

TEACHING RISK ANALYTICS USING R

Ravi Doddavaram
Canan Gunes Corlu

Metropolitan College
Boston University
1010 Commonwealth Avenue
Boston, MA, 02215 USA

ABSTRACT

We discuss our experience with using R, which is a free software that is particularly suitable for computer simulation, in a risk analytics course offered to students having different experience levels and technical sophistication. We highlight relevant packages in R that we found useful for the purposes of input modeling and stochastic optimization along with several examples. We also introduce a new package that allows the simulation of decision trees. Student feedback regarding the use of R in comparison to a spreadsheet-based platform has been very positive.

1 INTRODUCTION

Lawson and Leemis (2017) argue that R is a suitable software for teaching computer simulation. First, its capabilities are found to help develop student intuition. Second, it is particularly suitable for looping and for statistical analysis. Third, it has a high-level customizable graphical functionality that allows easy representation of simulation results. Our goal in this paper is to share our experiences with teaching Monte Carlo simulation (as part of a risk analytics course) with R to a diverse body of graduate students. Specifically, we discuss the packages that we found useful for input modeling and stochastic optimization. We also introduce a new package that allows the user to simulate decision trees. Suggestions are provided on how to make the course more attractive for students having various experience levels and taking the course in different delivery formats.

1.1 Metropolitan College at Boston University

Metropolitan College (MET) is one of the 17 degree-granting schools/colleges at Boston University. The mission of the college is to provide high-quality and practical graduate education in flexible modes. The Administrative Sciences Department at MET offers a variety of MS degrees in several fields including Applied Business Analytics, Supply Chain Management, Enterprise Risk Management, Financial Management, Statistical Practice & Data Science and Global Marketing Management.

We have both domestic and international students with a variety of backgrounds. Our students come in different skill levels depending on their work experience. Our full-time students are those who join the program right after they receive their bachelor degrees. Our part-time students are usually have some work experience and therefore are more sophisticated compared to the full-time students. We offer our courses both in face-to-face and online formats. The challenge is to make the course attractive for both student body considering the difference between their experience levels and technical sophistication.

1.2 Risk Analytics Course

Enterprise Risk Analytics is one of the specialization courses of the MS in Applied Business Analytics program and the MS Supply Chain Management Program. It is also offered as an elective course for other MS programs such as Enterprise Risk Management and Global Marketing Management. Our main goal in this course is to prepare our students being able to develop business models and use them to support decision making. Therefore, after the completion of this course, we expect our graduates to be comfortable simulation modellers and have the ability to make sound decisions using the simulation results. Additionally, we would like our graduates to be sound programmers who are comfortable with developing simulation codes from scratch. We strive to find a fine balance in the course among applications, problem solving, and coding.

We use R as the software in this course. It is important to note that our claim in this paper is not that R is the “best” software for teaching risk analytics. There are certainly several alternatives that exist, among which spreadsheet simulation and Excel add-ins such as Crystal Ball, ASP, @Risk are the most popular (Jain 2014). In fact, we were relying on Excel add-ins before we decided to move to R. However, a careful review of the job postings in the area of analytics made it clear that graduates with the knowledge of programming languages such as Python and R are in high demand and, in fact, these programming languages are a must for the graduates to have in their toolbox. We decided to use R in all of our analytics courses including risk analytics, and made R a key component of our curriculum. We should also mention that the move to R benefited students because students developed a modeling mindset and they were able to develop intuition easily. This was also observed in Lawson and Leemis (2017).

Topics covered in the course include developing a stochastic simulation model, the major components of a simulation model including input modeling (independent and identically distributed input data as well as correlated input data) and output analysis, simulation optimization, and decision trees with simulation. Many of these topics can be covered using Excel add-ins; however, students miss the underlying modeling nuances and coding experience. We spend about two-third of class time to teach modeling concepts and about one-third of the class time to teach R programming skills.

One of our challenges is to make the course effective for those students who might not have enough experience on working with R. Although students take a pre-requisite business analytics foundations course covering the fundamentals of R, many of them are not fluent enough to write a simulation model from scratch, perform simulation optimization or combine decision trees with simulation in R environment. In this paper, we discuss our experience with using some of the readily available packages in R for Monte Carlo simulation and some packages that we wrote and shared with our students that will help them achieve the aforementioned tasks. Those packages are available at <https://github.com/wxcode>.

The rest of this paper is organized as follows. Section 2 discusses the relevant literature. Section 3 presents the introductory material for the course. Section 4 presents input modeling and relevant R packages along with examples. Section 5 discusses stochastic optimization and Section 6 presents decision trees with simulation. Section 7 provides a brief overview of students’ feedback and Section 8 concludes.

2 LITERATURE REVIEW

Computer simulation has been taught for years at both undergraduate level and graduate level, and both to engineering students and business school students. Several software and programming languages have been used for teaching simulation. As such, tutorials exist that discuss available tools and programming languages. For example, Table 2 in Nance and Overstreet (2017) represents 48 simulation languages and the number of tutorials that exist for each language. Dagkakis and Heavey (2016) provide a survey and comparison of open-source software tools focusing on discrete-event simulation. Authors argue that the commercial software are limited by a number of factors including cost, flexibility and reusability whereas open-source tools do not suffer from those limitations. de Mesquita et al. (2019) perform a survey of

simulation educators and find that for many educators their choice of the software depends on the cost of the software.

Simulation education in business schools has been particularly discussed in the literature mainly because of the challenges experienced with regards to the quantitative and technological skills possessed by business school students. For example, Saltzman and Roeder (2013) discuss their experiences in teaching simulation to business students both at undergraduate and graduate level. Authors mainly use Microsoft Excel and Arena in their courses. A close look at the literature on the use of simulation software in business schools reveals that there is no clear software of choice. For example, Jain (2014) conducts a survey of teaching of simulation software at leading business schools and concludes that tools used for Monte Carlo simulation (MCS) and discrete-event simulation are vast. For MCS, add-ins such as Crystal Ball and @Risk have been reported.

Lawson and Leemis (2015) argue that the choice of a computing platform for an introductory simulation course depends on a number of factors including the ease-of-use, cost, and ability to customize. If the course focuses on the modeling aspect, then commercial software is recommended. However, if the emphasis is on programming and probabilistic and statistical aspects of simulation, then R is recommended. The following aspects of R is considered to be suitable for computer simulation: (i) R allows looping and functions, which make it suitable for simulations; (ii) R has built-in random variate generation functions, which make input modeling straightforward; (iii) R has high-level graphical functionality that is easy to customize. Authors demonstrate the use of 'ssq', which is an R package developed by the authors to perform a discrete-event simulation of a single-server queue.

There are a few R packages that can be used for simulation. For example, Brock and Slade (2015) introduced the 'poisson' package, which simulates both homogeneous and non-homogeneous Poisson processes. 'simmer' is designed by Ucar et al. (2017) as a generic simulation framework. 'simEd' is designed by Lawson and Leemis (2017) with a focus on simulation pedagogy. 'simEd' includes variate generation functions for discrete and continuous distributions; specific functions implementing single and multiple server queuing models; data sets for queuing input modeling; functions that allow the visualization of cumulative distributions functions and their inverse, probability density and probability mass functions.

We contribute to the existing literature as follows. We provide a guideline for instructors who would like to move to R for teaching simulation from another language that they are currently using. We discuss our experience with some of the readily available packages that can be used for input modeling and stochastic optimization. We further introduce an R package that allows students perform simulation on decision trees. This course was taught in two different delivery formats (online and on-campus) and the student profiles in these two cohorts are very different from each other - the online class typically had students with prior work experience and while slightly detached with the quantitative elements, whereas the face-to-face version had fresh graduates with limited industry experience. The paper also presents a discussion around how to make the course material attractive for both cohort of students.

3 INTRODUCTORY MATERIAL

We begin teaching the course by introducing basic statistical constructs that help understand generative processes and difference between a maximum likelihood estimate and a posteriori inference. There are several exercises which make students aware of relationships across various distributions. We provide examples of use cases where certain distributions best suit the modeling framework. For example, for a passenger no-show rate model, we explain why a *lognormal* distribution might be a likely candidate and for a financial return why a fat-tailed distribution like *t-Distribution* might represent the process accurately. We use this opportunity to introduce the R language to students to simulate these distributions and create their own custom distributions if necessary.

One of the learnings we had here was that most of the students, particularly those who worked in the industry for a significant amount of time, are accustomed to using spreadsheets for modeling. We found it useful to replicate the paradigm of spreadsheets using data frames in R along with the various *apply*

functions that mimic a spreadsheet approach. However, as we enter more complex problems in the course, we also expose students to programmatic/system modeling techniques using a functional approach in R. This delineation was extremely helpful in gaining the students buy-in to invest more time in using R and exploring more problems on their own. Following the introductory material we start discussing on ways to model the inputs in a simulation model.

In this part of the course, we also introduce the notion of systems thinking and modeling of business problems. A very simple block diagram as shown in Figure 1 teaches students at a very high level how we are going to approach problems in decision making under uncertainty and risk. The block diagram represents a model for decision making in a business context using data. One of the key takeaways here is to explain to students the sources of uncertainty (inputs and the system itself).

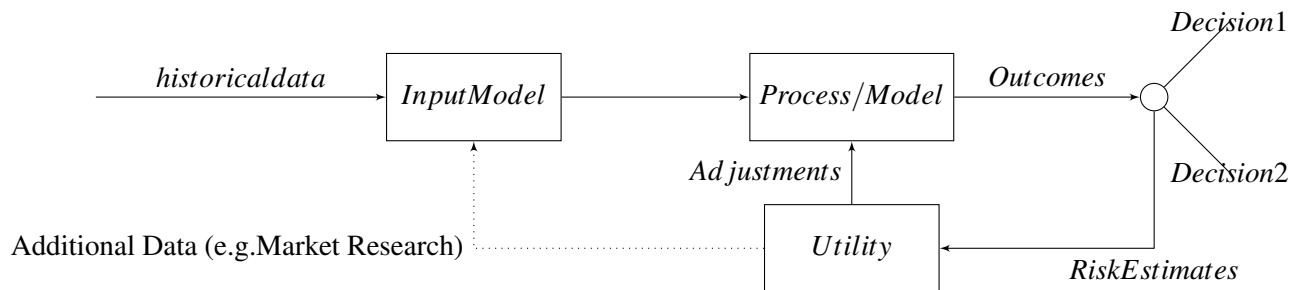


Figure 1: System model for decisions under uncertainty.

4 INPUT MODELING

Independent and Identically Distributed Data: We use R extensively in estimating the quality of fit for different types of inputs and sources of data. We show students how to fit distributions for simulation models that arise in different domains including finance, supply chain management, and market diffusion models. In particular, we use the *fitdistrplus* library in R which provides an extensive set of tools for probability distributions. In the case of univariate distributions, we explain to students about Kolmogorov-Smirnov (KS Test) and moment matching tests and how *fitdistrplus* library can be used to perform these tests.

Correlated Data: This section of the course is an extension of understanding input variables but this is where students understand the importance of correlation in input variables. We briefly discuss financial markets and particularly the 2008 crisis and the Gaussian copula (Felix 2009). We also discuss different types of risks; e.g., model risk, portfolio risk, and tail dependence. While it is tempting to get students to think about some of the newer techniques with larger parameterization and time-varying characteristics, we stick to cases where we provide a wider range of applicability and not limit ourselves to the financial industry. Before we introduce simulating these in R, we also introduce multivariate distributions, covariance and correlation matrices, their structure and how to construct them.

R provides a wide variety of tools to simulate correlated random variables. Although copulas are popular for modeling dependencies, we use two packages that make it easier for students to focus on the business aspects of modeling than getting the stochastic relationships correct. The first package is the library 'MASS' provides probably the more widely used function *mvrnorm* for simulating correlated random normal variables. This library, however, limits itself to multivariate normal distributions. To overcome this limitation, we utilize another package that uses NORTARA (Normal to Anything through Retrospective Approximation) algorithm (package goes by the same name) to generate any arbitrary joint distribution with specified correlation matrix. To understand the implications of correlation we use an example of a foreign exchange hedge between two currencies. We look at monthly returns between USD-EU and USD-CHF, which in most months have a strong negative correlation. We use the daily returns as our input data and

the code in Listing 1. Figure 2 demonstrates how a portfolio of equally weighted mix of USD-CHF and USD-EU with correlation has significantly narrow distribution and lower risk measured as a VAR or CVar metric.

Listing 1: R code – Modeling correlation as a multivariate normal.

```

1
2 library(MASS)
3 library(fitdistrplus)
4 usd_chf #Daily returns on USD to Swiss Francs
5 usd_eu # Daily returns on USD to Euros
6
7 eu_chf_cor <- cor(as.numeric(usd_chf),as.numeric(usd_eu))
8 #[1] -0.8735658
9
10 chf_parms <- fitdist(usd_chf/sum(usd_chf),'norm')
11 eu_parms <- fitdist(usd_eu/sum(usd_eu),'norm')
12
13 nTrials <- 10000
14 chfSims <- rnorm(nTrials,chf_parms$estimate['mean'],
15                 chf_parms$estimate['sd'])*sum(usd_chf)
16
17 euSims <- rnorm(nTrials,eu_parms$estimate['mean'],
18                eu_parms$estimate['sd'])*sum(usd_eu)
19
20 muV <- c(mean(chfSims),mean(euSims))
21
22 CovMat <- matrix(c(sd(chfSims)**2,
23                    sd(chfSims)*sd(euSims)*eu_chf_cor,
24                    sd(chfSims)*sd(euSims)*eu_chf_cor,
25                    sd(euSims)**2),2,2)
26
27 corSims <- mvrnorm(nTrials,muV,CovMat)
28
29 ggplot()+geom_density(aes(chfSims,col='CHF'),linetype='dashed') +
30   geom_density(aes(euSims,col="EU"),linetype='dashed') +
31   geom_density(aes(euSims+chfSims,col="EU+CHF")) +
32   geom_density(aes(corSims[,1]+corSims[,2],col="EU+CHF w Corr")) + xlab("<-
   Daily Returns") +
33   theme_bw()

```

5 STOCHASTIC OPTIMIZATION USING SIMULATION

In this part of the course, we start introducing the applications with all the background that we have built in building prescriptive models using uncertainty analysis. We introduce this through a functional approach, where we demonstrate the modeling as a sequence of functions. We showcase how to convert a deterministic model into a stochastic model by modeling the input parameters as generative processes (functions) based on the distributions that have been fit in the previous modules. This is a key insight that students learn in structuring their thinking and going on to use this in their careers.

One of the problems that we had to overcome was to find a versatile solver that does constrained optimization and works across different class of optimization problems (such as a multivariate objective

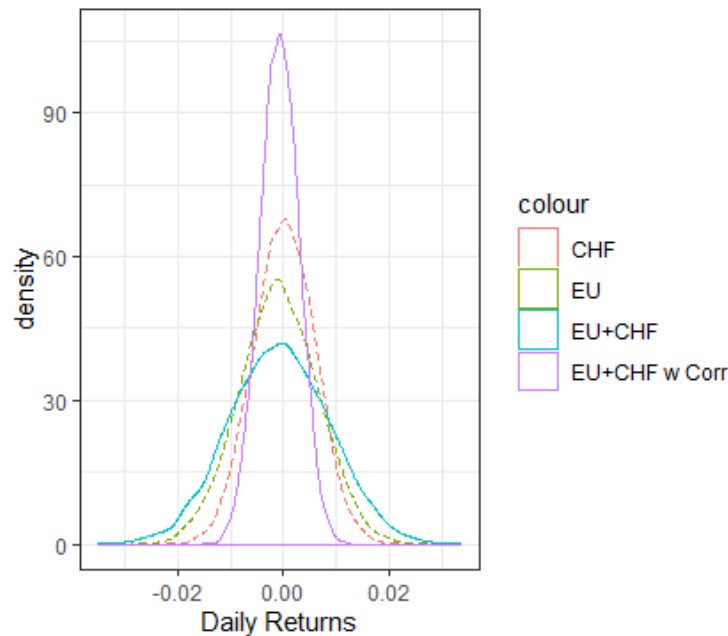


Figure 2: Daily returns modeled with and without correlation.

function with stochastic constraints etc). We found the DEOptim (David Ardia 2004) to be quite suitable for stochastic optimization through simulation and simple enough for a student without much programming experience to use. DEOptim uses the differential evolution optimization algorithm and iterates to arrive at the optimal solution. Although it does not always find the optimal solution, we still prefer DEOptim because of the ease of translating the business problem into a constrained optimization problem.

We use two examples in demonstrating optimization under uncertainty. In the first example, we use the airline revenue management problem as an example where we model the passenger cancellations and the overbooking capacity as the uncertain inputs. This gives a chance to see statistically the revenue expectations, likely penalties, and refunds that the airline revenue management system can handle. To keep the problem tractable, we deliberately avoid operating costs such as fuel and personnel but highlight to students how to incorporate them in the extended version of the problem. In the second example, we demonstrate a simple supply chain example of a store ordering widgets with uncertain lead time and uncertain demand. The goal is to minimize the total inventory holding cost while maintaining a target service level. Through this example we demonstrate the use of chance constraints in the optimization and the cost sensitivity in providing a higher service level guarantee. Students appreciate this model and often remark that they can immediately put this to use in their workplace.

6 DECISION TREES

There are several readily available packages for modeling decision trees; however, they usually require the use of a commercial software. We wanted an open source and programmable alternative in R instead of spreadsheet based models. To this end, we built an open source package for the course, available on Github (Doddavaram 2019). We heavily use an underlying library Data.Tree (Christoph Glur (R interface) 2019) package for building the underlying data structures for building trees. We demonstrate this approach with an example in Listing 2. We borrowed this example from (Albright and Winston 2015). The problem is classical market entry decision for a product with a range of outcomes. There is an opportunity to run a test market and evaluate the performance based on the outcome of the test market. We include a range of outcomes modeled as a normal distribution and then come up with the most likely estimate of an optimal

decision based on Expected Monetary Value (EMV) defined as

$$EMV = \sum p_i * X_i$$

where p_i is the probability of event i , and X_i is the payoff. Our goal is to give a flavor on how to use the library; therefore, we do not provide all the details of the problem and its solution. The library automatically calculates the EMVs of all the nodes and plots the decision tree (see Figure 3). The package provides built-in functions that navigate the tree to calculate the payoffs and utilities automatically. The package also provides a way to simulate the scenarios in the payoffs to estimate the uncertainty in any decision being made.

Listing 2: R code – Simulation of decision trees.

```

1 # Create Decision Tree
2 DT <- Tree$new(name = 'RunTest', type = 'decision')
3 ###
4 # A Decision Tree can have 3 types of node. Decision Node, Chance Node,
5 # Terminal Node
6 # Each Node will have to be instantiated with a parent node, route (Yes/No),
7 # Name of the Node, and probability of entering the node. For Terminal nodes<-
8 # there needs to be a payoff, and for Chance nodes there is always a cost of
9 # running the experimentation
10 #
11 #
12
13
14 ### Do not Run Test Market#####
15 DT$addDecision(name = 'EnterNM', route = 'No', pNode = 'RunTest', prob = 1)
16 DT$addTerminal(name = 'Stop', payoff=0, pNode = 'EnterNM', prob=1)
17 DT$addChance(name = 'NM_Outcomes', pNode = 'EnterNM', cost= -7.0)
18
19 DT$addTerminal(name = 'A', pNode = 'NM_Outcomes', prob=0.2, payoff=Awful)
20 DT$addTerminal(name = 'F', pNode = 'NM_Outcomes', prob=0.35, payoff=Fair)
21 DT$addTerminal(name = 'G', pNode = 'NM_Outcomes', prob = 0.45, payoff =Great)
22
23 DT$plot_graph()
24
25 #####RUN the Test Market#####
26 DT$addChance('TM_Outcomes', pNode = 'RunTest', route='Yes', cost = -0.1)
27 ###Test Market Outcome was Great#####
28 DT$addDecision(name = 'Outcome_TG', pNode = 'TM_Outcomes', prob = 0.369)
29 DT$addTerminal(name = 'Stop', pNode = 'Outcome_TG', prob = 1, payoff=0)
30
31 DT$addChance(name = 'TG_EnterNM', pNode = 'Outcome_TG', cost = -7.0)
32
33 DT$addTerminal(name = 'NM_G', pNode = 'TG_EnterNM', prob =0.7805, payoff=Great<-
34 )
35 DT$addTerminal(name = 'NM_F', pNode = 'TG_EnterNM', prob = 0.1707, payoff=Fair<-
36 )
37 DT$addTerminal(name = 'NM_A', pNode = 'TG_EnterNM', prob = 0.0488, payoff=<-
38 Awful)
39
40

```

```

37 DT$plot_graph()
38
39 ###Test Market Outcome was Fair#####
40 DT$addDecision(name = 'Outcome_TF', pNode = 'TM_Outcomes',prob = 0.4125)
41 DT$addTerminal(name = 'Stop',pNode = 'Outcome_TF', prob = 1,payoff=0)
42
43 DT$addChance(name = 'TF_EnterNM',pNode = 'Outcome_TF', cost = -7.0)
44
45 DT$addTerminal(name = 'NM_G',pNode = 'TF_EnterNM', prob =0.2386,payoff=Great↔
    )
46 DT$addTerminal(name = 'NM_F',pNode = 'TF_EnterNM', prob = 0.4836,payoff=Fair↔
    )
47 DT$addTerminal(name = 'NM_A',pNode = 'TF_EnterNM', prob = 0.2327,payoff=↔
    Awful)
48
49 ###Test Market Outcome was Awful#####
50 DT$addDecision(name = 'Outcome_TA', pNode = 'TM_Outcomes',prob = 0.2185)
51 DT$addTerminal(name = 'Stop',pNode = 'Outcome_TA', prob = 1,payoff=0)
52 DT$addChance(name = 'TA_EnterNM',pNode = 'Outcome_TA', cost = -7.0)
53
54 DT$addTerminal(name = 'NM_G',pNode = 'TA_EnterNM', prob =0.2059,payoff=Great↔
    )
55 DT$addTerminal(name = 'NM_F',pNode = 'TA_EnterNM', prob = 0.4005,payoff=Fair↔
    )
56 DT$addTerminal(name = 'NM_A',pNode = 'TA_EnterNM', prob = 0.3946,payoff=↔
    Awful)
57
58 #####Plot the Decision Tree#####
59 DT$plot_graph()
60
61 #####Update the Payoff and plot the payoff diagram↔
    #####
62 DT$update_payoff()
63 DT$plot_graph()

```

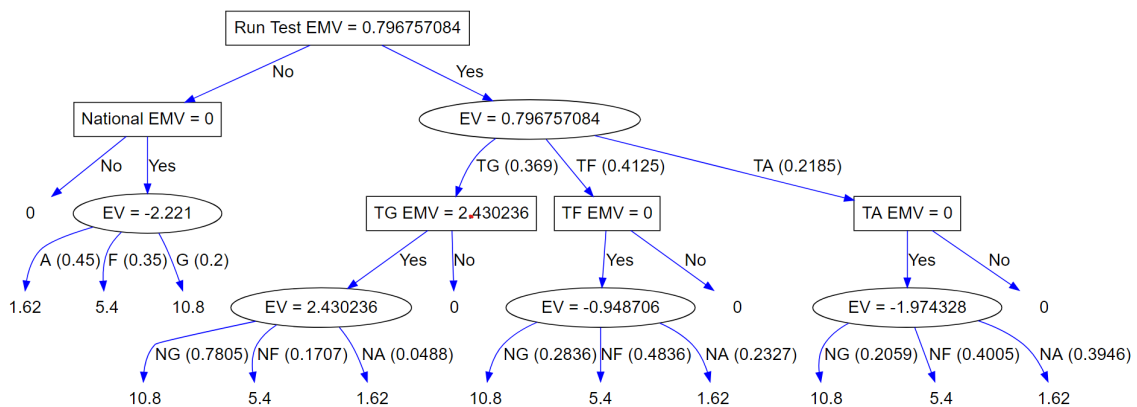


Figure 3: Decision tree with payoffs.

One of the advantages in using an R based approach is the ability to run simulations to model the uncertainties in the outcomes of the decision trees. We highlight to students a quick corollary to this approach is a bootstrapped methods (e.g., random forest) where we might just use historical data to bootstrap and arrive at an expectation for the optimal solution. While the concept of decision trees is very widely covered in many basic courses of decision making, in this course they are usually limited in guiding methodologically the choice of data and probabilities that go into building a decision tree. We realized this shortcoming and start addressing this by introducing the basics of graph modeling and causal structure and Bayesian inference.

7 STUDENT FEEDBACK

This course is taught in two delivery formats: face-to-face and online. The face-to-face version was taught over a period of 13 weeks 2 hours each, whereas the online version was taught over 6 weeks with 1 hour sessions each. The online class typically had students with prior work experience and while slightly detached with the quantitative elements, whereas the face-to-face version were fresh graduates with no industry experience.

In order to understand students' experience throughout the course, we conducted surveys anonymously at the end of the course using a university wide tool. The survey participation for every offering of the course ranged from 50% to 75% of the class size. The average size of the class for the online offering was about 22 and for the face-to-face version of the class of about 35. The feedback from students fell in two broad categories: Modeling of business problems with uncertainty and programming using R vis-a-vis Microsoft Excel based add-ins. A class of students who already work in the predictive analytics teams within their organization appreciate the modeling element of the course. There have been several positive comments about using most of the concepts that they learnt in this course in their daily professional lives. Some students expressed a desire to have a separate module on system modeling and simulation before starting to using R. The second category of feedback pertains to R programming language. The student response here has been mixed. Depending on the type of prior exposure the student had in programming this turned out to be a challenging or exciting experience. Overall there is a definite liking towards moving away from a Excel based platform since the scope for learning the underlying concepts is not tested to the extent as it is with R based approach.

Our recommendation is two-fold. For the online students, it is important to immerse the students into the quantitative elements of the course early on. It was crucial to navigate the difficult process upfront for the students to appreciate the rest of the course and have a rewarding experience afterwards. For the face-to-face students, it is important to give them realistic examples from business consulting so they appreciate the application of the skill sets they picked up. Case studies are one way to accomplish this where they deliberate a realistic business situation and arrive at appropriate solution with a quantitative backing.

8 CONCLUSION

This paper discussed authors' experiences in moving to R language for teaching an introductory simulation course. R is chosen as the preferred language because both modeling and programming elements were emphasized in the course in order to prepare our students majoring in Applied Business Analytics to the needs of the market. Among the topics discussed are input modeling, stochastic optimization, and simulation of decision trees in R. Some packages that we use were readily available while we have created the package that performs simulation on decision trees from scratch.

Another aspect of this paper is the discussion around how this material is perceived by a mix of online and face-to-face students with varying industry experience, programming and statistical skills. We provide recommendations about how the material can be introduced to both cohort of students. We hope that instructors who plan to teach an introductory simulation course using R will find this paper helpful.

REFERENCES

- Albright, S. C., and W. L. Winston. 2015. *Business Analytics: Data Analysis and Decision Making, 5e, 2015, Cengage Learning*. Cengage.
- Brock, K., and D. Slade. 2015. “poisson: Simulating Homogenous & Non-Homogenous Poisson Processes”. *R package version 1*.
- Christoph Glur (R interface), e. a. 2019. *data.tree: General Purpose Hierarchical Data Structure*.
- Dagkakis, G., and C. Heavey. 2016. “A Review of Open Source Discrete Event Simulation Software for Operations Research”. *Journal of Simulation* 10(3):193–206.
- David Ardia, Katharine Mullen [aut, cre], Brian Peterson [aut], Joshua Ulrich [aut], Kris Boudt [ctb] 2000–2004. “DEOptim”. <https://cran.r-project.org/web/packages/DEoptim/index.html>.
- de Mesquita, M. A., B. C. da Silva, and J. V. Tomotani. 2019. “Simulation Education: A Survey of Faculty and Practitioners”. In *Proceedings of the 2019 Winter Simulation Conference*, edited by N. Mustafee, K.-H. Bae, S. Lazarova-Molnar, M. Rabe, C. Szabo, P. Haas, and Y.-J. Son, 3344–3355. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Ravi Doddavaram 2019. “AD616 Package”. <https://github.com/WXcode>.
- Felix, Salmon 2009. “The Formula that killed the wall street”. <https://www.wired.com/2009/02/wp-quant/?currentPage=all>.
- Jain, S. 2014. “Teaching of Simulation at Business Achools”. In *Proceedings of the 2014 Winter Simulation Conference*, edited by A. Tolk, S. Y. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley, and J. A. Miller, 3684–3695. Institute of Electrical and Electronics Engineers, Inc.
- Lawson, B., and L. M. Leemis. 2015. “Discrete-event Simulation using R”. In *Proceedings of the 2015 Winter Simulation Conference*, edited by L. Yilmaz, W. K. V. Chan, I. Moon, T. M. K. Roeder, C. Macal, and M. D. Rossetti, 3502–3513. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Lawson, B., and L. M. Leemis. 2017. “An R Package for Simulation Education”. In *Proceeding of the 2017 Winter Simulation Conference*, edited by W. K. V. Chan, A. D’Ambrogio, G. Zacharewicz, N. Mustafee, G. Wainer, and E. Page, 4175–4186. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Nance, R. E., and C. M. Overstreet. 2017. “History of Computer Simulation Software: An Initial Perspective”. In *Proceedings of the 2017 Winter Simulation Conference*, edited by W. K. V. Chan, A. D’Ambrogio, G. Zacharewicz, N. Mustafee, G. Wainer, and E. Page, 243–261. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Saltzman, R. M., and T. M. Roeder. 2013. “Perspectives on Teaching Simulation in a College of Business”. In *Proceedings of the 2013 Winter Simulations Conference*, edited by R. Pasupathy, S.-H. Kim, A. Tolk, R. Hill, and M. E. Kuhl, 3620–3629. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Ucar, I., B. Smeets, and A. Azcorra. 2017. “simmer: Discrete-Event simulation for R”. *arXiv preprint arXiv:1705.09746*.

AUTHOR BIOGRAPHIES

RAVI DODDAVARAM is a senior lecturer at Boston University and he holds an MBA in Finance and Analytics from Sloan School of Management and a Masters in Electrical Engineering specializing in Signal Processing from Technical University of Eindhoven. His research interests are in non-parametric models and uncertainty estimation techniques in Finance and Enterprise Risk. He enjoys building products and in the past has successfully built a Telecommunication Hardware business and exited it. He can be reached at ravid@bu.edu

CANAN GUNES CORLU is an Associate Professor of Administrative Sciences Department of Metropolitan College at Boston University. She holds a Ph.D. in Operations Management from the Tepper School of Business at Carnegie Mellon University. Her research interest is in the area of the design and analysis of stochastic simulations with applications in operations management. Her email address is canan@bu.edu.