Simulation

Owen Ward

2021-03-05

Simulation and randomness occur throughout statistics and data science. We will see how to generate this randomness and some examples of how it can be used.

Why simulation is useful

- Simulation is an extremely powerful method which can be used to help solve lots of difficult problems and also to give insights which would not be available otherwise.
- Statistics is the study of randomness and therefore it seems reasonable to incorporate randomness into solving problems in statistics and data science.

More robust predictions

One simple example where simulation can be used is in making better predictions. For example, FiveThirtyEight, which has provided some of the best U.S election predictions in the past decade uses simulation to give their final predictions (and add the required uncertainty).

- See for example https://projects.fivethirtyeight.com/2021-nba-predictions/?ex_cid=rrpromo
- Simulate several thousand games to give win projections.

How do we do this?

Generating random numbers

- To do simulations we need to create randomness and to do this random numbers are needed.
- How can we generate random numbers? What if thousands of numbers are needed?
- How have people attempted to generate random numbers in the past?

Simple random number generators?

- One (perhaps) obvious way to get random numbers is to flip a coin or roll a dice. But this only gives either 0/1 or 1 to 6.

Simple random number generators?

- One (perhaps) obvious way to get random numbers is to flip a coin or roll a dice. But this only gives either 0/1 or 1 to 6.
- Could use multiple coin flips to get random numbers in binary and then convert. Binary numbering has been around for centuries.

Simple random number generators?

- One (perhaps) obvious way to get random numbers is to flip a coin or roll a dice. But this only gives either 0/1 or 1 to 6.
- Could use multiple coin flips to get random numbers in binary and then convert. Binary numbering has been around for centuries.
- People have also used what they believed to be random digits, such as the digits of π. [1]

More recently

- In the early 20th century, researchers managed to collect random numbers by analysing random electrical pulses and collected sequences of millions of random numbers in books for other people to use. [2]

More recently

- In the early 20th century, researchers managed to collect random numbers by analysing random electrical pulses and collected sequences of millions of random numbers in books for other people to use. [2]
- One way to get “truly” random numbers is to look at recordings of atmospheric noise. No idea how it is generated. These can be accessed at https://www.random.org/

Pseudo Random

- While some early computers generated true random numbers as above, today more computers compute pseudo-random numbers, numbers that appear random but are actually deterministic.
- Generally uses a method related to number theory called the Mersenne Twister. [3]
- This uses complicated number theory methods to produce numbers that “look” random and have properties of random numbers. These are sufficient for simulations.

In R

- R most easily generates random numbers in the interval $[0, 1]$. To generate from a different range it does this and then transforms them as appropriate.

```r
runif(1) # generates 1 random number on [0,1]
```

```
## [1] 0.9781024
```

```r
runif(3, min = -5, max = 5)
```

```
## [1] -4.920016 -4.797747 -3.281511
```

```r
runif(5, min = 0, max = 2)
```

```
## [1] 1.3071228 0.2172942 1.0672699 1.1671243 0.5450690
```

```r
# which is basically the same method as
2*runif(5, min = 0, max = 1)
```

```
## [1] 0.2764256 1.4952220 1.8433604 1.7995944 1.9703814
```

More generally

- Similarly we can generate from many other statistical distributions

```
rnorm(n = 3, mean = 1, sd = 2)
```

```
## [1] -1.9474868  2.1600487 -0.7239727
```

```
rpois(n = 3, lambda = 2)
```

```
## [1] 0 2 3
```

When to use simulation

- Sometimes questions of interest are simply to difficult to answer exactly. However, using simulation we can often (quite easily) get approximate answers which are close to the true answer.
- These can be mathematical problems (high dimensional integrals) or more straightforward problems such as

  Suppose an elevator cable will break if there are more than 1600 pounds in the elevator. If 8 people get in, what is the probability the cable will snap?

The elevator problem

Suppose an elevator cable will break if there are more than 1600 pounds in the elevator. If 8 people get in, what is the probability the cable will snap?

- Well known that weights are almost normally distributed. Lets say approximately has mean 180 pounds with standard deviation 20 pounds.
- So then we just want a simulation where, many times, we pick 8 random people and see how much they weigh.

```r
weights <- rnorm(n = 8, mean = 180, sd = 20)
sum(weights)
```

```
## [1] 1499.946
```

```r
greater <- c(rep(0,1000))
for(i in 1:1000){
  weights <- rnorm(n = 8, mean = 180, sd = 20)
  if(sum(weights) > 1600){
    greater[i] <- 1
  }
}
sum(greater)/1000
```

```
## [1] 0.001
```

For such small probabilities might need to do more simulations to get an accurate answer.

```r
greater <- c(rep(0,10000))
for(i in 1:10000){
  weights <- rnorm(n=8, mean = 180, sd = 20)
  if(sum(weights) > 1600){
    greater[i] <- 1
  }
}
sum(greater)/10000
```

```
## [1] 0.0019
```

Monte Carlo Simulation

Monte Carlo Simulation

- Suppose we didn't know how to compute the integral

$$\int_0^1 e^x dx$$

- We can solve this using Monte Carlo

Integrals

- Simulate numbers randomly over the range of x, here $[0, 1]$, calling them $x_1, \dots, x_n$.
- At each of these points we compute e^x
- Take the average of these values as our estimate $T_n = \frac{1}{n}\sum_{i=1}^{n} e^{x_i}$.
- As n increases then by the Law of Large numbers T_n will get closer to the true value of the integral $(e - 1)$.

High Dimensions

- While this is a simple example the exact same method works in higher dimensions, although the number of simulations needed to get a good estimate increases at an exponential rate.
- Much research is devoted to developing better ways to do these approximations.

Another example

Suppose you have two machines which make screws. In machine 1 the lengths are normal with mean 3 inches and standard deviation 0.5 inches. In machine 2 they have mean 2.5 inches and standard deviation 0.75 inches. If you pick one from each, what is the probability the screw from machine 1 is longer?

- If you know something about normal distributions this is straightforward. But can still be solved using only simulation.

```r
n <- 1000
m1 <- rnorm(n, mean = 3, sd = 0.5)
m2 <- rnorm(n, mean = 2.5, sd = 0.75 )
length(which( m1 > m2 ))/n
```

```
## [1] 0.732
```

The Monty Hall Problem

The Monty Hall Problem

This is a nice example of a problem which might appear difficult at first, but can be easily solved with some simulation.

> *Suppose you're on a game show, and you're given the choice of three doors: Behind one door is a car; behind the others, goats. You pick a door, say No. 1, and the host, who knows what's behind the doors, opens another door, say No. 3, which has a goat. He then says to you, "Do you want to pick door No. 2?" Is it to your advantage to switch your choice?*

Think about it for a few minutes?

The solution

- There are many methods to get to the solution here, including conditional probability and explicitly working through all the solutions. However an approximate solution can also be obtained quite easily by simulation.

R code

```r
doors <- c("A","B","C")
outcome <- c()
for(i in 1:100){
  prize <- sample(doors,1)
  pick <- sample(doors,1)
  open <- sample(doors[which(doors != pick &
                               doors != prize)],1)
  switch_outcome <- doors[which(doors != pick &
                                  doors != open)]
  if(pick == prize){
    outcome = c(outcome,"NoSwitchWin" )
  }
  if(switch_outcome == prize){
    outcome = c(outcome,"SwitchWin")
  }
}
summary(as.factor(outcome))
```

More simulations

Try $n = 1000, 100000$.

```
## NoSwitchWin   SwitchWin
##         360         640
```

```
## NoSwitchWin   SwitchWin
##       33160       66840
```

- Getting closer and closer to the true value, with win probability of $2/3$ if you switch.

Fake data simulation

- This is a really simple way to check your model fits the data well.
- Fit your model to the data and use the model to predict new “fake” data.
- Compare this fake data to the true data and see if they look inherently different.
- If they do could indicate a poor model which needs to be updated.

Accuracy

- How do we know that, as we take more and more simulations, an estimate is guaranteed to get closer and closer to the true answer?
- In many settings, without many assumptions, it can be shown that these simulation estimates do get close to the true value.
- Simularly, this is a statistical method, so it is interesting to see what type of distribution the estimate converges to.
- This can also be proved in general, by the Law of Large Numbers and the Central Limit Theorem.

Buffons Needle

The problem

In the 18th century, the following problem was posed by a French aristocrat, Georges-Louis Leclerc, Comte de Buffon.

Suppose we have a floor made of parallel strips of wood, each the same width, and we drop a needle onto the floor. What is the probability that the needle will lie across a line between two strips?

The Problem

This is a problem in geometric probability and can be solved exactly using calculus, to give an answer in terms of the length of the needle l and the gap between the strips of wood, t. For $l < t$ then this probability is

$$P = \frac{2l}{t\pi}$$

Getting the answer

- The distance from the center of the needle to the nearest line is a random number between 0 and $t/2$. The angle between the needle and this nearest line is random between 0 and 90 degrees (or, in radians, between 0 and $\pi/2$).
- The needle crosses the line if

$$x \leq \frac{l}{2}\cos\theta$$

- This probability is given by

$$\int_{\theta=0}^{\pi/2}\int_{x=0}^{(l/2)\cos\theta}\frac{4}{t\pi}dx d\theta = \frac{2l}{t\pi},$$

when the needle is shorter than the gap.

Estimating Pi

Estimating Pi

For centuries, people have spent time constructing ways to give more and more accurate estimates for π, an irrational, transcedental number which has infinite digits in it's decimal expansion.

Estimating Pi

- Several methods have been devised to estimate π numerically, the simplest being to attempt to accurately measure the circumference of a circle of known radius.
- Laplace realised in 1812 that π could be estimated by using the needle experiment.
- If n needles are dropped and h of them cross the line, then we would estimate P with $\frac{h}{n}$ and rearrange the previous equation to give
$$\pi \approx \frac{2ln}{th}.$$

Doing this simulation

- Pick our $l < t$ and the number of needles to drop n
- For each needle generate a random x and θ and if $x \leq l/2 \cos \theta$ add than needle to h, the number which cross the lines.
- We can do this easily in R.

Some code to show this

```
l = 1
t = 2
n = 10000
x = runif(n, min = 0, max =  t/2)
theta = runif(n, min=0, max = pi/2)
length_line = l/2*cos(theta)
h =length(which(x<length_line))
estimate = 2*l*n/(t*h)
estimate
```

```
## [1] 3.075031
```

- Increasing n will give a more accurate answer.

Some problems

- One problem with this code is that to estimate π we need to use π to simulate the random angles.

Some problems

- One problem with this code is that to estimate π we need to use π to simulate the random angles.
- How accurate is pi, the estimate stored in R?

Some problems

- One problem with this code is that to estimate π we need to use π to simulate the random angles.
- How accurate is pi, the estimate stored in R?
- Would we do better with a better estimate of π?

Lazzarini

- In 1901, an Italian mathematician Lazzarini claimed to have done this experiment over 3000 times by hand, getting a correct answer to 6 decimal places.
- Given the intermediate numbers he recorded, the probability of getting the answer he got it extremely small, indicating he may have continued the experiment until he got a number he wanted. [4][5]

References

1. Are the digits of π random? Paul Preuss https://www2.lbl.gov/Science-Articles/Archive/pi-random.html
2. https://www.amazon.com/Million-Random-Digits-Normal-Deviates/dp/0833030477
3. https://en.wikipedia.org/wiki/Mersenne_Twister
4. https://en.wikipedia.org/wiki/Buffon%27s_needle
5. https://www.maa.org/sites/default/files/pdf/upload_library/22/Allendoerfer/1995/Badger.pdf