

Computer simulated exercises

G. Palafox

November 9, 2020

In the following, exercises from the book of Grinstead and Snell [2006] are simulated computationally on a Jupyter [Kluyver et al., 2016] notebook¹ with R [R Core Team, 2020].

Exercise 1 (Ex. 19, p. 249). *A multiple choice exam is given. A problem has four possible answers, and exactly one answer is correct. The student is allowed to choose a subset of the four possible answers as his answer. If his chosen subset contains the correct answer, the student receives three points, but he loses one point for each wrong answer in his chosen subset. Show that if he just guesses a subset uniformly and randomly his expected score is zero.*

Computer simulation. The exercise is simulated as follows. A vector of possible answers, **a**, **b**, **c**, **d** is created. One of these is randomly selected as the correct answer. Then a random subset is taken from the possible answers, and a score is computed as per the rules laid out in the exercise. This is repeated 10,000 times to calculate the average score. The computation was done using the code in Listing 1. On average, the score obtained is -0.0129 , which is close to zero. A boxplot of the scores can be seen in Figure 1.

Listing 1: Code for Exercise 1.

```
answers = c('a', 'b', 'c', 'd')
scores <- numeric()
for(i in 1:10000){
  correct_answer <- sample(answers, 1)
  answerset <- sample(answers, sample(c(1,2,3,4), 1))
  if (correct_answer %in% answerset){
    score <- 3 - (length(answerset)-1)
  }else{
    score <- -length(answerset)
  }
  scores <- c(scores, score)
}
```

¹The notebook with the code for the experiments, as well as this report, can be found in the Github Repository: <https://github.com/palafox794/AppliedProbabilityModels/tree/master/Assignment10>

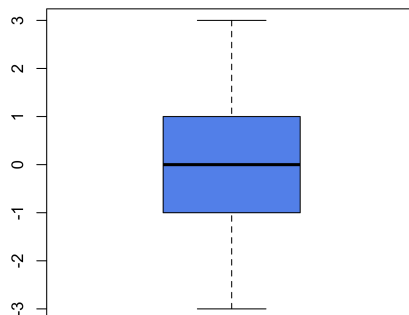


Figure 1: Scores obtained in experiment for Exercise 1.

Exercise 2 (Ex. 12, p. 264). Let X be a random variable with $\mu = \mathbb{E}(X)$ and $\sigma^2 = \text{Var}(X)$. Define $X^* = (X - \mu)/\sigma$. The random variable X^* is called the standardized random variable associated with X . Show that this standardized random variable has expected value 0 and variance 1.

Computer simulation. A thousand numbers following some specific distribution are generated. Their mean and variance is calculated. Then, the numbers are *standardized* as the exercise suggests, and the mean and variance are re-calculated. The mean and variance of the original and standardized values are stored, and this is repeated a thousand times. The code in Listing 2 shows a function for standardizing a vector, and the experiment performed for uniformly distributed numbers. The experiment was also performed for normal (mean 1, standard deviation 0.5) distributed numbers, and for exponential distributed numbers with rate 10. It can be seen in Figure 2 that the mean always shifts to zero, and variance always shifts to one, as expected.

Listing 2: Code for Exercise 2.

```
standardize <- function(vector){
  return ((vector - mean(vector))/sqrt(var(vector)))
}

means_unif <- numeric()
means_unif_std <- numeric()

var_unif <- numeric()
var_unif_std <- numeric()

for(i in 1:1000){
  x <- runif(1000)

  means_unif <- c(means_unif, mean(x))
  means_unif_std <- c(means_unif_std, mean(standardize(x)))

  var_unif <- c(var_unif, var(x))
  var_unif_std <- c(var_unif_std, var(standardize(x)))
}
```

Exercise 3 (Ex. 3, p. 278). The lifetime, measured in hours, of the ACME super light bulb is a random variable T with density function $f_T(t) = \lambda^2 t e^{-\lambda t}$, where $\lambda = 0.05$. What is the expected lifetime of this light bulb? What is its variance?

Computer simulation. Analytically, it can be shown the light bulb has an expected lifetime of 40 hours, with a variance of 800. In order to simulate the lifetime of the light bulb, a thousand random numbers with density $f_T(t) = \lambda^2 t e^{-\lambda t}$ were generated, and its mean and variance computed. This was repeated a thousand times, storing the mean and variance in each repetition. On average, the mean was 40.138 and the variance 798.26. The computations were performed with the code in Listings 3. The mean and variance obtained in each repetition are shown in the boxplot in Figure 3.

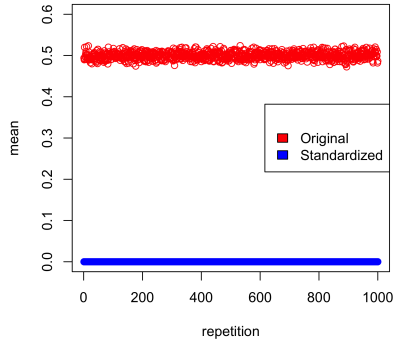
Listing 3: Code for Exercise 3.

```
library(distr)
p <- function(x){ (0.05**2) * (x*exp(-0.05*x)) } # probability density function
dist <- AbscontDistribution(d=p) # signature for a dist with pdf ~ p
rdist <- r(dist) # function to create random variates from p

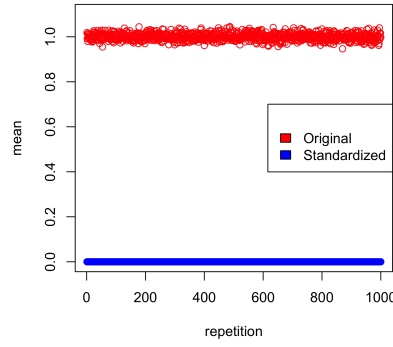
mean_x <- numeric()
var_x <- numeric()

for (i in 1:1000){
  X <- rdist(1000)
  mean_x <- c(mean_x, mean(X))
  var_x <- c(var_x, var(X))
}
```

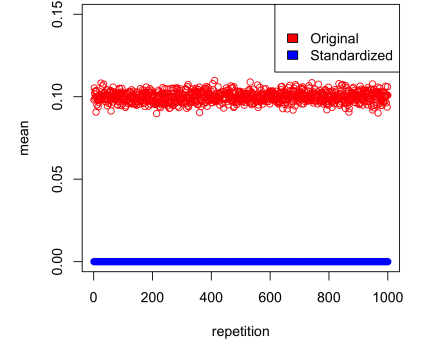
Exercise 4 (Ex. 1, p. 247). A card is drawn at random from a deck consisting of cards numbered 2 through 10. A player wins 1 dollar if the number on the card is odd and loses 1 dollar if the number is even. What is the expected value of his winnings?



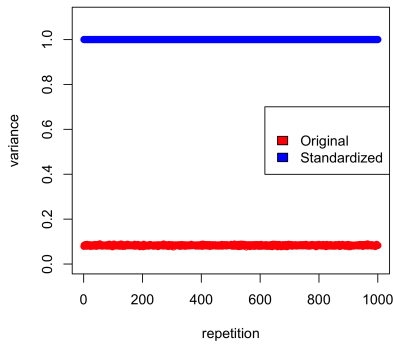
(a) Mean, with original numbers uniformly distributed.



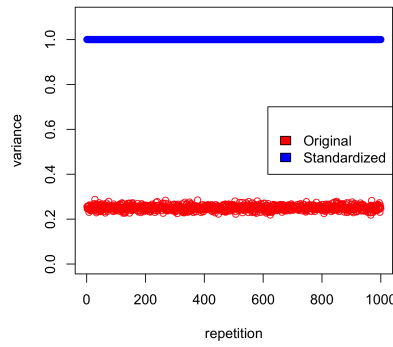
(b) Mean, with original numbers normally distributed.



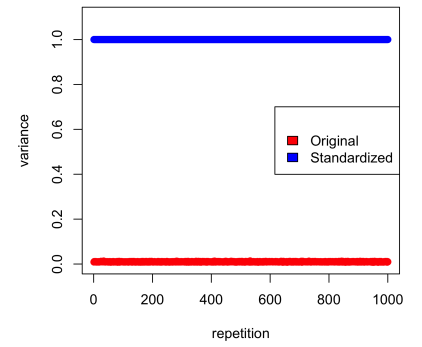
(c) Mean, with original numbers exponentially distributed.



(d) Variance, with original numbers uniformly distributed.

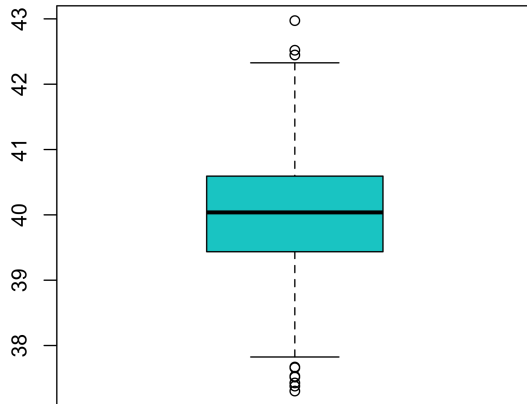


(e) Variance, with original numbers normally distributed.

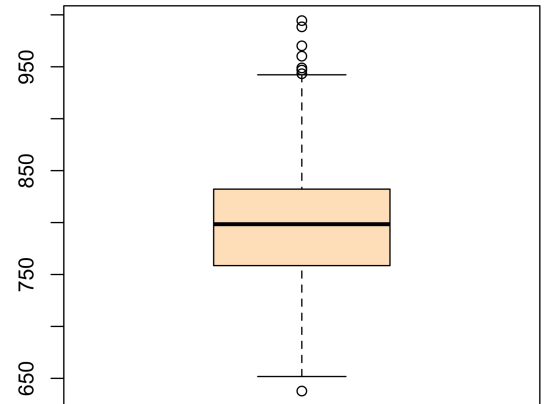


(f) Variance, with original numbers exponentially distributed.

Figure 2: Mean and variance before and after standardizing, Exercise 2.



(a) Mean in repetitions.



(b) Variance in repetitions.

Figure 3: Mean and variance in the experiment repetitions for Exercise 3.

Computer simulation. We simulate 50,000 repetitions of the experiment, drawing a random card and storing the corresponding score. A mean of -0.113 is obtained from this experiment, while analytically the expected value is $-1/9 \approx -0.111$. The code used for the simulation is in Listing 4.

Listing 4: Code for Exercise 4.

```
cards = c(2, 3, 4, 5, 6, 7, 8, 9, 10)

winnings <- numeric()

for (i in 1:50000){
  drawn_card <- sample(cards,1)
  winnings <- c(winnings, 2*(drawn_card %% 2) - 1)
}
```

References

- C. M. Grinstead and J. L. Snell. *Introduction to Probability*. 2006.
- T. Kluyver, B. Ragan-Kelley, F. Pérez, B. Granger, M. Bussonnier, J. Frederic, K. Kelley, J. Hamrick, J. Grout, S. Corlay, et al. Jupyter notebooks—a publishing format for reproducible computational workflows. In *Positioning and Power in Academic Publishing: Players, Agents and Agendas: Proceedings of the 20th International Conference on Electronic Publishing*, page 87. IOS Press, 2016.
- R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2020. URL <https://www.R-project.org/>.