Homework: Simulation Theory

Hayden Atchley

16 October 2023

1

The following terms are given along with their definitions: - **System:** A subsection of reality which we want to understand - **Experiment:** A controlled test used to gain information about a system - **Model:** An abstraction of reality used to understand a system - **Simulation model:** A model that uses algorithms to determine behavior of individual units and their effects on each other, often including a time element - **Analytical model:** A model that uses mathematical equations to determine the state of the system given specific inputs

2

I would model a Sodalicious drive-thru after General Conference priesthood session with a simulation, because there is a lot of randomness in arrival and potentially departure rates. A simulation would also be useful to model “knock-on” effects of the increased demand, such as queues backing up to other roadways, etc.

Water flowing through a hydraulic structure could be modeled analytically, as water generally behaves predictably and there is little randomness.

The relationship between flow and density on a freeway segment could be modeled either analytically or with a simulation. Many such analytical models exist (such as Greenshield’s), and depending on the use case predict traffic accurately enough. However, if more granular understanding is desired, a simulation may be better.

3

Equation 1 is a pseudorandom number generation algorithm which produces output values X between 0 and 1.

$$\begin{aligned} N_{i+1} &= (3945867 \times N_i) \mod 92348867 \\ X_{i+1} &= \frac{N_i}{92348867} \end{aligned} \tag{1}$$

This algorithm is implemented in R as follows:

```
p1 <- 3945867
p2 <- 92348867

ni <- function(n, pmult = p1, pmodulo = p2) (pmult*n) %% pmodulo
xi <- function(ni, pmodulo = p2) ni/pmodulo
```

Using a seed of 8.9347856×10^7, the first few iterations of this algorithm give the following values:

iteration	n	x
1	89347856	0.9675035
2	27897272	0.3020857
3	91348361	0.9891660
4	50455548	0.5463581
5	22804964	0.2469436
6	64785786	0.7015331
7	24767210	0.2681918
8	20467187	0.2216290
9	51295023	0.5554483
10	17994366	0.1948520
11	25103702	0.2718355

To see if this algorithm is a good pseudorandom number generator, the algorithm is run for 10^6 iterations, and Figure 1 shows a histogram of the results. This figure shows that the algorithm is fairly well distributed, though it is not perfect.

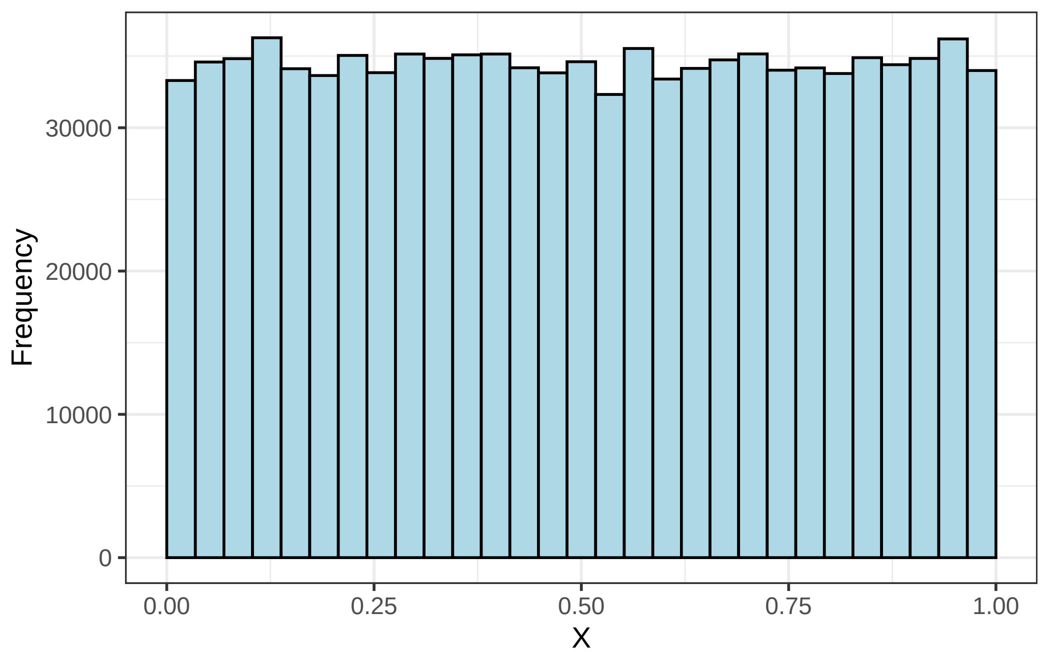

Figure 1: Histogram of the first 1,000,000 iterations of the algorithm in Equation 1.