	17	0.00020704	0.00024010	0.00003306
3	116	0.00141272	0.00144058	0.00002785
4	155	0.00188769	0.00196540	0.00007771
5	286	0.00348309	0.00392465	0.00044156
6	1777	0.02164144	0.02112845	0.00051299
7	3882	0.04727747	0.04753902	0.00026155
8	34692	0.42250125	0.42256903	0.00006778
9	41183	0.50155278	0.50117739	0.00037538
	100000	1.00000000	1.00000000	

[2] Probability of Royal Flush

The difference column shows numbers at the fourth and fifth decimal places. The Chi-square, *goodness-of-fit* test produces a large *p-value,* which confirms a perfect fit. Again, this illustrates how through the knowledge and skills of modeling and simulation, one can obtain reliable simulated results, thereby compensating for the lack of an analytical approach to a random phenomenon under study. Note that if we generate more random hands, say by one or two orders of magnitude, the accuracy of results will increase and thus, the difference column will further decrease. However, the only drawback is that the volume of generated data will be too large to handle due to the limitations of the spreadsheet.

FRAUD DETECTION

It may be counterintuitive, but in many naturally occurring or human generated data sets, the first digits of numbers often follow a distribution similar to the table below. That is, the number 1 appears as the significant leading digit[3] about 30% of the time, while 9 appears as the most significant digit less than 5% of the time.

First digit	1	2	3	4	5	6	7	8	9
Probability	0.3010	0.1761	0.1249	0.0969	0.0792	0.0669	0.0580	0.0512	0.0458

This distribution is known as Benford's law, and a surprisingly large number of data sets, including lengths of rivers, street addresses, IRS Service files, and electricity bills, follow this law (We refer the interested reader to [Gleich, 2010] and [Amouzegar, Moshirvaziri, and Snyder, 2018]). Additionally, see [Wikipedia: https://en.wikipedia.org/wiki/Benford's law] for a more elaborate discussion of Benford's Law and its role in Data Science, Analytics, and Information Security. One of the primary applications of this phenomenon is fraud detection. An example of such an application is in recent Covid-19 data reported by various countries. For example, Farhadi and Lahooti [https://doi.org/10.3390/covid2040034], using Benford's Law and Goodness-of-Fit test, showed how several countries offered highly unreliable (or at least suspect) Covid results.

Mathematically, the first significant digit, $(n = 1)$, distribution of Benford's dataset is governed by (10), where D denotes the first significant digit. Thus, for $n = 1$,

$$f(d) = \mathrm{P}(D = d) = \log_{10}\left(1 + \frac{1}{d}\right) \ \ for\ d = 1, 2, \dots, 9. \tag{10}$$

Then for every <u>other</u> significant digit $(n > 1)$

$$f(d) = \sum_{k=a}^{b} \log_{10}\left(1 + \frac{1}{10k + d}\right) \ \ for\ d = 1, 2, \dots, 9. \ \ and\ a = 10^{n-2},\ \ b = -1 + 10^{n-1} \tag{11}$$

For example, the table below shows the distribution for $n = 2, \cdots, 6$ significant digits of Benford's law dataset.

[3] The *first significant digit* in a number is the first nonzero digit when reading from left to right. For example, the first significant digit of 218.81 is 2 and that of 0.0375 is 3. The first significant digit is always nonzero, but second and higher significant digits can be 0. The second significant digit of 0.102 is 0.

#	Prob (digit 2)	Prob (digit 3)	Prob (digit 4)	Prob (digit 5)	Prob (digit 6)
0	0.119679268596881	0.101784364644217	0.100176146939936	0.100017591505929	0.100001758918451
1	0.113890103407556	0.101375977447801	0.100136888117578	0.100013681135446	0.100001368036203
2	0.108821499005508	0.100972198137042	0.100097672594615	0.100009771195224	0.100000977158284
3	0.104329560230959	0.100572932110926	0.100058500283487	0.100005861685164	0.100000586284648
4	0.100308202267579	0.100178087627948	0.100019371096905	0.100001952605187	0.100000195415329
5	0.096677235802322	0.099787575692177	0.099980284947841	0.099998043955201	0.099999804550292
6	0.093374735783036	0.099401309944962	0.099941241749526	0.099994135735125	0.099999413689550
7	0.090351989269603	0.099019206561896	0.099902241415449	0.099990227944871	0.099999022833114
8	0.087570053578861	0.098641184154777	0.099863283859372	0.099986320584355	0.099998631980963
9	0.084997352057692	0.098267163678253	0.099824368995291	0.099982413653486	0.099998241133135

Cost Dataset – One of the interesting applications of Benford's law is detecting fraud in accounts payable reports. We collected hundreds of daily payments for a year, assessing the first five significant digits of each payment. The table below shows the results of d the distribution of the appropriate significant digits.

	Probability function for dataset digits				
Digits	d1	d2	d3	d4	d5
0	0.00000	0.11004	0.09707	0.10043	0.09851
1	0.30178	0.12206	0.10476	0.10139	0.09947
2	0.17347	0.09659	0.10476	0.09467	0.10043
3	0.12158	0.11341	0.10380	0.09659	0.09419
4	0.10043	0.10524	0.10812	0.10380	0.09851
5	0.08650	0.08554	0.10332	0.09995	0.10428
6	0.06583	0.10428	0.09082	0.09611	0.08650
7	0.05382	0.08938	0.09082	0.10139	0.11293
8	0.05094	0.09034	0.09611	0.10908	0.09226
9	0.04565	0.08313	0.10043	0.09659	0.11293

STATISTICAL TESTING

We conducted two effective statistical tests to detect fraudulent activity on the dataset.

Chi-Square Goodness-of-Fit Test – Testing the *Null Hypothesis* that the dataset is not subject to fraudulent activity. We applied the Chi-square goodness-of-fit test to the first significant digit of the dataset. The Chi-square goodness-of-fit test can be applied to discrete distributions and is reliable for relatively small dataset. The table below shows the results of the Chi-square "test."

$$\chi^2 = \sum_{i=0}^{9} \frac{(y_i - x_i)^2}{x_i} \tag{12}$$

Significant Digit	Chi-square Test
1	0.00125117
2	0.00646580
3	0.00286393
4	0.00169824
5	0.00634058

The value of Chi-squared, χ^2 derived by equation (12) above, for every significant digit, as illustrated in the table, is minimal. This indicates a large *p-value* and that the corresponding distribution of the observed dataset, the $y_i's$ are a good fit for the expected distribution, Benford's, the $x_i's, i = 1, \cdots, 5$.

Absolute Norm Test - We also computed the relative error using the absolute value norm (see table below)

ERROR	Relative error function for dataset digits				
Digits	d1	d2	d3	d4	d5
0	0.00000	0.08051	0.04633	0.00252	0.01507
1	0.00248	0.07171	0.03335	0.01252	0.00542
2	0.01486	0.11241	0.03749	0.05429	0.00423
3	0.02692	0.08700	0.03205	0.03418	0.05820
4	0.03635	0.04915	0.07929	0.03589	0.01492
5	0.09239	0.11524	0.03535	0.00028	0.04279
6	0.01663	0.11675	0.08631	0.03835	0.13498
7	0.07194	0.01076	0.08278	0.01495	0.12938
8	0.00422	0.03165	0.02568	0.09235	0.07724
9	0.00231	0.02193	0.02204	0.03238	0.12946

Both Chi-square and the relative error test do not show any significant deviation, and the null hypothesis that the dataset is Benford cannot be rejected. Thus, we can conclude the goodness of the data reported by the accounts payable.

Regenerative Method of Simulation (RMS) - There are side considerations in modeling and simulation methods designed to aid practitioners in improving upon the process and enhancing the accuracy of the simulated results. RMS provides a mechanism by which new samples are generated. Upon generation of each sample, usually, the simulator extracts statistics of interests from the current sample to be augmented with those collected from prior samples. The question is when to start generating a new sample. RMS suggests designating a particular state of the system initially. Then keep generating random data for the current sample and only stop when revisiting the same state (see [Haas, 2013]). Under certain assumptions, samples generated with RMS will constitute a series of reliable, independent samples. RMS has successfully been applied in inventory control systems and queueing and renewal point processes. The quality and accuracy of simulated results have improved due to considering sampling methods governed by the RMS strategy.

CONCLUDING REMARKS

In this paper, we demonstrated the effectiveness of spreadsheets in facilitating the learning process of modeling and simulation. The results obtained for our examples were consistent with those produced via analytical solutions and those found by specialized commercially available simulation software tools such as ExtendSim [Diamond, B., et al., 2017]. However, for many real-world applications, a simulation may be the only alternative to finding reasonable estimates because of the lack of theoretical foundation or the existence of a closed-form analytical solution for the particular case at hand. Not to mention that time and cost could factor in choosing simulation over other alternatives.

REFERENCES

Moshirvaziri, K., Amouzegar, M. A., and Rezayat, F. " Exploring the Birthday Attack / Paradox: A Powerful Vehicle Underlying Information Security," Proceedings of the Forty-Sixth Annual Meetings of the Western Decision Sciences Institute, Vancouver, BC, Canada, 2017.

U.S. Births, FiveThirtyEight.com https://github.com/fivethirtyeight/data/tree/master/births, last visited 10/1/2022

Gleich, David, "Birthday distribution," David Gleich's Notebook. Stanford University, 2010. Source: http://www.stanford.edu/~dgleich/notebook/2009/04/birthday_distribution.html

Sun, Jared, "The Birthday Paradox: A Simple and Surprising Probability Problem," Aug. 2011, Cluster6

Amouzegar, M. A., Moshirvaziri, K., and Snyder, D., "Benford's Law and its Application to Modern Information Security," Proceedings of the Forty-Seventh Annual Meetings of the Western Decision Sciences Institute, Kauai, Hawaii, 2018.

Wikipedia: "Benford's Law" Generalization of the digits beyond the first, https://en.wikipedia.org/wiki/Benford%27s_law Visited 10/9/2022

Farhadi, N. and Lahooti, H., "Forensic Analysis of COVID-19 Data from 198 Countries, Two Years after the Pandemic Outbreak," COVID 2022, 2, 472–484. https://doi.org/10.3390/covid2040034

Haas, P.J. (2013). Regenerative Simulation. In: Gass, S.I., Fu, M.C. (eds) Encyclopedia of Operations Research and Management Science. Springer, Boston, MA. https://doi.org/10.1007/978-1-4419-1153-7_1163

Diamond, Bob; Lamperti, Steve; Krahl, Dave; Nastasi, Anthony, and Damiron, Cecile "ExtendSim User Guide Release," Published by Imagine That, Inc., 2017