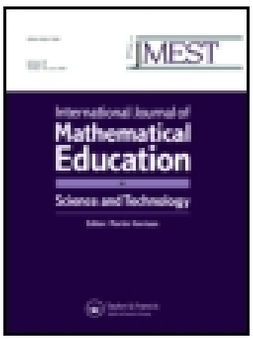

International Journal of Mathematical Education in Science and Technology

ISSN: 0020-739X (Print) 1464-5211 (Online) Journal homepage: http://www.tandfonline.com/loi/tmes20

Introducing computational thinking through hands-on projects using R with applications to calculus, probability and data analysis

Nadia Benakli, Boyan Kostadinov, Ashwin Satyanarayana & Satyanand Singh

To cite this article: Nadia Benakli, Boyan Kostadinov, Ashwin Satyanarayana & Satyanand Singh (2016): Introducing computational thinking through hands-on projects using R with applications to calculus, probability and data analysis, International Journal of Mathematical Education in Science and Technology, DOI: 10.1080/0020739X.2016.1254296

To link to this article: http://dx.doi.org/10.1080/0020739X.2016.1254296

Published online: 01 Dec 2016.

INTERNATIONAL JOURNAL OF MATHEMATICAL EDUCATION IN SCIENCE AND TECHNOLOGY, 2016
http://dx.doi.org/10.1080/0020739X.2016.1254296

Taylor & Francis
Taylor & Francis Group

Introducing computational thinking through hands-on projects using R with applications to calculus, probability and data analysis

Nadia Benakli[a], Boyan Kostadinov [a], Ashwin Satyanarayana[b] and Satyanand Singh[a]

[a]Mathematics Department, NYC College of Technology, CUNY, Brooklyn, NY, USA; [b]Computer Systems Technology Department, NYC College of Technology, CUNY, Brooklyn, NY, USA

ABSTRACT

The goal of this paper is to promote computational thinking among mathematics, engineering, science and technology students, through hands-on computer experiments. These activities have the potential to empower students to learn, create and invent with technology, and they engage computational thinking through simulations, visualizations and data analysis. We present nine computer experiments and suggest a few more, with applications to calculus, probability and data analysis, which engage computational thinking through simulations, visualizations and data analysis. We are using the free (open-source) statistical programming language R. Our goal is to give a taste of what R offers rather than to present a comprehensive tutorial on the R language. In our experience, these kinds of interactive computer activities can be easily integrated into a smart classroom. Furthermore, these activities do tend to keep students motivated and actively engaged in the process of learning, problem solving and developing a better intuition for understanding complex mathematical concepts.

ARTICLE HISTORY
Received 30 June 2015

KEYWORDS
Technology in mathematics education; scientific programming and simulations using R; visualization of Weierstrass functions; Monte Carlo games and simulations; data analysis with R; computational probability with R; computational problem solving

1. Introduction

Computational thinking, in this age of technology, should be considered a fundamental analytical skill in education, along with reading, writing and arithmetic. This is a vision for the twenty-first century classroom supported by the National Research Council of the Academy of Sciences [1]. Many important applied and pure research questions across the sciences involve computing as well as theory. We believe that computing brings additional insight and understanding that theory alone cannot achieve. There are studies that support this view, see the work of Thomas and Lin [2], and the references therein. It is our opinion that the ever-increasing use of computational devices must be supported by widespread promulgation of computational thinking across the STEM disciplines, starting at the K-12 level, and further supported and enhanced by college curricula.

The aim of this paper is to present 10 technology-infused, hands-on, computational projects with applications to calculus, probability and data analysis, using the free

CONTACT Boyan Kostadinov bkostadinov@citytech.cuny.edu

computational platform R, maintained by the R Project [3]. One of our goals is to give a taste of the rich functionality that R offers rather than to present a comprehensive introduction to the R language. Given the rapid change of modern technologies, this article addresses a need to emphasize presentation of difficult concepts using a modern technology such as R. We focus on the mathematical and programming content of these computational projects. One of our main objectives is to offer a collection of carefully solved and classroom tested mathematical problems, using R, with code provided, so that they can be incorporated into syllabi, reproduced and adapted quickly to the needs of educators interested in experimenting with computational thinking in their classes.

We can encourage computational thinking in our students through hands-on computational activities that empower them to do mathematics and solve problems with technology. We offer a number of projects that engage computational thinking and may attract students with diverse interests and backgrounds. Of course, technology is not a substitute for knowledge. Technology is a tool to gain insights into complex problems and it can be a valuable tool to better understand difficult concepts and learn the scientific method of inquiry. This is a pedagogical approach strongly supported by the the National Research Council of the National Academy of Sciences [4, p.66]:

> Modern science often uses computational models that are based on scientific principles and whose use depends on visualizations. Understanding these models requires computational thinking - scientific models and visualizations allow students to visualize the computations that are going on in near real time ... students learn better by seeing models and interacting with them, and that by exploring the model in a spirit of inquiry, they learn about the science in the model in much the same way that scientists learn about nature by using the scientific method ... students can learn complicated, deep concepts this way.

Additionally, the National Council of Teachers of Mathematics has also expressed their strong support for the use of technology in mathematics education [5, p.5]:

> An excellent mathematics program integrates the use of mathematical tools and technology as essential resources to help students learn and make sense of mathematical ideas, reason mathematically, and communicate their mathematical thinking.

Using modern technology offers students the opportunity to learn how to better communicate and present mathematics, skills of ever-increasing importance, both in academia and industry. An added pedagogical benefit of introducing computational projects is the opportunity to get a better idea of how well our students have mastered the subject matter, and which mathematical concepts create difficulties for them.

In our classroom experience, these kinds of interactive computer activities tend to keep students motivated and actively engaged in the process of learning. Since spring 2014, we have been using the R programming language, together with RStudio [6] (as an integrated development environment) in several upper-level mathematics courses, including Stochastic Models, Numerical Methods, Probability and Statistics, Computational Statistics, Financial Mathematics and Risk Management, offered by the Applied Mathematics and Mathematics Education programs. We also have plans to introduce R at the lower-level courses on introductory probability and statistics, offered to associate degree as well as baccalaureate degree students from a number of STEM fields. The introductory statistics course has changed significantly in recent years, with a greater focus on active learning, use of technology for conceptual understanding and analysis of raw and real data [7].

Based on our classroom experience, our observation is that students greatly benefit from the kind of computational projects we are presenting in this paper. They start by playing an actual pen and paper game of 'darts', which really helps them better appreciate and relate to the computer implementation of the same Monte Carlo game. All of our projects are scaffolded into many smaller parts, so that the students are guided step by step and they can more confidently go through the entire solution. An important feature of our projects is that we mix computational and mathematical questions, so that the students have the opportunity to investigate the same questions from both mathematical and computational perspectives. This is a key component that builds students' confidence, since they can compare their computational and mathematical results, and be sure that they are on the right track. The visualizations are also a key component of these projects, since in our experience they really help students build very good intuition, and they also offer the students a way to start investigating the problem without fear. One of the authors has industry experience and has a first-hand knowledge of how important coding skills are in the industry, in addition to the pure mathematical skills, and they both go hand-in-hand. Coding helps students pay attention to details and truly understand the mathematical problem, as well as implement a solution approach following the correct mathematical logic. We see coding as a form of computational problem solving, which is intertwined with the purely mathematical form of problem solving. These are the skills and benefits we want students to take away from these projects.

2. Installing and learning R

R is a free (open-source) scientific programming language, maintained by the R Project. R can be downloaded from [3]. Precompiled binaries are available for Mac, Linux and Windows. R comes with a simple user interface. A more sophisticated integrated development environment (IDE) is offered by RStudio, which has become very popular because of its enhanced report generation capabilities, unifying computing with R and mathematical typesetting with LATEX. RStudio is also free and binaries are available for Mac, Linux and Windows. RStudio can be downloaded from [6]. R has evolved into a powerful computational platform for experimenting with mathematical ideas through graphical explorations, simulations and animations of deterministic and stochastic models, statistical computing and data analysis, and more recently, web-based interactive apps. According to the KDnuggets software poll [8], R has been getting increasingly popular and it is now the most popular software tool in data science, with 46.9% share in 2015, compared to 38.5% in 2014. In our experience, exposing students to R increases their chances of getting quality internships and full-time jobs. In 2014, the R Markdown v2 package was released [9], which allows for seamless integration of computing with R and typesetting with LATEX, thus offering a modern technology for creating dynamic project reports, consistent with the principles of reproducible research [10,11].

In this paper, we provide all R code, required for the implementation of all projects, in full detail, so that one can easily translate it into other computational environments. There is evidence of pedagogical value in using multiple technologies in the mathematics classroom, as argued by Lassak [12]. There are some excellent books and online tutorials on R, and we refer the interested reader to some of them for a more comprehensive approach to learning the R language for scientific programming, visualization and simulation [13–19].

3. Applications to calculus

We present four computational projects from Calculus. The first project (Section 3.1) is about visualizing the graph of a nowhere differentiable function for building a visual intuition and gaining insight into non-differentiability. The second project (Section 3.2) explores visually the Weierstrass family of functions and the Weierstrass parametric curve that fails to have a tangent at any point on its graph. The third project (Section 3.3) applies Monte Carlo simulations to estimating volumes of hyper-balls and ellipsoids. The fourth project (Section 3.4) applies Monte Carlo simulations to estimating difficult integrals in just a couple of lines of code. Additional computer activities from Calculus, implemented in Maple, can be found in Benakli et al. [20] and Taraporevala et al. [21]. The Maple examples presented in these references can be adapted and implemented in R or other computing environments. At the more advanced level, in the context of discrete Fourier analysis, we have adapted the problems presented in Kostadinov [22] to develop computational and visualization projects for a course on Numerical Methods.

3.1. Looking at a nowhere differentiable function under the microscope

Advanced Calculus and Real Analysis students encounter examples of 'exotic functions', which are continuous everywhere but nowhere differentiable. In addition to having difficulties understanding and visualizing functions of this nature, students often wonder if functions that possess similar properties would have any real-world applications. Hollywood studios adopt such 'exotic', nowhere differentiable, functions to create special sound effects, exploiting the so-called Shepard's ever-ascending tones. In the movies, 'The Dark Knight' and 'The Dark Knight Rises', a Shepard's tone was adopted to create the sound of the Batpod. In music, The Beatles, Led Zeppelin, Pink Floyd, Queen, among others, have used Shepard's tones at the end of some famous songs.

It is a challenge to visualize the graph of a continuous function that is nowhere differentiable. We consider one specific example of this nature, the function $f(x)$, the mathematics of which is discussed in [23, p.154], in a somewhat terse form, but one can find a more detailed discussion of it in [24, p.508–510]. Students can construct $f(x)$ in several steps, through compositions of more elementary functions, and we use R to implement a good approximation to $f(x)$ and plot its graph.

We can start by implementing the triangular function $r(x)$, as the first building block of $f(x)$, defined by:

$$r(x) = \begin{cases} x, & \text{if } 0 \le x < 1 \\ 2 - x, & \text{if } 1 \le x \le 2 \\ 0, & \text{elsewhere} \end{cases} \tag{1}$$

We implement $r(x)$ as a piece-wise function and plot its graph in Figure 1.

```
r<-function(x) x*(x>=0 & x<1)+(2-x)*(x>=1 & x<=2) # piece-wise function
x<-seq(from=-1,to=3,by=0.01) # partition [-1,3] by step 0.01
plot(x, r(x),type="l",lwd=3,col="blue",main="Graph of r(x)")
```

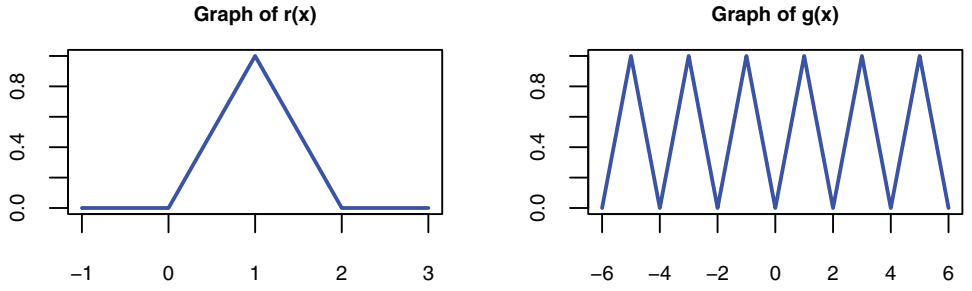

Figure 1. The graphs of the triangular function $r(x)$ and its periodic extension $g(x)$.

Note that `(x>=0 & x<1)` is a logical expression in R (`&` is the logical and), which results in a vector of logical values, when the argument x is a vector, and is being coerced into a binary vector of 1's and 0's, corresponding to `TRUE` and `FALSE`, depending on whether the conditions are satisfied or not. Finally, the expression `x*(x>=0 & x<1)` returns a vector of 0's and only those values of the vector x for which the conditions are satisfied. The sum of the two terms in the R function `r` implements $r(x)$, as defined by (1).

The function $r(x)$ is continuous everywhere but it is not differentiable at $x = 0, 1, 2$ (the cusps at these points do not allow for having a well-defined tangent line there). By means of translates of this function, it is possible to define everywhere continuous functions that fail to be differentiable at each point of an arbitrarily given finite set. Thus, we extend $r(x)$ into a periodic sawtooth function $g(x)$.

$$g(x) = r\left(x - 2\text{Floor}\left(\frac{x}{2}\right)\right), \tag{2}$$

where we define $\text{Floor}(x)$ as the largest integer not greater than x. We can implement $g(x)$ in R, as defined by (2), using the already defined function `r` and the built-in function `floor()`, and we can then plot the saw-tooth function $g(x)$.

```
x<-seq(-6,6,by=0.01) # a partition of [-6,6] by step 0.01
g<-function(x)r(x-2*floor(x/2)) # sawtooth function
plot(x, g(x),type="l",lwd=3,col="blue",main="Graph of g(x)")
```

Figure 1 shows the graphs of both $r(x)$ and $g(x)$. Finally, we define the sequence of functions $f_n(x)$, and then $f(x)$ as the limit of this sequence:

$$f(x) = \lim_{n\to\infty} f_n(x) \quad \text{where} \quad f_n(x) = \sum_{k=0}^{n} \left(\frac{3}{4}\right)^k g(4^k x) \tag{3}$$

To get a better understanding of the behaviour of $f_n(x)$, we plot the graph of $f_{20}(x)$ in Figure 2, which we use as an approximation to the graph of $f(x)$. First, we implement $f_{20}(x)$ as the R function `f20`, according to (3).

```
f20<-function(x){
  k<-0:20 # a vector of powers
  return(sum((3/4)^k*g(4^k*x)))} # returns f_{20}(x)
f20<-Vectorize(f20) # vectorized function needed for plotting
```

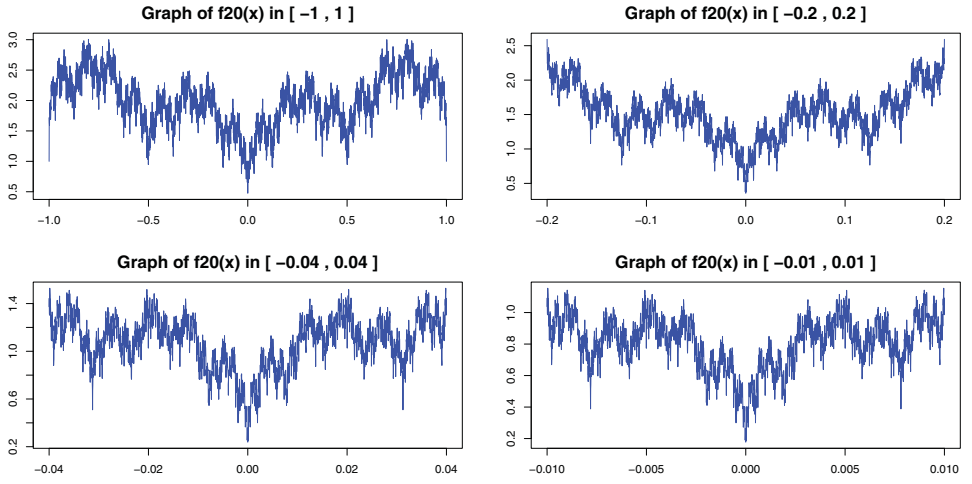

Figure 2. Under the microscope: zooming into the graph of $f_{20}(x)$, approximating $f(x)$.

We show how to create the matrix plot in Figure 2 with four plotting windows. Using a `for()` loop allows us to populate the 2×2 matrix plot with four graphs of $f_{20}(x)$ over different intervals. The last graph is magnified by a factor of 100. We zoom into the graph of $f_{20}(x)$ around 0 in four steps, starting from the interval $[-1, 1]$, and ending with the interval $[-0.01, 0.01]$:

```
zoom<-c(1,0.2,0.04,0.01) # c() combines numbers into a vector
par(mfrow=c(2,2)) # 4-plot window
for(n in 1:4){ # for loop to create the 4 plots
  x<-seq(-zoom[n],zoom[n],length.out=1e4)
  plot(x,f20(x),type="l",lwd=0.5,col="blue",
       main=paste("Graph of f20(x) in [",-zoom[n],",",zoom[n],"]"))
}
```

Clearly, the functions $r(x)$ and $g(x)$ are continuous everywhere. Since $f(x) = \lim_{n \to \infty} f_n(x)$, it follows that the function $f(x)$ is continuous since $f_n(x)$ is continuous, as a finite linear combination involving compositions of continuous functions. Figure 2 shows the rugged pattern of the graph of $f_{20}(x)$, which appears to exhibit self-similarity, a fractal-like structure, and the roughness in the graph keeps repeating, even if we keep zooming in (the last graph is magnified by a factor of 100 relative to the first graph), as seen in the four plots. It is very important for students to understand that Figure 2 presents a rather appealing visual insight, but not a rigorous proof that the function $f(x)$ is nowhere differentiable, since its graph does not smooth out as we observe it under the microscope.

To illustrate the difference with a differentiable function let us create the plot of $h(x) = \sin(x - 1)^2$, and observe it under the microscope around the point $x = 0$. It is evident from the plots in Figure 3 that the graph of $h(x)$ smooths out around $(0, h(0))$, as we keep magnifying it. In the last plot, after 100 times of magnification, we have zoomed into an interval around 0, where the graph of $h(x)$ has almost become identical with the tangent line to the graph of $h(x)$ at the point $(0, h(0))$.

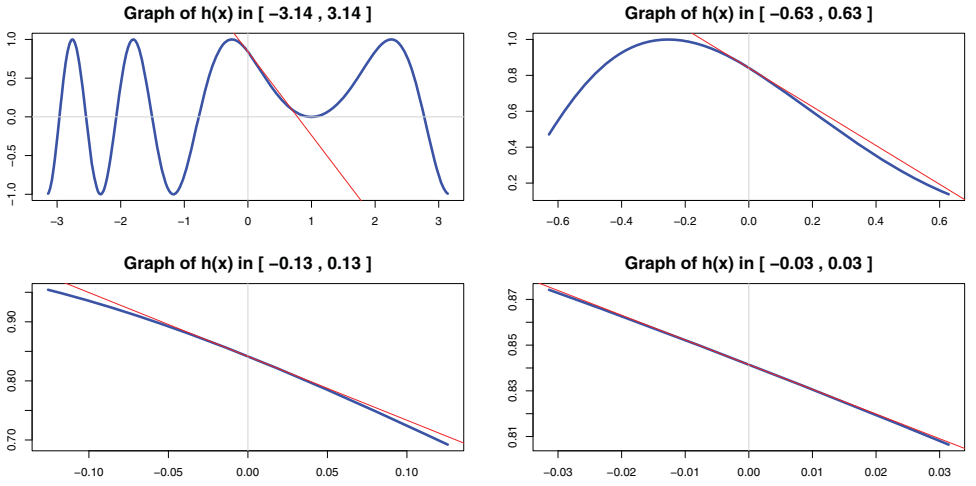

Figure 3. Under the microscope: zooming into the differentiable function $h(x)$.

The equation of the tangent line to $h(x)$ at the point $(0, h(0))$ is given by:

$$y = h(0) + h'(0)x = 0.84 - 1.08x \tag{4}$$

3.1.1. Mathematical details

We can justify our claims about nondifferentiability in a mathematically rigorous way. We now present a sequence of steps that students can carry out for Dirichlet's function $f(x) = \sum_{k=0}^{\infty} \left(\frac{3}{4}\right)^k g(4^k x)$ to establish our claims. This should give the students an appreciation for both rigor and experimentation through simulation in learning mathematics.

(1) Why is $f(x)$ continuous on $\mathbb{R}$? Observe that $g(x)$ is bounded and as such the series $\sum_{k=0}^{\infty} \left(\frac{3}{4}\right)^k g(4^k x)$ is uniformly convergent to $f(x)$ on $\mathbb{R}$.
(2) Choose an arbitrary real number x and an arbitrary positive integer n. Explain why if there is an integer in the open interval $(4^n x, 4^n x + 0.5)$, then there are no integers in $(4^n x - 0.5, 4^n x)$.
(3) Conclude from above that we can always produce $\delta_n = \pm 0.5 \cdot 4^{-n}$ where there are no integers in the open interval $(4^n x, 4^n (x + \delta_n))$.
(4) Show that $\left|\frac{g(4^m(x+\delta_n)) - g(4^m x)}{\delta_n}\right| = 4^m$ for $0 \le m \le n$, and 0 otherwise.
(5) For a fixed n, $\frac{g(4^m(x+\delta_n)) - g(4^m x)}{\delta_n}$ have the same sign for $m = 0, 1, 2, \ldots, n$. Show that $\left|\frac{f(x+\delta_n) - f(x)}{\delta_n}\right| = \left|\sum_{m=0}^{n} \left(\frac{3}{4}\right)^m \frac{g(4^m(x+\delta_n)) - g(4^m x)}{\delta_n}\right| = \sum_{m=0}^{n} 3^m = \frac{3^{n+1}-1}{2}$.
(6) Show that $\lim_{n \to \infty} \delta_n = 0$, and use the previous step to conclude that $\lim_{h \to 0} \frac{f(x+h) - f(x)}{h} = \pm\infty$.
(7) Conclude from above that $f(x)$ is nowhere differentiable but everywhere continuous.

3.2. Weierstrass functions and musical paradoxes

We present here the Weierstrass family as another example of nowhere differentiable functions. In fact, this is the family of functions used by Hollywood Studios to create Shepard's

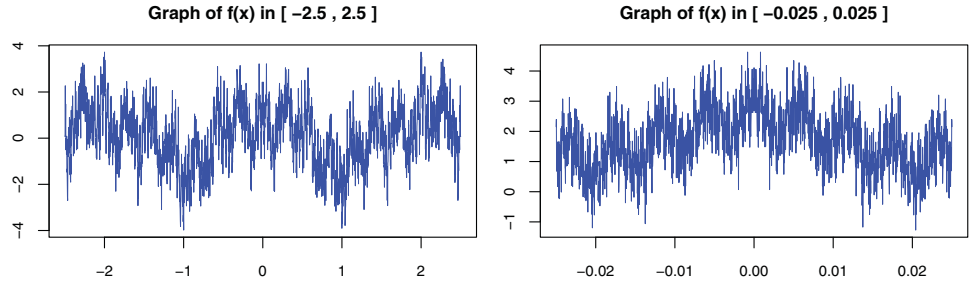

Figure 4. Under the microscope: zooming into the nowhere differentiable Weierstrass function.

tones and other musical 'paradoxes' [see 25, p.96–97]. The implementation details are left as an exercise as they are similar to the previous section. We refer the reader to [26, p.38–39] for details on the non-differentiability of the Weierstrass family:

$$W(x) = \sum_{n=0}^{\infty} a^n \cos(b^n \pi x), \tag{5}$$

where b is an odd integer and a is such that $0 < a < 1$ and $ab > 1 + \frac{3}{2}\pi$. Note that there is a typo in [26], interchanging the roles of a and b, but it is clear that the sum would be divergent in that case. For $b = 7$ and $a = 0.85$, we implement the function $f(x) = \sum_{n=0}^{30} a^n \cos(b^n \pi x)$ as an approximation to $W(x)$. In Figure 4, we plot the graph of $f(x)$ and we see strong visual evidence, suggesting that the limiting function $W(x)$ would be nowhere differentiable, since its graph does not smooth out as we zoom in.

Another visualization example, given as an exercise for the students, is the Weierstrass curve, related to (5), and defined parametrically by:

$$x = \sin(\theta), \quad y = \sum_{n=1}^{\infty} \frac{1}{2^n} \cos(3^n \theta) \tag{6}$$

In Figure 5, we plot the Weierstrass curve using 20 terms from the sum defining y. This is an example of a continuous closed curve such that *at none of its points can one draw a tangent to it*, because $dy/d\theta$ fails to exist for all θ.

3.3. Monte Carlo simulations for estimating volumes of unit n-balls

3.3.1. A hands-on, pen and paper, Monte Carlo experiment

Before we use a computer to implement the Monte Carlo method for estimating volumes of n-Balls, we try to help our students develop a good intuition and understanding of the ideas behind the technique by playing a real, hands-on, pen and paper game. Playing a real game of 'darts' that imitates the Monte Carlo method makes the computer simulations very real to students. Thus, before coding our first Monte Carlo computer experiments, we invite our students to run simple Monte Carlo experiments by hand, in order to get confident about correctly applying the procedure. In this simple game, we estimate the area shown in Figure 6. Of course, this problem can easily be solved exactly, but we use it as a playground

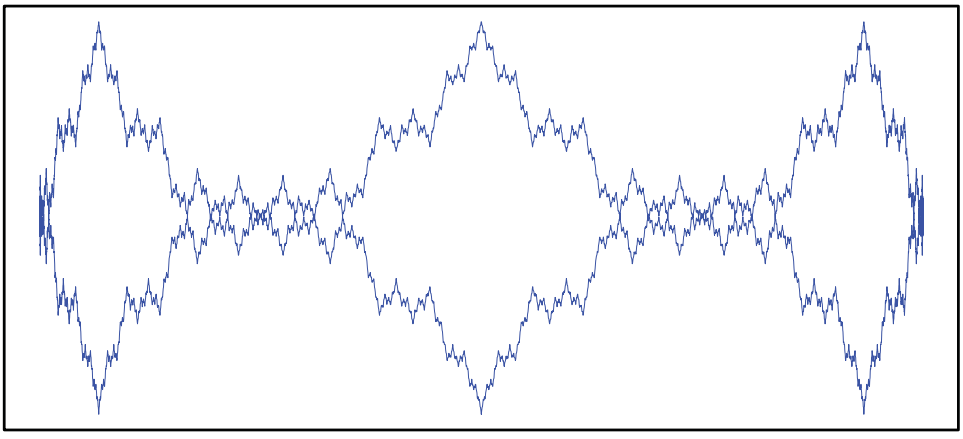

Figure 5. The Weierstrass curve.

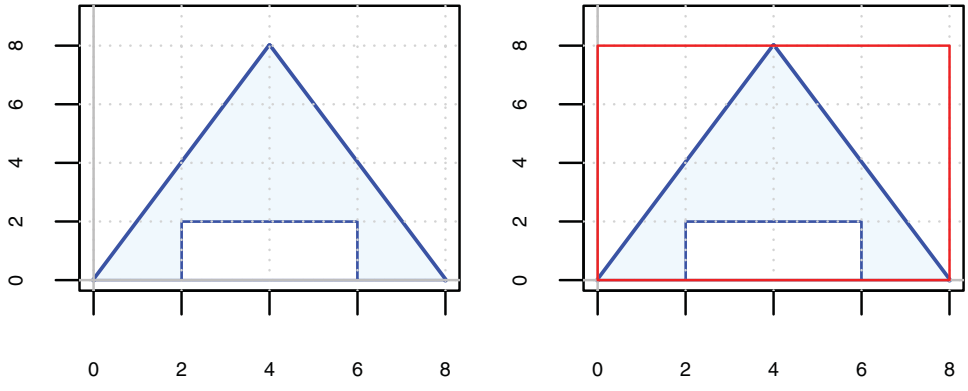

Figure 6. The first plot shows the area to be estimated, and the second plot adds a target frame.

to explain the ideas behind the Monte Carlo method. The true value of the Monte Carlo method lies in solving complex problems where no exact answers can be derived. What is the exact area of the shaded region?

Let us play a game of 'darts' with pen and paper:

(1) Throw 100 random 'darts' by dropping your pen or pencil at a random point inside the square target, in a way that the pen drops appear to be uniformly distributed over the square target.
(2) Compute the fraction of 'darts' (pen drops) that fall inside the shaded area.
(3) Multiply this fraction by the area of the square target region.

The number resulting in step (3) is an estimate for the area of the shaded region. Here are some questions that we ask the students:

- Do you expect that you should all get the same area estimate?
- What did you actually get? Compile a list of estimates from all students.

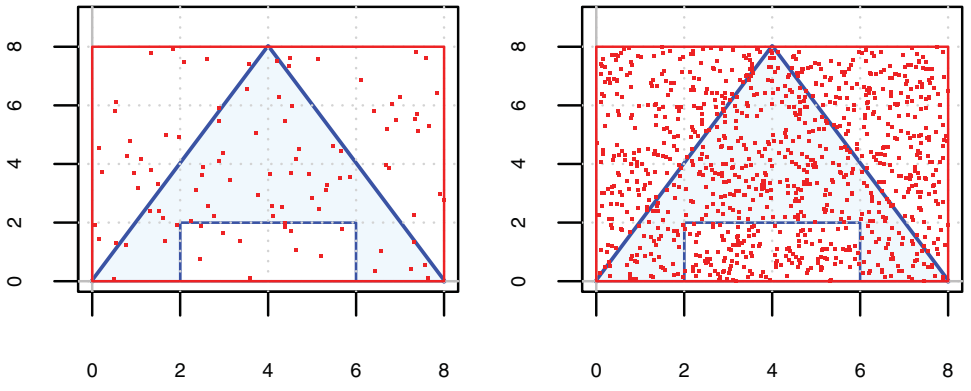

Figure 7. The first plot shows 100 random darts, and the second plot shows 1000 random darts.

- What is the average of all estimates?
- Draw the frequency histogram of the data with all area estimates.
- What does that histogram represent?

We define a *hit* to be a random point from the pen or pencil that lands inside the shaded area. We counted 37 hits out of 100 random drops of the pen or pencil within the target area. The fraction or relative frequency of hits is thus $f = 37\%$. The estimated area of the shaded region is 23.68, based on only 100 random drops of the pen. Note that the area of the triangle is $S_\Delta = 32$ and the area of the rectangle is $S_\square = 8$, thus the exact area of the shaded region is $S_{\Delta - \square} = 24$. In Figure 7, we show 100 and 1000 random drops of a pen inside the target area. Of course, the more random 'darts' we use, the more accurate area estimates we get. See Section 3.4.3 for more details on the error of the Monte Carlo estimates as a function of the number of random experiments.

3.3.2. Estimating the volume of the unit 3-ball

The volume of the three-dimensional ball of radius r, $B_3(r) \subset \mathbb{R}^3$, is the triple integral:

$$V = \iiint_{B_3(r)} 1 \, dx \, dy \, dz, \tag{7}$$

where the 3-ball is given by all points $(x, y, z) \in \mathbb{R}^3$ satisfying $x^2 + y^2 + z^2 \leq r^2$. Students can estimate the volume of the unit ball in $\mathbb{R}^3$ by randomly 'generating points' inside the cube of side 2, centred at (0, 0, 0), and counting how many fall inside the unit ball.

```
library(scatterplot3d) # must be installed before loaded
N<-1e4 # number of random points
# x,y,z vectors of random numbers sampled from U(-1,1)
x<-runif(N,-1,1); y<-runif(N,-1,1); z<-runif(N,-1,1)
scatterplot3d(x,y,z, pch="+", highlight.3d=TRUE,tick.marks=F,
              main="Random points inside the cube")
```

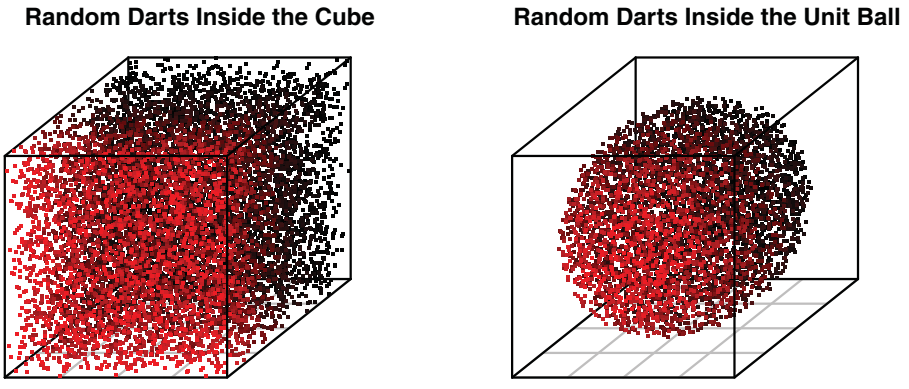

Figure 8. Generating random points inside the cube and visualizing those inside the unit ball.

We generate the vector of logical values `(x^2+y^2+z^2<=1)` that we use to index the random points falling inside the unit ball.

```
ind<-(x^2+y^2+z^2<=1) # vector of logical values
scatterplot3d(x[ind],y[ind],z[ind], pch="+", highlight.3d=TRUE,
              tick.marks=F,main="Random points inside the unit ball")
```

The two `scatterplot3d()` calls generate the two 3D plots shown in Figure 8 that visualize the game of points. We need to compute the fraction of points that fall inside the unit ball. This fraction can be computed as the mean of the logical vector `(x^2+y^2+z^2<=1)`. The product of this fraction and the volume of the cube of side 2 ($V = 2^3$) gives an estimate for the volume of the unit 3-ball.

```
# throw one million points inside the cube of side 2, centered at 0
x<-runif(1e6,-1,1); y<-runif(1e6,-1,1); z<-runif(1e6,-1,1)
frac<-mean(x^2+y^2+z^2<=1) # fraction of points inside the unit ball
```

The fraction of points that fall inside the unit ball can be computed by taking the arithmetic mean of the vector of logical values `x^2+y^2+z^2<=1`, which gives 0.5239.

```
vol<-2^3*mean(x^2+y^2+z^2<=1) # approximate volume of the unit ball
```

The computed estimate vol ≈ 4.191 for the volume of the unit ball is very close to the exact value, given by the formula that Archimedes discovered nearly 2000 years ago:

$$V(B_3(1)) = \frac{4}{3}\pi \approx 4.189 \tag{8}$$

3.3.3. Estimating the volume of the unit 4-ball

The Monte Carlo simulation can be easily generalized to n dimensions. The ball $B_n(r)$, of radius r in $\mathbb{R}^n$, is the set of points $(x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$ such that:

$$x_1^2 + x_2^2 + \cdots + x_n^2 \leq r^2 \tag{9}$$

Let us estimate the volume of the unit ball $B_4(1)$ in $\mathbb{R}^4$. The simulation in this case is basically the same as the simulation in $\mathbb{R}^3$, for the unit ball $B_3(1)$, except for adding one extra coordinate w. The following R code implements the entire simulation.

```
# generate random samples of size 10^6 from U(-1,1) for x,y,z,w
x<-runif(1e6,-1,1); y<-runif(1e6,-1,1); z<-runif(1e6,-1,1)
w<-runif(1e6,-1,1) # extra 4th dimension
vol<-2^4*mean(x^2+y^2+z^2+w^2<=1) # approx. volume of the unit 4-ball
```

Based on one million simulations for each spatial coordinate, the estimate `vol` ≈ 4.922 for the volume of the unit 4-ball is very close to the exact value obtained from the general formula. The formula, given in (10), can be derived recursively for any dimension n, a nice Calculus exercise we encourage the students to do.

$$V(B_n(1)) = \frac{\pi^{\frac{n}{2}}}{\Gamma\left(\frac{n}{2}+1\right)}, \tag{10}$$

where $\Gamma(x)$ is the gamma function, which is an extension of the factorial function. See Section 3.3.5 for the key steps leading to the derivation of (10). For $n = 4$, the exact volume of the unit 4-ball is then $V(B_4(1)) \approx 4.935$, according to (10). Similarly, students can estimate with high accuracy the hypervolumes of higher dimensional balls.

3.3.4. Estimating volumes of n-balls and ellipsoids

A related problem that could serve as a student project would be to visualize the limit of hypervolumes as the dimension n gets larger. It turns out that the volume of the unit n-ball goes to 0 as the dimension n goes to ∞, and the maximum volume is attained in dimension 5. Monte Carlo simulations can be used to estimate volumes of other solids. For example, consider the ellipsoid in Figure 9, with semi-axis $a = 1$, $b = 2/3$, $c = 1/2$, given by:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1 \tag{11}$$

The simulation for estimating the volume of the ellipsoid, given by (11), is left as a hands-on exercise for the students. In addition, we encourage students to reproduce Figure 9.

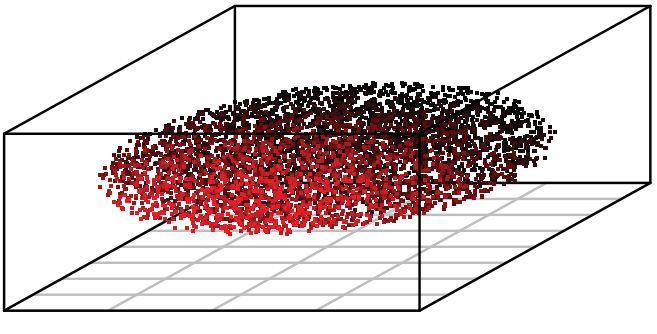

Figure 9. Random points inside the ellipsoid.

In particular, a Monte Carlo approach, based on 10^6 simulations, gives an estimate of $V \approx 1.397$ for the volume of the ellipsoid given by (11). Of course, in this case we do have a general formula for the volume $V = \frac{4}{3}\pi abc \approx 1.396$. However, it is important to understand that the same simulation approach can be used for much more complicated solids for which we may not have a formula for the volume.

3.3.5. Mathematical details

In the series of exercises below we will compute the volume $V(B_n(1))$ for $n = 3, 4$ and $n > 4$. We will now outline how to evaluate the integral $\iiint_{B_3(1)} dx\,dy\,dz$ by having students fill in the appropriate details in the following exercises.

(1) To find the volume of the unit sphere (3-Ball) we will consider the region $B_3(1)$ specified by the conditions $-1 \le z \le 1$, $-\sqrt{1-z^2} \le y \le \sqrt{1-z^2}$, and $-\sqrt{1-z^2-y^2} \le x \le \sqrt{1-z^2-y^2}$. Explain why
$V(B_3(1)) = \iiint_{B_3(1)} dx\,dy\,dz = \int_{-1}^{1} \int_{\sqrt{1-z^2}}^{\sqrt{1-z^2}} \int_{-\sqrt{1-z^2-y^2}}^{\sqrt{1-z^2-y^2}} dx\,dy\,dz.$

(2) Substitute $y = \sqrt{1-z^2}\sin\theta$ in the above integral and simplify to show that $V(B_3(1)) = \frac{4\pi}{3}$.

(3) To find the volume of the unit 4-Ball we will consider the region $B_4(1)$ specified by the conditions $-1 \le w \le 1$, $-\sqrt{1-w^2} \le z \le \sqrt{1-w^2}$, $-\sqrt{1-w^2-z^2} \le y \le \sqrt{1-w^2-z^2}$, and $-\sqrt{1-w^2-z^2-y^2} \le x \le \sqrt{1-w^2-z^2-y^2}\}$. Explain why
$V(B_4(1)) = \iiint_{B_4(1)} dv = \int_{-1}^{1} \int_{\sqrt{1-w^2}}^{\sqrt{1-w^2}} \int_{\sqrt{1-w^2-z^2}}^{\sqrt{1-w^2-z^2}} \int_{-\sqrt{1-w^2-z^2-y^2}}^{\sqrt{1-w^2-z^2-y^2}} dv$, where $dv = dx\,dy\,dz\,dw$.

(4) Substitute $y = \sqrt{1-w^2-z^2}\sin\theta_1$ and eventually $w = \sin\theta_2$ in the above integral and simplify to show that $V(B_4(1)) = \frac{\pi^2}{2}$.

(5) We will now proceed to find the volume of the unit n-Ball. Consider the unit n-ball in $\mathbb{R}^n$ described by $x_1^2 + x_2^2 + \cdots + x_n^2 \le 1$. Mimic the above steps for $n = 3$ and $n = 4$ for the n-Ball and use the substitutions $x_j = \sqrt{1-x_n^2-x_{n-1}^2-\cdots-x_{j+1}^2}\sin\theta_j$ and the formula $\int_{-\pi/2}^{\pi/2} \cos^n x\,dx = \frac{\sqrt{\pi}\,\Gamma\left(\frac{n+1}{2}\right)}{\Gamma(n/2+1)}$ to show that $V(B_n(1)) = \frac{\pi^{n/2}}{\Gamma(n/2+1)}$.

3.4. Monte Carlo integration

One important application of Monte Carlo simulations is the computation of difficult integrals. In the case of multi-dimensional integrals, using random sampling to estimate them has some advantages over deterministic numerical schemes, which perform poorly. Monte Carlo integration allows students to estimate difficult integrals in just a couple of lines of code, which leaves a lasting impression on them.

3.4.1. Estimating 1D integrals

Let us consider the integral in (12), which cannot be evaluated analytically.

$$J = \int_0^2 \sin(x)e^{-x^3}\,dx \tag{12}$$

We can represent the integral J, given by (12), in terms of the expected value of the integrand function applied to the random variable $U \sim U(0, 2)$.

$$J = 2\int_0^2 \frac{1}{2}\sin(x)e^{-x^3}dx = 2E\left[\sin(U)e^{-U^3}\right] \tag{13}$$

All we need to do is simulate a large sample `u` from $U(0, 2)$, using the random number generator `runif()` for the Uniform distribution, and take the `mean()` of the integrand function applied to this sample.

```
u<-runif(1e6,0,2) # a sample of size 10^6 from U(0,2)
J<-2*mean(sin(u)*exp(-u^3)) # an estimate for J
```

The Monte Carlo integration gives an estimate of $J \approx 0.40439$. On the other hand, **Wolfram Alpha** gives the following result, based on deterministic numerical schemes.

```
From Wolfram Alpha: integral_0^2 sin(x) exp(-x^3) dx = 0.40442
```

3.4.2. Estimating 3D integrals

In 1939 the English mathematician G.N. Watson showed in [27] how to evaluate three difficult integrals, suggested by the physicist W.F. van Peype, over a 3D cube with edge-length π, normalized to the volume of that cube. The Watson/van Peype integrals turn-up in the physics of frozen magnetic crystals, and in the pure mathematics of random walks. The integral I, defined in (14), was the most difficult one, which even G.H. Hardy, the famous slayer of integrals, could not solve.

$$I = \frac{1}{\pi^3}\int_0^\pi\int_0^\pi\int_0^\pi \frac{dudvdw}{3-\cos(u)-\cos(v)-\cos(w)} \tag{14}$$

The exact value of I, given in (15), was obtained by Watson after pages of clever analysis.

$$I = \frac{\Gamma(\frac{1}{24})\Gamma(\frac{5}{24})\Gamma(\frac{7}{24})\Gamma(\frac{11}{24})}{16\sqrt{6}\pi^3} \approx 0.5055, \tag{15}$$

where $\Gamma(x)$ is the Gamma function. See Nahin [28, p.206-212] for a mathematical discussion of the clever tricks Watson employed. We demonstrate how our students can estimate I using the Monte Carlo approach. The idea is basically the same as in the 1D case. First, we rewrite I as the expected value of the integrand function applied to the three independent random variables $U, V, W \sim U(0, \pi)$.

$$I = E[g(U, V, W)] = \int_0^\pi\int_0^\pi\int_0^\pi f_{U,V,W}(u, v, w)g(u, v, w)dudvdw, \tag{16}$$

where $g(u, v, w) = \frac{1}{3-\cos(u)-\cos(v)-\cos(w)}$ is the integrand function of I, and $f_{U,V,W}(u, v, w) = f_U(u)f_V(v)f_W(w) = 1/\pi^3$ is the joint PDF of U, V, W, which splits into the product of the three marginal PDFs, thanks to independence. The Monte Carlo

simulation is given by the following compact chunk of R code:

```
set.seed(12345) # for reproducible results
u<-runif(1e6,0,pi); v<-runif(1e6,0,pi); w<-runif(1e6,0,pi) # ind. samples
I<-mean(1/(3-cos(u)-cos(v)-cos(w))) # Monte Carlo estimate for I
```

which gives a Monte Carlo estimate of $I \approx 0.507$, very close to the exact value in (15).

Monte Carlo integration can be applied to any integral of any multiplicity and any difficulty, using a bag of Monte Carlo simulation techniques, the general *Hit-or-Miss* approach, chief among them.

3.4.3. The error in Monte Carlo simulations

The error δ of the Monte Carlo method, in estimating the expected value of a random variable, is related to the number of simulations N:

$$\delta \sim \frac{1}{\sqrt{N}} \tag{17}$$

In practical problems, the Monte Carlo method gives an error of the order of 1% to 0.1% of the true value, corresponding to $N = 10^4$ and $N = 10^6$, respectively. There are general variance reduction techniques that could be used for improving the Monte Carlo estimates, without increasing the sample size N. See Ross [29] for a mathematical discussion of variance reduction techniques and other simulation topics. More examples of Monte Carlo integration with R can be found in Kostadinov [13], while Boros and Moll [30] offer a good collection of challenging integrals.

4. Applications to probability

We present three computational projects from probability, inspired by [31,32]. The first project (Section 4.1) simulates a sample from the distribution of a random variable and computes a probability of interest based on the simulated sample. The second project (Section 4.2) simulates a sample from the distribution of a random sum of random variables, and computes key sample statistics and a probability of interest. The third project (Section 4.3) simulates a sample from the distribution of an insurance company payout, given as a call option on an underlying exponential distribution, and estimates the expected payout made by the insurance company as well as a probability of interest based on the simulated sample.

Additional problems related to Geometrical Probability, which could serve as good computational projects for students, can be found in the UMAP modules by Dahlke and Fakler [31,33]. We also recommend Dobrow's book [32], which emphasizes simulations through the use of R and illustrates important computational and theoretical results through many applications from fields as diverse as biology, computer science, cryptology, ecology, public health and sports. Other simulation projects, implemented in the spirit of this article, can be found in Kostadinov [13]. A more leisurely introduction to hands-on coding in R, with emphasis on simulations, can be found in Grolemund [16].

4.1. Citizen band radio simulation

Example 4.1: Two CB radio operators, Jose and Thomas work for Lorenzo's trucking business. The range of their CB radios is 45 km. Jose is travelling towards the base from the east and at 3:00 P.M. is somewhere within 50 km of the base. Thomas is travelling towards the base from the north and at 3:00 P.M. is somewhere within 50 km of the base. Calculate analytically (by-hand) the probability Jose and Thomas can communicate with each other using their radios at 3:00 P.M., and then estimate it using computer simulations by computing the relative frequency of the event they can communicate.

First, the students are encouraged to *think mathematically* about the problem and the analytical approach they need to develop by answering some key questions:

(1) Draw on paper a diagram that represents the geometry of the problem.
(2) Express mathematically the event of interest by introducing appropriate notation.
(3) Draw on paper the sample space and the event success region inside the sample space that visualizes the conditions for the event of interest to be realized.
(4) Make assumptions about the distributions of the distances of the two trucks from the base, and express mathematically the distance between the two trucks.
(5) Express mathematically the probability Jose and Thomas can communicate with each other, and think how to use the geometry of the sample space and the success region that defines the event to express this probability in terms of areas.

Second, the students are encouraged to *think computationally* about the problem, and how to implement in R a simple algorithm for estimating the desired probability using computer simulations, by addressing the following questions:

(1) Given your assumptions, generate large samples from the distributions of all random variables of interest, using the built-in random number generators in R.
(2) Visualize the geometry of the problem by creating plots and diagrams using R.
(3) Estimate the desired probability by computing the relative frequency of the event.
(4) Design your algorithm in the spirit of *vectorized functional computing*, that is, apply functions to the vectors of simulated samples as if doing mathematics.
(5) Write a project report in RStudio comparing the analytical and computational results, using RMarkdown. The idea is to *integrate the technological and by-hand techniques*, the former integrated through chunks of R code and the latter typeset using a simplified LATEX format supported by RStudio. For more details on how to unify computing in R and mathematical typesetting with RMarkdown, see [9,10].

We next present detailed analytical and computational solutions. On the analytical side, we assume that the distances, X and Y, of the two trucks from the base are uniformly and independently distributed within the given range, that is $X, Y \sim U(0, 50)$. The distance D between the two trucks is a random variable D, which is given in terms of X and Y by $D = \sqrt{X^2 + Y^2}$, based on the Pythagorean theorem. The probability that the two trucks can communicate is specified by the condition $D \leq 45$ that the distance between them is at most

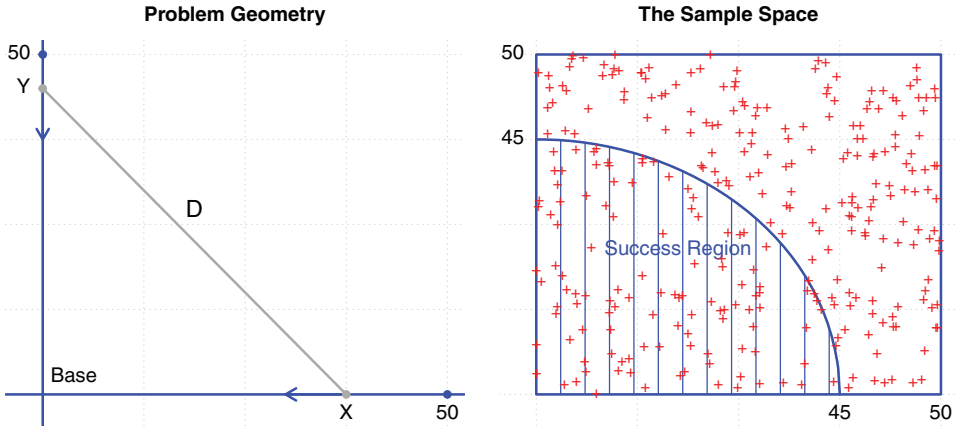

Figure 10. The geometry of the problem and its sample space.

45 km, which defines the plane region $\mathcal{D} : X^2 + Y^2 \leq 45^2, X \geq 0, Y \geq 0$, representing the quarter disk of radius 45, in the first quadrant. The sample space is given by the square of size 50 km, located in the first quadrant, whose lower left corner is centred at the origin. Figure 10 illustrates the geometry of the problem and its sample space. Since X and Y are independent, the probability is given by:

$$p = \mathbb{P}(D \leq 45) = \int\int_{\mathcal{D}} \frac{1}{50^2} dxdy = \frac{\text{Area}(\mathcal{D})}{50^2}, \tag{18}$$

where $\text{Area}(\mathcal{D}) = \frac{1}{4}\pi 45^2$, thus $p = \frac{\pi}{4}\left(\frac{45}{50}\right)^2 = 0.6362$ for the desired probability.

On the computational side, students need to generate samples `x` and `y`, from the distributions of the random variables X and Y, and use them to generate a sample from the distribution of the distance D between the two trucks.

```
set.seed(123) # for reproducible results
x<-runif(1e6,0,50) # a sample of size one million from U(0,50)
y<-runif(1e6,0,50) # another independent sample from U(0,50)
d<-sqrt(x^2+y^2) # sample of size 10^6 for the distance between trucks
```

The command `summary(d)` displays key sample statistics of `d`.

```
summary(d) # summary statistics of d

##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##  0.05964 28.20000 39.88000 38.26000 48.88000 70.66000
```

All arithmetic and logical operators in R act on vectors element-wise, which allows for vectorizing the code and simulating the distribution of D with just few lines of code. The probability the two trucks can communicate is specified by the event $D \leq 45$, and it is estimated in (19) by computing the relative frequency `mean(d<=45)` of this event.

$$\mathbb{P}(D \leq 45) \approx \texttt{mean(d<=45)} = 0.6361. \tag{19}$$

This is an example of a vectorized functional computing, where all operations are done with vectors and functions applied to them. In the code, `x` and `y` are vectors of size one million and so is `d<-sqrt(x^2+y^2)` (computed entry-wise). The comparison operator in `(d<=45)` creates a vector of logical values, `TRUE`, if the inequality is satisfied, and `FALSE` otherwise. Taking the mean of a logical vector coerces `TRUE` to 1 and `FALSE` to 0. In (20), we show how the arithmetic mean of a vector of logical values is equivalent to computing the relative frequency that the condition `(d<=45)` is satisfied.

$$\texttt{mean(d<=45)} = \frac{1}{N}\sum 1's = \frac{\#1's}{N}, \tag{20}$$

where $N =$ `length(d)`. Therefore, the code `mean(d<=45)` can be used to estimate the probability of the event `(d<=45)`. This is a key simulation insight for estimating probabilities by computing relative frequencies, based on vectorized functional programming.

In the right-hand plot in Figure 10, we show a different way of thinking about the probability $\mathbb{P}(D \leq 45)$, based on the so called *hit-or-miss* simulation approach. The idea is inspired by the game of darts discussed in Section 3.3.1. In fact, we 'dropped' some 300 random 'darts' inside the sample space (shown as +), and if we compute the fraction of 'darts' that have fallen inside the 'success region', that is, the quarter disk of radius 45, we get an estimate of the desired probability. In this particular experiment, we have 185 'darts' inside the quarter disk, so the success fraction is 0.62, which is a rough estimate of the probability $\mathbb{P}(D \leq 45)$ when we play with 300 'darts' only. In our experience, students find thinking about probabilities in terms of relative frequencies much more intuitive in the context of the *hit-or-miss* picture, inspired by the game of 'darts'.

4.2. A restaurant revenue problem

Example 4.2: The number of customers N who come in every day to Elena's Restaurant follow a Poisson distribution with an average of 100 customers each day. Each customer spending follows a Normal distribution with an average of \$22 and standard deviation of \$4. Customers' spending is independent of each other and of N. Simulate the distribution of the customers' total spending, and estimate the expected total spending and its standard deviation. Compute the probability the total spending is at least \$2000.

Let $B_1, B_2, \ldots, B_N$ be the bills of the N customers. Each bill B_k follows the Normal distribution, $B_k \sim N(\mu_B = 22, \sigma_B = 4)$. The total number of customers N follows the Poisson distribution $N \sim \text{Pois}(\lambda = 100)$, which is completely determined by the average number of 100 customers per day. The total spending at the restaurant is given by:

$$S = \sum_{k=1}^{N} B_k, \quad N \sim \text{Pois}(\lambda = 100), \quad B_k \sim N(\mu_B = 22, \sigma_B = 4) \tag{21}$$

The goal is to simulate the distribution of total spending S, which is a random sum of random variables given by (21). The implementation idea is to design a random experiment that simulates a single realization of the random variable S, and then students must replicate this

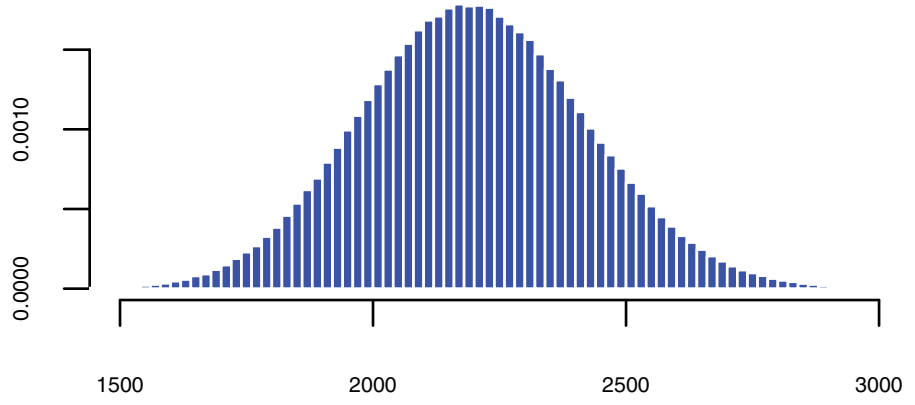

Figure 11. Density histogram of total spending.

random experiment many times to generate a sample from S.

```
spending<-function(){ #function simulating the random experiment once
  N<-rpois(1,100) # a single Poisson sample
  bills<-rnorm(N,22,4) # N Normal samples
  return(sum(bills)) # a single sample of total spending
}
```

A single realization of the random variable S can be generated by calling the function `spending()`, which implements the random experiment of interest. To get the distribution of total spending, we replicate the random experiment 10^6 times, using the R function `replicate()`, which takes as arguments the number of times we want to replicate the random experiment, and the R function that simulates the random experiment once. This way, we can generate a sample of size one million from the distribution of total spending. We can get the summary of sample statistics:

```
spending.sample<-replicate(1e6,spending())
summary(spending.sample) # sample statistics

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1158    2047    2196    2200    2349    3377
```

The sample standard deviation of S is computed by `sd(spending.sample)` $=$ 223.44. In Figure 11, we visualize the distribution of S by creating a density histogram of the simulated sample.

```
hist(spending.sample,breaks=100,freq=F,col="blue",border="white",
     xlab="total spending",main="Density histogram of total spending")
```

Having a large sample from the distribution of total spending allows students to estimate any statistics of interest. In particular, the probability that the total spending is at least \$2000 is computed in (22).

$$\mathbb{P}(S \geq 2000) \approx \texttt{mean(spending.sample >= 2000)} = 0.81 \tag{22}$$

4.2.1. Mathematical details

It is instructive to derive a formula for the expected value of total spending $S = \sum_{k=1}^{N} B_k$, using the *law of total expectation*, which allows us to compute the expected value of S by conditioning on another random variable X: $\mathbb{E}[S] = \mathbb{E}[\mathbb{E}[S|X]]$. Keep in mind that the conditional expectation $\mathbb{E}[S|X]$ is a random variable itself. For a proof of the law of total expectation, see Ross [29, p.31–32]. This is a very useful technique that students usually find difficult to follow. The key steps in deriving $\mathbb{E}[S]$ are the following:

(1) Using the law of total expectation with $X = N$, we get: $\mathbb{E}[S] = \mathbb{E}[\mathbb{E}[\sum_{k=1}^{N} B_k|N]]$.
(2) Since conditional expectation is linear, we can take it inside the sum in the right-hand side, when we consider a fixed N, thanks to the conditioning. In particular, we have $\mathbb{E}[S|N = n] = \mathbb{E}[\sum_{k=1}^{N} B_k|N = n] = \sum_{k=1}^{n} \mathbb{E}[B_k|N = n]$.
(3) Since the B_ks are independent of N for all k: $\mathbb{E}[B_k|N = n] = \mathbb{E}[B_k]$.
(4) Since $\mathbb{E}[B_k] = \mathbb{E}[B_1]$ for all k, $\mathbb{E}[S|N = n] = n\mathbb{E}[B_1]$, thus $\mathbb{E}[S|N] = N\mathbb{E}[B_1]$.
(5) Finally, since $\mathbb{E}[S] = \mathbb{E}[\mathbb{E}[S|N]] = \mathbb{E}[N\mathbb{E}[B_1]]$, and since $\mathbb{E}[B_1]$ is a number, by linearity: $\mathbb{E}[S] = \mathbb{E}[B_1]\mathbb{E}[N] = 22 \times 100 = 2200$.

In a similar way, one can use the *law of total variance* or the *conditional variance formula*, as referred to in Ross [29, p.32–33], to find the variance of total spending S by conditioning on N:

$$\mathrm{Var}(S) = \mathbb{E}[\mathrm{Var}(S|N)] + \mathrm{Var}(\mathbb{E}[S|N]) = \sigma_B^2\mu_N + \mu_B^2\sigma_N^2 = 5 \times 10^4 \tag{23}$$

where we used that $\mathrm{Var}(N) = \mathbb{E}[N] = \lambda$ since N has a Poisson distribution. Keep in mind that the conditional variance $\mathrm{Var}(S|N)$ is a random variable itself. We leave to the reader the details of applying the law of total variance in order to derive the right-hand side of Equation (23). Thus, the standard deviation of S is $sd(S) = \sqrt{\mathrm{Var}(S)} = 223.61$. Note that while it is relatively easy to derive formulas for the expected total spending and its variance, it is much more difficult to derive a formula for the probability $\mathbb{P}(S \geq 2000)$, yet estimating it by Monte Carlo simulation is as easy as computing the sample mean and variance, given a simulated large sample of S.

4.3. A medical insurance problem

Example 4.3: Elena's insurance will pay for a medical expense subject to a \$100 deductible. Let the amount of the expense be exponentially distributed with rate $\lambda = 0.001$.

- Simulate a large sample from the distribution of the insurance payout.
- Create a density histogram of the insurance payout.
- Derive (on paper) the theoretical PDF of the insurance payout and overlay it on top of the density histogram to compare the two.
- Estimate the probability of zero insurance payout using the simulated sample.
- Derive (on paper) the exact probability of zero insurance payout.
- Estimate the expected insurance payout using the simulated sample.
- Derive (on paper) the exact expected value of the insurance payout.

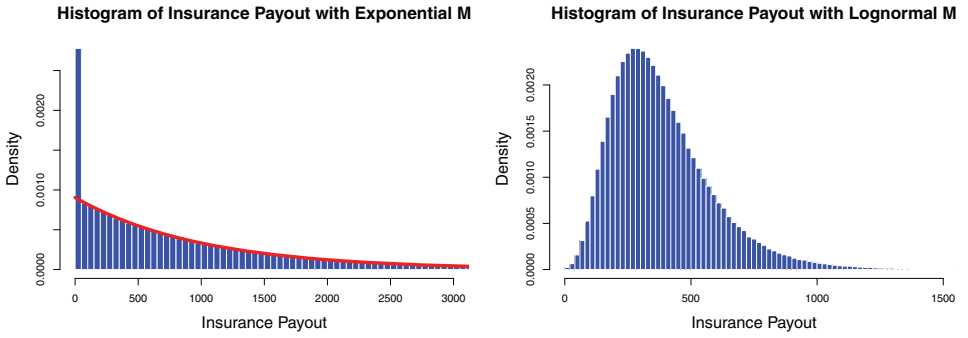

Figure 12. Distribution of insurance payout with exponentially and log normally distributed medical expense.

Let M be the amount of the medical expense and let X be the insurance payout.

$$X = \begin{cases} M - 100, & \text{if } M > 100 \\ 0, & \text{if } M \leq 100, \end{cases} \tag{24}$$

where $M \sim \text{Exp}(\lambda = 0.001)$ and the PDF of M is $f_M(x) = \lambda e^{-\lambda x}$ for $x \geq 0$, and $f_M(x) = 0$ for $x < 0$. *In this problem, students encounter something they usually have not seen before, namely that the random variable X has both discrete and continuous components since $X = 0$ if and only if $M \leq 0$ and thus the event $(X = 0)$ has a non-zero probability*. The goal is to simulate a sample from the distribution of X, and estimate the expected insurance payout $\mathbb{E}(X)$, among other things. Note that the insurance payout X can be written in the following equivalent mathematical form:

$$X = \max(M - 100, 0) = (M - 100)^+, \tag{25}$$

which can be interpreted as a **call option** payoff (of European type), with a strike price equal to the deductible – a widespread vanilla style financial derivative, ubiquitous in insurance and finance.

We can simulate the distribution of X by generating a large sample from the exponential distribution of the medical expense M, for the given parameter $\lambda = 0.001$. To do so, we use the random number generator `rexp()`, and apply the `pmax()` function, which implements the `max()` function defining X in (25), applied in a vectorized fashion (point-wise). Once, we have a large sample from X, we can easily create a density or frequency histogram of its distribution by simply applying the `hist()` function to the sample. We can add the theoretical PDF to the plot of the density histogram of X. This is shown in the left-hand plot in Figure 12, where we observe a very close match between the theoretical PDF and the density histogram of the simulated sample.

```
lambda<-0.001 # the Exp rate
M<-rexp(1e6,lambda) # sample of size 10^6 from Exp(lambda)
X<-pmax(M-100,0) # sample of size 10^6 from X
hist(X,500,freq=F,col="blue",border="white",xlim=c(0,5000),
     main="Distribution of Insurance Payout",cex.main=0.9)
```

We can derive the theoretical PDF $f_X(x)$ of X by first obtaining the CDF function $F_X(x) = \mathbb{P}(X \leq x)$ and then taking the derivative with respect to x: $f_X(x) = \frac{d}{dx}F_X(x)$. For $x \geq 0$, we can compute the probability that defines the CDF function by conditioning on the two mutually exclusive events $M > 100$ and $M \leq 100$:

$$F_X(x) = \mathbb{P}(X \leq x|M > 100)\mathbb{P}(M > 100) + \mathbb{P}(X \leq x|M \leq 100)\mathbb{P}(M \leq 100) \tag{26}$$

Equation (26) is often called the *law of total probability* and it gives the desired probability as the weighted average of conditional probabilities, where the weights are the probabilities of the mutually exclusive events. Note that $\mathbb{P}(X \leq x|M \leq 100) = 1$ since the event $X \leq x$ is certain as in this case $X = 0$, given that $M \leq 100$, and we are considering the case $x \geq 0$. On the other hand, $\mathbb{P}(M \leq 100) = \int_0^{100} \lambda e^{-\lambda x} dx = 1 - e^{-100\lambda}$. Thus, we have:

$$F_X(x) = \mathbb{P}(X \leq x|M > 100)\mathbb{P}(M > 100) + 1 - e^{-100\lambda} \tag{27}$$

Since $X = M - 100$, given that $M > 100$, the definition of conditional probability gives:

$$\mathbb{P}(X \leq x|M > 100) = \mathbb{P}(M \leq 100 + x|M > 100) = \frac{1}{\mathbb{P}(M > 100)} \int_{100}^{100+x} \lambda e^{-\lambda m} dm \tag{28}$$

From (27) and (28), we obtain the CDF of X:

$$F_X(x) = \begin{cases} 0, & \text{if } x < 0 \\ 1 - e^{-\lambda(100+x)}, & \text{if } x \geq 0 \end{cases} \tag{29}$$

One can derive the same result by conditioning on M as a continuous random variable, and using the continuous version of the law of total probability. We leave this approach as an exercise for the interested reader. Note that the CDF of X is not continuous but rather it has a jump discontinuity at $x = 0$ of size $1 - e^{-100\lambda}$, consistent with having a discrete component at $X = 0$. The PDF is obtained after differentiating the CDF:

$$f_X(x) = \frac{d}{dx}F_X(x) = \begin{cases} 0, & \text{if } x < 0 \\ \lambda e^{-\lambda(100+x)}, & \text{if } x \geq 0 \end{cases} \tag{30}$$

We can implement the PDF as a piece-wise function and add its graph to the histogram plot by using the `curve()` command with the optional parameter `add=TRUE`.

```
f<-function(x) lambda*exp(-lambda*(100+x))*(x>=0) # the exact pdf
curve(f,0,5000,col="red",lwd=4,add=TRUE) # add exact pdf to histogram
```

The sharp peak of the density at zero in Figure 12 represents the probability for the discrete event $X = 0$ of the random variable X. We can easily estimate this probability by computing the relative frequency of the event $X = 0$ for the simulated sample of X:

$$\mathbb{P}(X = 0) \approx \texttt{mean(X=0)} = 0.1 \tag{31}$$

Of course, if the distribution of X was purely continuous, then this probability would have been zero. We can derive the exact formula for the probability of $X = 0$ by observing

that $X = 0$ if and only if $M \leq 100$:

$$\mathbb{P}(X = 0) = \mathbb{P}(M \leq 100) = \int_0^{100} \lambda e^{-\lambda x} dx = 1 - e^{-100\lambda} = 0.1 \tag{32}$$

The simulated sample from the distribution of X allows us to immediately estimate any desired statistics associated with X. For example, the sample mean of X is simply `mean(X)`, and it gives the estimate $\mathbb{E}[X] \approx$ `mean(X)` $= 904.63$. It is instructive to compute the exact value of $\mathbb{E}[X]$ by conditioning on the continuous random variable M, using *the continuous version of the law of total probability*:

$$\mathbb{E}[X] = \mathbb{E}[\mathbb{E}[X|M]] = \int_0^{\infty} \mathbb{E}[X|M = m]\lambda e^{-\lambda m} dm \tag{33}$$

$$= \int_{100}^{\infty} \mathbb{E}[M - 100|M = m]\lambda e^{-\lambda m} dm \tag{34}$$

$$= \int_{100}^{\infty} (m - 100)\lambda e^{-\lambda m} dm \tag{35}$$

$$= \frac{e^{-100\lambda}}{\lambda} = 904.84 \tag{36}$$

where the third equality follows because $X = (M - 100)^+ = M - 100$ when $M > 100$, and $X = 0$ when $M \leq 100$. Alternatively, $\mathbb{E}[X]$ can be calculated by conditioning on the two mutually exclusive events $M \leq 100$ and $M > 100$, using the discrete version of the law of total probability and the definition of conditional expectation. We leave the details of this approach to the interested reader. We can also ask students to repeat this simulation in the case when the medical expense has a log-normal distribution of the form $M = 100e^X$, where $X \sim \mathrm{N}(\mu = 1.5, \sigma = 0.4)$. The density histogram of the insurance payout in this case is shown in the right-hand plot in Figure 12. Note that in this case, the discrete component in the payout distribution becomes negligible. It is worth mentioning that the log-normal distribution is the classical probability model for the price distributions of tradable assets in the financial markets and it is the basis for the derivation of the famous *Black-Scholes-Merton formula* for pricing European call and put options written on stocks and other financial assets.

5. Applications to data analysis

We present two computational projects based on performing data analysis, using real-world data. The first project (Section 5.1) illustrates exploratory data analysis and visualization of Boston housing data from the 1980s, including regression analysis of fitting linear models to the data. The second project (Section 5.2) covers analysis and visualization of categorical breast cancer data, including 2-way contingency tables, and illustrates Bayes' Theorem in this context. We refer the interested reader to [15–19] for a more comprehensive introduction to data analysis and visualization using R.

5.1. Analysis of Boston housing data

In this example, we give a taste of what R offers for exploratory data analysis. Often, our students use data-sets that are already available in base R. In particular, we use the data-set Boston from the MASS library, which comes with base R. This data-set has 14 variables (as columns) and 506 observations (as rows). The dimensions of `data` can be extracted with the `dim(data)` command. We remove three columns from the original data and show the first four rows from the resulting data.

```
library(MASS) # loads the library MASS that comes with R
data<-Boston[,c(-2,-3,-8)] # removes columns 2,3 and 8
head(data,4) # displays the first 4 rows of the data

##      crim chas   nox    rm  age rad tax ptratio  black lstat medv
## 1 0.00632    0 0.538 6.575 65.2   1 296    15.3 396.90  4.98 24.0
## 2 0.02731    0 0.469 6.421 78.9   2 242    17.8 396.90  9.14 21.6
## 3 0.02729    0 0.469 7.185 61.1   2 242    17.8 392.83  4.03 34.7
## 4 0.03237    0 0.458 6.998 45.8   3 222    18.7 394.63  2.94 33.4
```

We can see the names of the 11 variables left by calling `names(data)`.

```
names(data) # displays the names of all variables in data

##  [1] "crim"    "chas"    "nox"     "rm"      "age"     "rad"     "tax"
##  [8] "ptratio" "black"   "lstat"   "medv"
```

To find out more about the data set Boston, type `?Boston` at the R or RStudio console. This data has records from the 1980s for the median house value `medv` for 506 neighbourhoods around Boston. Our objective is to predict `medv` using one or more predictors, such as the average number of rooms `rm` and percent of households with low socioeconomic status `lstat`. We can visualize the sample distribution of the median house value (given in units of $1000s) by creating the frequency histogram, shown in Figure 13.

```
attach(data) # makes the variables inside data available to R
hist(medv,breaks=20,col="blue",border="white",xlab="Median House Value",
     main="Histogram of Median House Value")
```

We can also get the summary sample statistics for the median house value.

```
summary(medv) # sample statistics

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    5.00   17.02   21.20   22.53   25.00   50.00
```

If we are interested in houses next to Charles River, we can use the binary variable `chas` (1 next to the river, 0 otherwise) to extract `medv` for houses next to the river. The frequency histogram of `medv` for houses next to Charles river is shown in Figure 14.

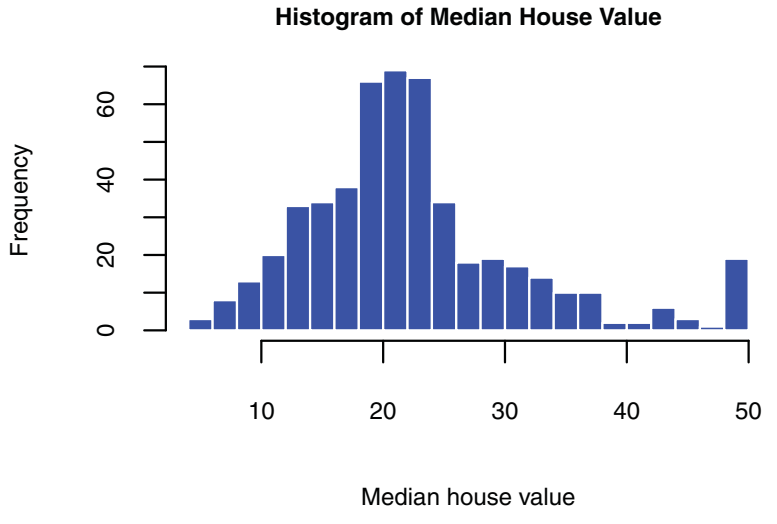

Figure 13. Distribution of median house value in thousands of dollars.

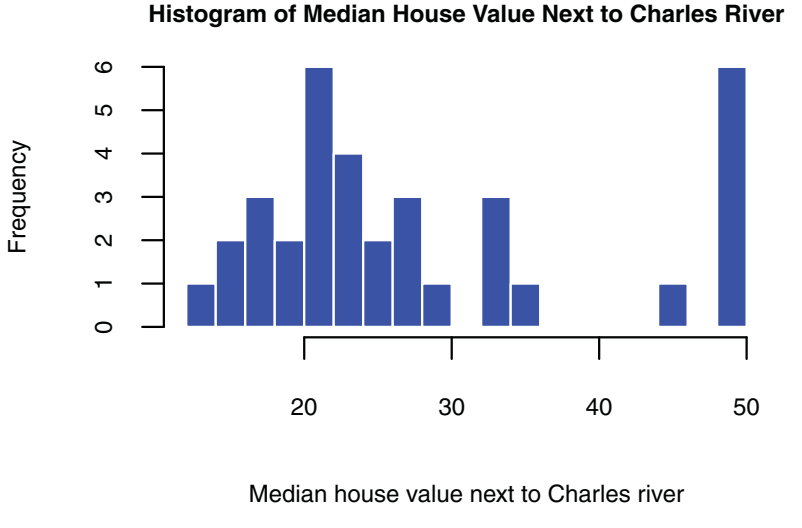

Figure 14. Distribution of median house value next to Charles river.

```
hist(medv[chas==1],col='blue',main='Histogram of Median House Value
     Next to Charles River',breaks=25,border="white",
     xlab="Median house value next to Charles river")
```

The summary sample statistics for the median house value next to Charles river:

```
summary(medv[chas==1]) # sample statistics

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   13.40   21.10   23.30   28.44   33.15   50.00
```

We can coerce the binary variable `chas` into a factor and rename the factor levels to `no river` and `river`, corresponding to 0 and 1. We can then use the R package `lattice` to visualize the data for the two factor levels by using a conditional plot and the formula

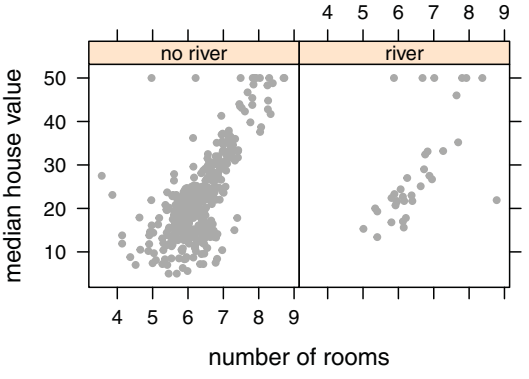

Figure 15. Conditional plot of data, conditioned on the river and no river factor levels.

object `medv~rm|chasf`, which creates two plots of `medv` vs. `rm`, one for each factor, `no river` and `river`. The conditional plot is shown in Figure 15.

```
library(lattice) # the lattice package must be installed first
chasf<-as.factor(chas) # convert chas to a factor
levels(chasf)<-c("no river","river") # rename the factor levels
# conditional plot with chasf as the conditioning factor
xyplot(medv~rm|chasf,pch=20,col="darkgray",xlab="number of rooms",
       ylab="median house value")
```

We can fit a linear regression model by using the (linear model) `lm()` function, with `medv` as the response and `lstat` as the predictor. The linear regression is done using the R formula object `mdev~lstat`.

```
fit<-lm(medv~lstat) # fitting a linear model by least squares
```

We can see the model fit statistics by calling `summary(fit)`.

```
summary(fit) # model fit statistics

##
## Call:
## lm(formula = medv ~ lstat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.168  -3.990  -1.318   2.034  24.500
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 34.55384    0.56263   61.41   <2e-16 ***
## lstat       -0.95005    0.03873  -24.53   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '  1
##
## Residual standard error: 6.216 on 504 degrees of freedom
## Multiple R-squared:  0.5441,Adjusted R-squared:  0.5432
## F-statistic: 601.6 on 1 and 504 DF,  p-value: < 2.2e-16
```

Since the p-value of the fit (`<2.2e-16`) is virtually zero, we can say that this is a statistically significant fit. The fitted linear model is given by:

$$\text{medv} = 34.554 - 0.95 \times \text{lstat}, \tag{37}$$

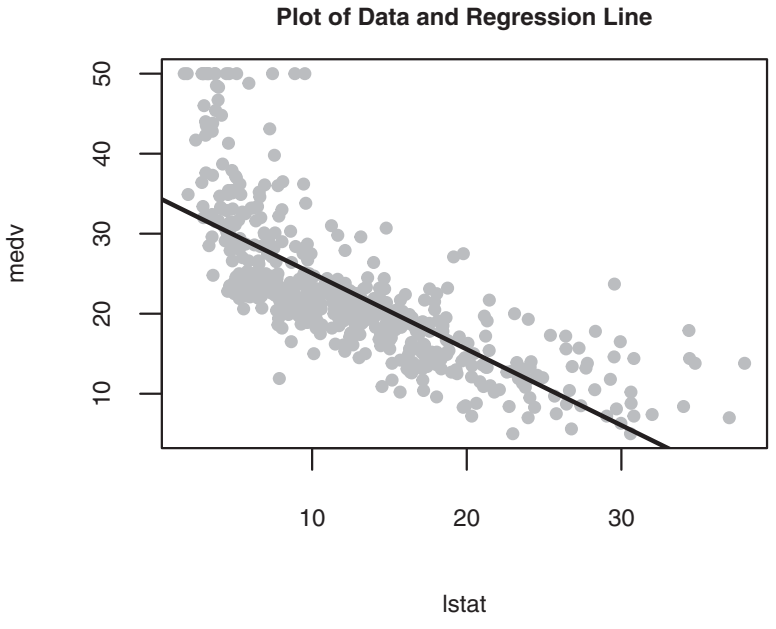

Figure 16. Plot of data and fitted regression line.

where the displayed coefficients are rounded off. We can extract the fitted model parameters using `coef(fit)`.

```
coef(fit) # y-intercept and slope of the fitted linear model

## (Intercept)       lstat
##  34.5538409  -0.9500494
```

We can create the scatterplot of `medv` versus `lstat` and add the regression line (37), using `abline()`. This is shown in Figure 16.

```
# scatterplot of medv vs. lstat
plot(lstat,medv, pch = 20, col = "darkgray",main="Plot of Data and
     Regression Line")
abline(fit, lwd = 2) # adding the regression line with line width=2
```

We can also fit a multiple linear regression model, using again the `lm()` function. The syntax `lm(y~x1+x2)` is used to fit a linear model with two predictors (`x1` and `x2`), given mathematically by the linear model $y = c_0 + c_1x_1 + c_2x_2$.

```
fit2<-lm(medv~lstat+rm) # linear regression with 2 predictors: lstat,rm
```

The fitted linear regression model is given by $\text{medv} = -1.36 - 0.64 \times \text{lstat} + 5.09 \times \text{rm}$. Since the p-value of the intercept (0.67) is very high, suggesting the intercept is statistically insignificant, we can fit the model without the intercept, that is by removing the constant term c_0 from the linear model.

```
fit3<-lm(medv~lstat+rm+0) # +0 removes the constant term (y-intercept)
```

The resulting linear model with no constant term, and improved R-squared, is given by:

$$\text{medv} = -0.66 \times \text{lstat} + 4.91 \times \text{rm} \tag{38}$$

We can make predictions using the fitted linear model and plot the residuals of the fit using the functions `predict()` and `residuals()`, respectively.

5.2. Analysis of categorical breast-cancer data

In this example, we use breast-cancer data, provided by the Machine Learning Repository at UC Irvine. The data file `breast-cancer.csv` can be downloaded from their data repository [34]. We did some basic pre-processing of the data in Excel and removed some variables and missing observations from the original data. All variables in this data are categorical, and when the data is loaded in R, the categorical variables are treated as factors, by default. We can load the data in RStudio by: calling `read.csv()`:

```
mydata<-read.csv("~/fullpath/breast-cancer.csv") # loading the data
```

which creates the dataframe `mydata`, and `fullpath` points the location of the data.

The data has 7 variables and 285 observations. The dimensions are obtained by calling the command `dim(mydata)`, and the names of the variables are retrieved with `names()`.

```
names(mydata)

## [1] "age"         "menopause"   "tumor.size"  "inv.nodes"   "deg.malig"
## [6] "breast"      "breast.quad"
```

We can peek into the data by calling the command `head()`.

```
head(mydata,3) # displays the first 3 rows of mydata

##     age menopause tumor.size inv.nodes deg.malig breast breast.quad
## 1 50-59   premeno      50-54      9-11         2  right     left_up
## 2 40-49   premeno      50-54       0-2         2  right    left_low
## 3 60-69      ge40      50-54       0-2         2   left    left_low
```

All variables in our data are categorical, and we can compute the frequency counts of each categorical data by using the `table()` function, which returns a vector with the frequency counts for the different levels of the factor variable. For example, we display the frequency counts for the `breast.quad` factor variable. Note that we can access individual variables from `mydata`, represented by the columns in the data, using the `$` operator for component extraction.

```
table(mydata$breast.quad) # frequency table

##
##   central  left_low   left_up right_low  right_up
##        21       110        97        24        33
```

In addition to the frequency counts for a single factor, the `table()` function can compute the two-way contingency table for any two factor variables. In addition, we can visualize the two-way contingency table by using the R package `vcd`, which allows for creating mosaic plots to visualize categorical data. Note that the first argument in the `mosaic()` function is a formula object `~breast+breast.quad` specifying the variables used to create a contingency table from the data.

```
library(vcd) # vcd must be installed first
mosaic(~breast+breast.quad, data=mydata,shade = F,keep_aspect_ratio=F,
     legend=F,main="Visual Contingency Table of Breast Cancer Locations")
```

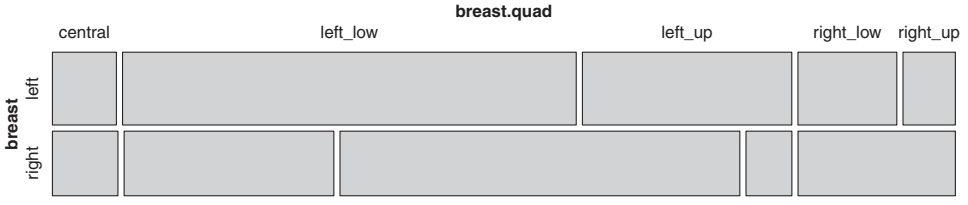

Figure 17. Mosaic plot visualizing the contingency table of breast cancer locations.

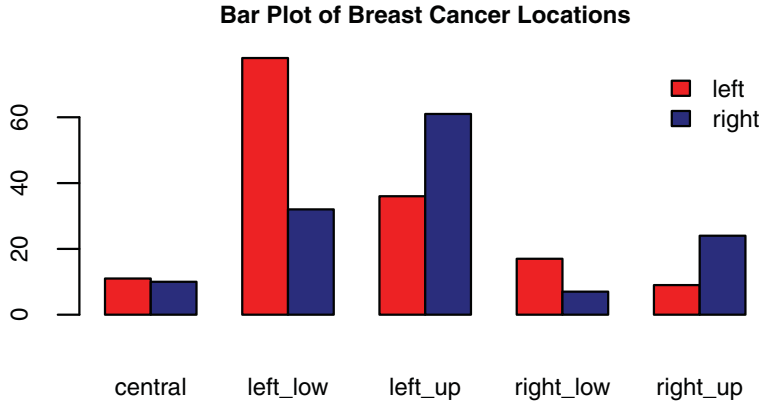

Figure 18. Grouped bar plot of breast cancer locations.

The mosaic plot in Figure 17 shows some evidence that the most frequent occurrences of breast cancer appear to be in the lower left part of the left breast, as well as in the upper left part of the right breast. We can get a different visual insight into this observation by creating a grouped bar plot for the two factors based on their contingency table. The grouped bar plot is shown in Figure 18.

```
counts <- table(mydata$breast,mydata$breast.quad) # 2-way freq. table
barplot(counts, main="Bar Plot of Breast Cancer Locations",
  xlab="", ylab="",col=c("red","darkblue"),
        legend = rownames(counts), beside=TRUE)
```

Both the mosaic plot and the grouped bar plot hint at the apparent symmetries between the left and the right breast in terms of the locations for breast cancer occurrences.

In addition, the contingency table provides a nice playground for computing conditional probabilities. We can compute any conditional probability for more complex contingency tables by using the `margin.table(table,margin=1)` function, which sums across margins (1 for rows and 2 for columns). Let us display the contingency table `counts` for the `breast` and `breast.quad` factors, and compute some probabilities of interest.

```
show(counts) # displays the 2-way freq. table

##
##         central left_low left_up right_low right_up
##   left       11       78      36        17        9
##   right      10       32      61         7       24
```

The probability to have a breast cancer occurrence in the left breast (event A), given that the cancer is in the lower left part of the breast (event B), is given by:

$$P(A|B) = \frac{78}{78+32} = 0.709 \tag{39}$$

The probability the breast cancer is in the left breast (event A) is given by:

$$P(A) = \frac{11+78+36+17+9}{285} = 0.53, \tag{40}$$

where the numerator in (40) is the sum of the elements in the first row of the contingency table, corresponding to left breast, and the denominator of 285 is the sum of all elements in the contingency table. The probability the breast cancer is in the lower left part of the breast is given by:

$$P(B) = \frac{78+32}{285} = 0.386 \tag{41}$$

The probability the breast cancer is in the lower left part of the left breast (event $A \cap B$)

$$P(A \cap B) = \frac{78}{285} = 0.274 \tag{42}$$

In particular, $\frac{P(A \cap B)}{P(B)} = 0.709$ is exactly equal to $P(A|B)$ (note that all numbers shown are rounded off), which illustrates an important result:

$$\textbf{Bayes' Theorem}: P(A|B) = \frac{P(A \cap B)}{P(B)} \tag{43}$$

In a similar way, students can investigate the relationships between all other categorical variables in the data and compute any relative frequencies of interest. Investigating real-world categorical data with cross-frequency tables, by using a hands-on computational approach based on R, is proving to have a good pedagogical value when introducing conditional probabilities and Bayes' Theorem, at any probability level.

6. Assessment results

Under the guidance of the Office of Assessment and Institutional Research at NYC College of Technology (CUNY), we conducted a survey on how students felt about using R in the classroom. Even in this small sample, we believe, there is evidence for the pedagogical effectiveness of using R as a scientific programming language for stochastic and deterministic modelling and simulation, data analysis, visualization, statistical computing, etc. As evident from the survey results, the majority of students were quite enthusiastic to have some hands-on experience using R and RStudio. The survey provides some evidence that using R improves students' understanding of difficult concepts, as well as their problem-solving, project-writing and presentations skills. In addition, some students appreciated the mar-

keting potential of R when it comes to looking for internships and industry jobs. We plan to conduct additional surveys at different course levels to better gauge students' sentiments towards using technology in general, and R, in particular.

6.1. Survey questions and data analysis

We conducted a survey consisting of ten questions, phrased positively towards the use of technology in general, and R in particular.

Question 1: How often had you used R and RStudio prior to taking this course?
Question 2: I find the R syntax to be simple and understandable for visualization and simulation purposes.
Question 3: I find it quick and easy to visualize mathematics and create simulations with R.
Question 4: Doing visualizations and simulations in R helps me understand better difficult concepts from stochastic modelling, probability and statistics.
Question 5: Using RStudio makes my homework and projects easier to read and understand.
Question 6: I like that I can create publication quality project reports either as pdf, word doc or html in RStudio with minimum effort.
Question 7: I enjoy using R and RStudio and I would like to use them in all courses dealing with visualization, data analysis and simulations.
Question 8: Using R for visualization and simulations gives me more confidence in solving problems and I believe I will get a higher grade thanks to using this technology.
Question 9: I would like to have a more comprehensive introduction to the programming capabilities of R and more time dedicated on learning the basics of R.
Question 10: I like using technology in the classroom because it gives me hands-on experience, improves my problem-solving, project-writing and communication skills.

Question 1 has five possible answers Q1A1, Q1A2, Q1A3, Q1A4, Q1A5, and questions 2–10 all have the same six possible answers A1, A2, A3, A4, A5, A6:

Question 1 Answers		**Questions 2-10 Answers**	
Q1A1:	had never heard of it	A1:	no opinion
Q1A2:	never	A2:	strongly disagree
Q1A3:	infrequently	A3:	disagree
Q1A4:	a few times	A4:	indifferent
Q1A5:	frequently	A5:	agree
		A6:	strongly agree

The survey was conducted in the single section of the upper-level Stochastic Modelling class and had a sample size of 17. The results show that 41% of students either had never heard of R before or never used it before, and 59% of students have used R before, either a few times or frequently, mostly in other upper-level classes offered by the department.

In Figure 19, we show the barplot for questions 2–10, based on the collected data.

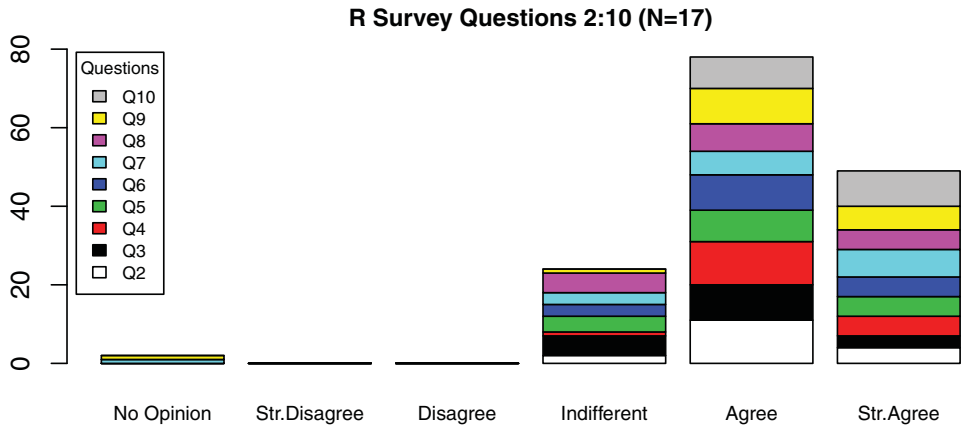

Figure 19. Barplot of collected survey data.

The relative frequency table for questions 2–10 is computed as follows:

```
##        A1 A2 A3    A4   A5   A6
## Q2  0.000  0  0 0.118 0.65 0.24
## Q3  0.000  0  0 0.294 0.53 0.18
## Q4  0.000  0  0 0.059 0.65 0.29
## Q5  0.000  0  0 0.235 0.47 0.29
## Q6  0.000  0  0 0.176 0.53 0.29
## Q7  0.059  0  0 0.176 0.35 0.41
## Q8  0.000  0  0 0.294 0.41 0.29
## Q9  0.059  0  0 0.059 0.53 0.35
## Q10 0.000  0  0 0.000 0.47 0.53
```

Based on the answers to questions 2–10, we have the following fractions of students who either *agree* or *strongly agree* with the premise of the respective questions:

Q2: 88.24% of students *agree* or *strongly agree* with question 2.
Q3: 70.59% of students *agree* or *strongly agree* with question 3.
Q4: 94.12% of students *agree* or *strongly agree* with question 4.
Q5: 76.47% of students *agree* or *strongly agree* with question 5.
Q6: 82.35% of students *agree* or *strongly agree* with question 6.
Q7: 76.47% of students *agree* or *strongly agree* with question 7.
Q8: 70.59% of students *agree* or *strongly agree* with question 8.
Q9: 88.24% of students *agree* or *strongly agree* with question 9.
Q10: 100% of students *agree* or *strongly agree* with question 10.

6.2. Reflections and critical perspectives

Fear of technology:. One possible challenge with integrating any technology into a project-based mathematics course, which is essential for completing all projects and homework, is that students who are more comfortable with technology would be advantaged over those who feel less comfortable and maybe even fear technology. This question remains to be investigated by conducting student surveys at the beginning and at the end of a technology-infused course, and looking for correlations with students' final grades.

Cost to the instructor:. From the point of view of the instructor, while there is some cost in time and effort to introducing R and R Markdown to the students, the long-term benefits are well worth it. We have developed a 50-page tutorial for our students covering the basics of R and R Markdown for creating project reports in RStudio, by combining computing with R and typesetting with LATEX. That was a time-consuming effort indeed, but a necessary one, so that we can quickly introduce our students to these modern technologies, in the first week of the course.

Common challenges with computing in R:. Students typically face a number of common challenges, including, but not limited to the following: improper use of R code chunks, invalid syntax for R and Markdown commands, failure to load required R packages, difficulties with reading external data files and properly formatting the output by changing plot sizes, as well as becoming familiar with other optional chunk arguments.

Typesetting with LATEX:. Most of the students are typically not familiar with LATEX but we only introduce the very basics of typesetting inline and display style mathematical expressions, so that we do not overwhelm our students with so many technologies.

Assessment challenges:. We started experimenting with low-stakes assessments to gradually build scaffolding of final, high-stake projects and presentations, as a way to introduce the students to hands-on, project-based learning in a less stressful and more writing-intensive approach.

Internship experience:. With the introduction of R in our upper-level courses, four of our students completed their required internships, using R and RStudio, based on projects assigned by one of the authors, while two other students completed external internships, which also required knowledge of R. In their own words, their active-learning, project-based, classroom experience using R was instrumental for the successful completion of the internship programs. This internship experience also inspired a couple of these students to think about pursuing Master's degrees in data and actuarial science.

6.3. Selected student comments from the survey

The best part about using technology at City Tech is that I get introduced to a range of software throughout my college career. Learning R, Maple and Matlab is the best way to prepare me for a future industry job.
The college should consider introducing R as early as possible, even for the elementary statistics class.
I really like using R. It helps me visualize problems and makes calculations much easier. The programming component is also useful experience you can use in real life jobs.
I would like to have more lectures that go over R and basic (R Markdown) formatting.
I would like to understand this software better. I find it easy to make mistakes without a proper understanding of the engine and the functions.

R is useful for job opportunities.

7. Conclusions

The R language provides high-level programming tools for simulation, visualization and data analysis that allow for rapid development and compact solutions of complex problems, with minimum effort. The examples discussed in this paper were designed to give a taste of the wide range of simulation and visualization capabilities offered by R. It is the authors' observation that students benefit extensively by using technology in a responsible way. Computational tools, such as R, lead to greater insight and conceptual understanding of complex problems and abstract concepts. The majority of students appear to enjoy and appreciate this hands-on experience as a useful one, not just for building computational, presentation and communication skills, but also for increasing their marketability at obtaining quality internships and jobs. As we conclude, it is essential that students do understand that technology should not be used as a substitute for rigorous proofs but only as a means to an end.

Acknowledgments

We want to thank the anonymous referees for their constructive criticism and many helpful comments and suggestions, which undoubtedly improved the quality of the paper. We also acknowledge the valuable help we received from Professor Arnavaz Taraporevala.

Disclosure statement

No potential conflict of interest was reported by the authors.

ORCID

Boyan Kostadinov http://orcid.org/0000-0003-0087-8078

References

[1] National Research Council. Report of a workshop on the scope and nature of computational thinking. Washington (DC): The National Academies Press; 2010. doi:10.17226/12840.
[2] Thomas MOJ, Lin C. Designing tasks for use with digital technology. In: Margolinas C, editor. Task design in mathematics education. Proceedings of ICMI Study 22; 2013; Oxford; 2013. p. 109–117.
[3] The R Project for Statistical Computing [Internet]; [cited 2016 Sept 23]. Available from: http://www.r-project.org
[4] National Research Council. Report of a workshop on the pedagogical aspects of computational thinking. Washington (DC): The National Academies Press; 2011. doi:10.17226/13170.
[5] National Council of Teachers of Mathematics. Principles to actions: ensuring mathematical success for all. Reston (VA): NCTM; 2014.
[6] RStudio [Internet]; [cited 2016 Sept 23]. Available from: http://www.rstudio.com
[7] GAISE College Group. Guidelines for assessment and instruction in statistics education. American Statistical Association; 2005. [Internet]; [cited 2016 Sept 23]. Available from: http://www.amstat.org/education/gaise
[8] KDnuggets [Internet]; [cited 2016 Sept 23]. Available from: http://www.kdnuggets.com/polls/2015/analytics-data-mining-data-science-software-used.html
[9] R Markdown: Dynamic Documents for R [Internet]; [cited 2016 Sept 23]. Available from: http://rmarkdown.rstudio.com

[10] Xie Y. Dynamic documents with R and knitr. 2nd ed. Boca Raton (FL): Chapman & Hall/CRC; 2015.
[11] Gandrud C. Reproducible research with R and RStudio. 2nd ed. Boca Raton (FL): Chapman & Hall/CRC; 2015.
[12] Lassak M. Effectively using multiple technologies. Int J Math Educ Sci Tech. 2015;46(5):783–790.
[13] Kostadinov B. Simulation insights using R. PRIMUS. 2013;23(3):208–223.
[14] Torfs P, Brauer C. A (very) short introduction to R [Internet]; [cited 2016 Sept 23]. Available from: http://cran.r-project.org/doc/contrib/Torfs+Brauer-Short-R-Intro.pdf
[15] Kabacoff R. R in action: data analysis and graphics with R. Shelter Island (NY): Manning Publications; 2015.
[16] Grolemund G. Hands-on programming with R. Sebastopol (CA): O'Reilly; 2014.
[17] Verzani J. Using R for introductory statistics. Boca Raton (FL): CRC; 2014.
[18] Bloomfield V. Using R for numerical analysis in science and engineering. Boca Raton (FL): CRC; 2014.
[19] Teetor P. R cookbook. Sebastopol (CA): O'Reilly; 2011.
[20] Benakli N, Satyanarayana A, Singh S, et al. Learning by visualizing. Proceedings of the American Society for Engineering Education, Mid-Atlantic Section Spring Conference; 2013 Apr 26–27; Brooklyn, NY. Brooklyn (NY): Computer Systems Technology Department; 2013. p. 158–175. Available from: https://www.asee.org/papers-and-publications/papers/section-proceedings/middle-atlantic/ASEE-Middle-Atlantic-Spring-2013-Proceedings.pdf
[21] Taraporevala A, Benakli N, and Singh S. Visualizing calculus by way of Maple. New York (NY): McGraw-Hill; 2011.
[22] Kostadinov B. Limiting forms of iterated circular convolutions of [lanar polygons. Int J Appl Comput Math. 2016 [cited 2016 August 12]; [20 p.]. doi:10.1007/s40819-016-0224-1
[23] Rudin W. Principles of mathematical analysis. 3rd ed. New York (NY): McGraw-Hill; 1976.
[24] Spivak M. Calculus. 4th ed. Houston (TX): Publish or Perish; 2008.
[25] Schroeder M. Fractals, chaos, power laws. New York (NY): W. H. Freeman; 1991.
[26] Gelbaum B, Olmsted J. Counterexamples in analysis. San Francisco (CA): Holden-Day; 1964.
[27] Watson G. Three triple integrals. Q J Math. 1939;10(1):266–276
[28] Nahin P. Inside interesting integrals. New York (NY): Springer; 2015.
[29] Ross S. Simulation. 5th ed. San Diego (CA): Academic Press; 2013.
[30] Boros G, Moll V. Irresistible integrals: symbolics, analysis and experiments in the evaluation of integrals. Cambridge: Cambridge University Press; 2004.
[31] Dahlke R, Fakler R. Applications of high school mathematics in geometrical probability. UMAP Module 660; 1985.
[32] Dobrow R. Probability: with applications and R. Hoboken (NJ): Wiley; 2014.
[33] Dahlke R, Fakler R. Applications of calculus in geometrical probability. UMAP Module 694; 1988.
[34] Machine Learning Repository [Internet]. Irvine (CA): UC Irvine [cited 2016 Sept 23]. Available from: https://archive.ics.uci.edu/ml/datasets/Breast+Cancer