

# Introduction to Ecological Analyses in R (Day 5): Monte-Carlo Methods and Bayesian Beginnings

Matthew K. Lau

## 1 Monte Carlo Simulations: Coins, Dice and Kangaroos

Back up and forget everything you know about statistics. What do we want to know? Answers to ecological questions. Statistics is a means to make probabilistic statements related to questions of interest, when we don't have complete information. The statistics that are typically taught in introductory courses or frequentist-based, parametric statistics. That is certain aspects about the true population values of interest, such as the form of the distribution, can be assumed. However, there are many instances where we either cannot assume a distributional shape or, even if we do, analytical solutions for mathematical quantities are not approachable or even possible. Monte-Carlo methods provide a way to get around such analytical limitations and have made many non-parametric and Bayesian analyses possible.

Monte-Carlo (MC) methods are rooted in gambling probability (hence the name in reference to the famous gambling city in Europe). Simply, MC methods repeat a physical process in order to obtain quantitative estimates of the probability of particular events. This can be done in "reality" or by simulation. Simulation provides a much more practical and consistent means. At the heart of the simulation is an algorithm (i.e. mathematical representation) that represents the physical process. Here we go over several simple examples of MC methods.

### 1.1 Coin Games: simulating a coin flip

#### 1.1.1 What is the probability of flipping a heads with a fair coin?

First, we need to make representation of our "fair" coin, which will have two sides that have equal probability of coming "up" when the coin is flipped. We can do this by creating an object with two values that will represent heads and tails:

```
> coin <- c("H", "T")
```

Next, we need to be able to "flip" our coin. This can be achieved by using the `sample` function:

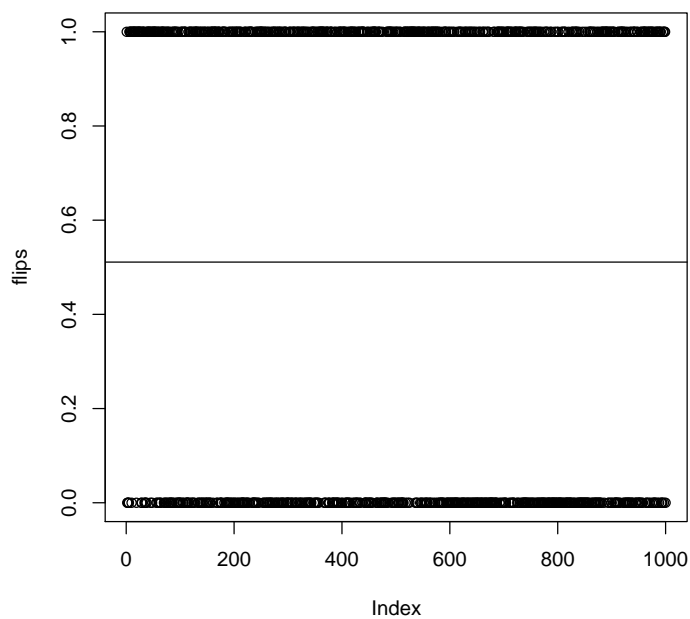
```
> sample(x = coin, size = 1, replace = FALSE, prob = NULL)
```

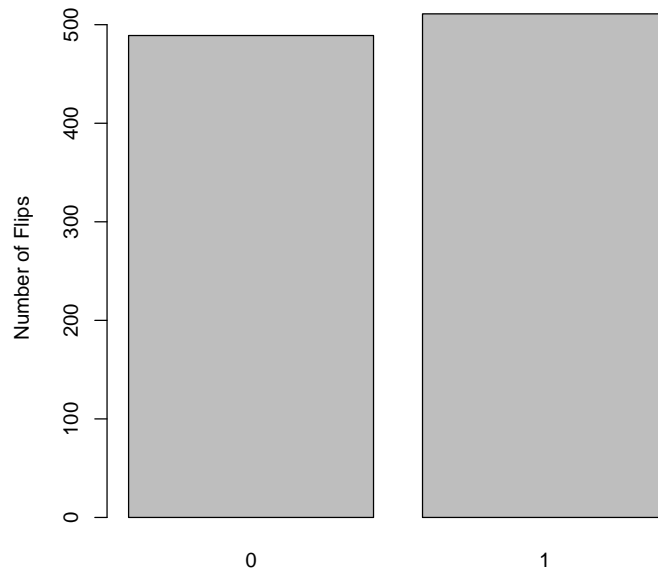
```
[1] "H"
```

Here you can see that the function drew a "sample" of size one from our object "coin." The latter two arguments specify whether or not to choose previously chosen values from the object when the sample size is larger than one and the probability of sampling a particular value in the object being sampled where the default (i.e. NULL) applies equal probability to all values.

Last, we use a for-loop to apply an algorithm that simulates many "flips" of our coin (i.e. repeated sampling of our coin vector).

```
> coin <- c(0, 1)
> flips <- numeric()
> for (i in 1:1000) {
+   flips[i] <- sample(x = coin, size = 1, replace = FALSE, prob = NULL)
+ }
> plot(flips, type = "p")
> abline(h = mean(flips))
```





Looking at the first plot, we can see that the outcome of each one of our flips tend to come up pretty evenly as tails (=0) and heads (=1). The center line here shows the mean value of our simulation, which in this case would be equivalent to the probability of the flip coming up heads. We can calculate the probability of either heads or tails by counting the number of flips that produced either one and dividing by the total number of flips:

```
> p.heads = length(flips[flips == 1])/length(flips)
> p.tails = length(flips[flips == 0])/length(flips)
> cbind(p.heads, p.tails)

      p.heads p.tails
[1,]  0.511  0.489
```

Both of these probabilities are very close to the analytical solutions:  $P(\text{heads}) = 1/2 = 0.5$  and  $P(\text{tails}) = 1/2 = 0.5$ . However, the values are just a little off of the mark. This is because there is a small amount of random error inherent to the MC algorithm. Theoretically, as we increase the number of simulations toward infinity, this error would diminish to zero. It is generally recommended to not only run a large (>1000) number of simulations, but also assess the MC error of the simulation by looking at the variance of multiple MC simulations (i.e. repeat the MC simulation over and over, each time calculating the probability value of interest and then calculate the variance of the simulated probabilities). This is an instance where writing functions can really help to reduce the length of the script!

## 1.2 Dice: craps simulation

Here is another example where the analytical solution is easy to obtain, giving us an easy way to confirm our MC simulation. Say we are going to Las Vegas and we would like to try our luck at the craps table. Let's figure out the probability of rolling a 7 or 11 in one roll of two dice.

This is easy to do analytically:

	1	2	3	4	5	6
1	2	3	4	5	6	7
2	3	4	5	6	7	8
3	4	5	6	7	8	9
4	5	6	7	8	9	10
5	6	7	8	9	10	11
6	7	8	9	10	11	12

The above table shows all of the possible outcomes of one roll of two dice. From this we can calculate the analytical solution which is  $P(7) + P(11) = 6/36 + 2/36 = 0.17 + 0.06 = 0.23$ .

Now let's make a simulation to compute this probability using a strategy similar to the coin flip simulation we did above.

1. Make two "fair" dice
2. Simulate 1000 rolls of the two dice
3. Calculate the number of times either a 7 or 11 come up

```
> die1 <- c(1, 2, 3, 4, 5, 6)
> die2 <- c(1, 2, 3, 4, 5, 6)
> roll <- numeric()
> n <- 1000
> for (i in 1:n) {
+   d1 <- sample(die1, 1)
+   d2 <- sample(die2, 1)
+   roll[i] <- d1 + d2
+ }
> length(roll[roll == 7])/length(roll) + length(roll[roll == 11])/length(roll)

[1] 0.203
```

As you can see this MC simulation solution is very close to the analytical solution, but just a little off, as with the coin example above.

### 1.3 Monte Carlo Test of Two Samples from the Same Population: Boxing Kangaroos

A well-respected biologist suggests to you that kangaroos from W. Australia are better boxers (i.e. better intra-specific competitors) than kangaroos from E. Australia, due to selective pressure from greater resource limitation in the West. You set out to test this hypothesis. After collecting kangaroo boxing ability estimates, which continuous and range from zero to infinity, from both sides of the continent, you wish to test the hypothesis that the western population has no better or worse boxing ability than the eastern population.

#### 1.3.1 We can test this with a Monte-Carlo simulation, where we simulate samples from a common (i.e. "Null") distribution.

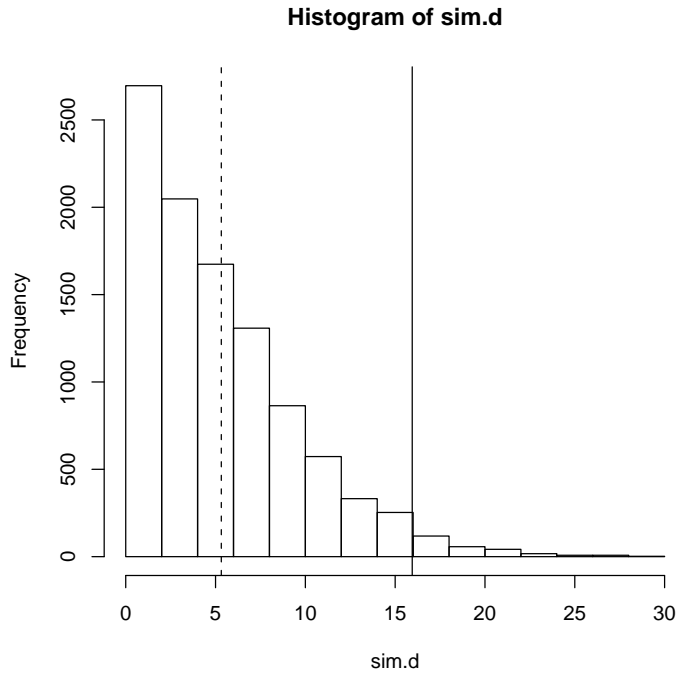
```
> W.ability <- c(25, 32, 80, 10, 22, 23.5, 20, 19, 23, 19.5)
> E.ability <- c(10, 11, 12, 11, 10.5, 9, 12, 10, 19, 10)
> null <- c(W.ability, E.ability)
> n = 10000
> obs.d <- abs(mean(W.ability) - mean(E.ability))
> sim.d <- numeric()
> for (i in 1:n) {
+   W <- sample(null, 10, replace = TRUE)
+   E <- sample(null, 10, replace = TRUE)
+   sim.d[i] <- abs(mean(W) - mean(E))
+ }
> length(sim.d[sim.d >= obs.d])/n

[1] 0.0264

> p.value <- length(sim.d[sim.d >= d])/n
> p.value

[1] 0.0264

> hist(sim.d)
> abline(v = c(mean(sim.d), obs.d), lty = c(2, 1))
```



Taking a closer look at the simulation, what’s happening is that we mixed together all of our observations and drawing two samples, like we did in pseudo-reality, from this over and over ( $n=1000$  times), each time calculating a value for our statistic ( $d$ ), which is the absolute value of the difference between our means from our two samples. We can then use this to as a simulated null distribution, which we then use to calculate our p-value by counting the number of times that the simulated  $d$  is greater than or equal to our observed  $d$  and dividing by the total number of simulations.

Although MC methods are non-parametric (i.e. there is no assumptions about the population distribution), there are still assumptions that are important to be aware of:

- 1.The samples are representative (i.e. random),
- 2.The Monte-Carlo algorithm is actually doing what we want (this is often not always possible to assess).

For more information on MC methods, see this online [bibliography](#).

## 2 The Bayesian Stats MCMC Revolution

Bayesian Statistics are characterized by their use of a subjective definition of probability: the degree of belief in the occurrence of an event. This is distinct and more general than the Frequentist definition of probability, which is based

on the long run frequency of an event. The results of Bayesian Statistical approaches are often more intuitive, since they typically give us direct estimates of our hypotheses of interest, rather than awkwardly tiptoeing around with p-values in the Null Hypothesis Testing framework that is the most prevalent approach. Also, Bayesian methods are logically in line with the scientific method, because they develop a subjective estimate of probability, as a degree of belief, in a particular hypothesis by quantitatively modifying prior beliefs (prior probability) with current observations (likelihood) to generate an updated probability estimate (posterior probability).

But why talk about Bayesian methods in conjunction with MC methods? Because, MC methods are at the core of a modern revolution in Bayesian Statistics. Observe Bayes' theorem, which allows one to calculate the posterior probability:

$$Posterior = \frac{Prior * Likelihood}{Marginal Probability}$$

Ignoring the denominator for a second, the numerator shows us that Posterior is simply a modification of the Prior by the Likelihood, which is our data that we have in hand. In other words, we get an estimate of the probability of our hypothesis of interest, which must be quantitative, collect data and calculate the Likelihood, which is the probability of observing our data given our hypothesis, multiply these two together. This however only produces a value proportional to the Posterior probability. To get the true Posterior probability estimate, we must divide by the Marginal Probability, which is simply a normalizing constant.

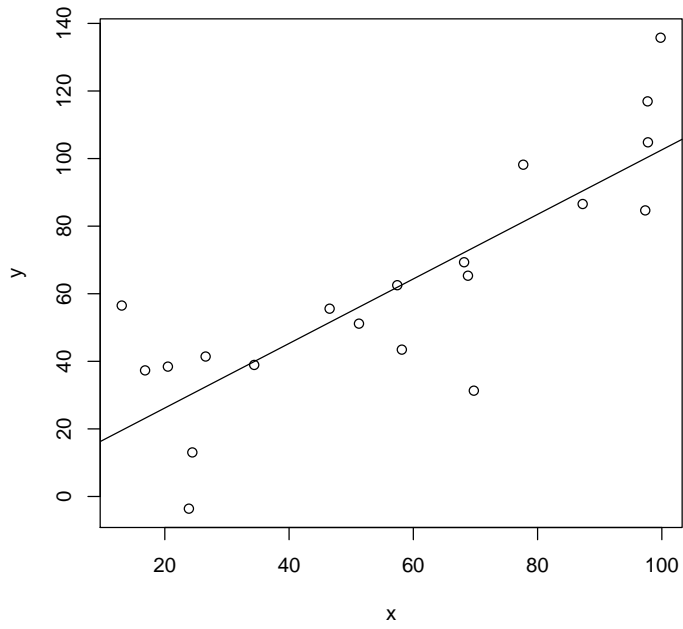
Until MC methods were developed to calculate the Marginal Probability, Bayesian Statistics was often limited, because an analytical solution for the Marginal Probability were intractable at best, and analytically impossible at worst. The development of MC methods, specifically Markov Chain Monte-Carlo (MCMC), have allowed for easy, quick computation of the Marginal Probability.

## 2.1 Example: Bayesian Regression

Here, we use functions in the *MCMCpack* package to conduct a Bayesian regression.

First we create a simple data set.

```
> x <- runif(20, 10, 100)
> y <- rnorm(20, x, 4^2)
> plot(y ~ x)
> abline(lm(y ~ x))
```



Now, we load up our *MCMCpack* functions and conduct our Bayesian regression:

```

> library(MCMCpack)
> mcmc.output <- MCMCregress(y ~ x)
> xtable(summary(mcmc.output)[1]$statistics, caption = "Bayesian Results1")

```

	Mean	SD	Naive SE	Time-series SE
(Intercept)	7.22	10.59	0.11	0.10
x	0.95	0.17	0.00	0.00
sigma2	465.97	178.26	1.78	2.35

Table 1: Bayesian Results1

```

> xtable(summary(mcmc.output)[2]$quantiles, caption = "Bayesian Results2")

```

	2.5%	25%	50%	75%	97.5%
(Intercept)	-13.53	0.49	7.30	13.98	28.29
x	0.62	0.85	0.95	1.06	1.28
sigma2	236.90	343.20	428.94	543.97	905.77

Table 2: Bayesian Results2

```

> xtable(summary(lm(y ~ x)), caption = "Frequentist Results")

```

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	7.0849	9.9123	0.71	0.4839
x	0.9550	0.1550	6.16	0.0000

Table 3: Frequentist Results

What has happened here is that a MC procedure utilizing a Markov-Chain algorithm has generated posterior probability estimates of our parameters. In this case these are the intercept (which we are not particularly interested in in this situation), the slope and the variance of the slope parameter. We can compare the Bayesian results (upper) to a least squares regression analysis (lower) and see that we come to very similar conclusions. In the case of the Bayesian inference, we get a posterior probability distribution for values of our slope parameter, which can be used to calculate credibility intervals, which can be interpreted as the probability of the true value being within that interval.

We could conclude here that the true slope parameter is not equal to zero with 95% probability, which is a similar, but not identical, conclusion that we would come to with the frequentist, null hypothesis approach. However, there is more that can be done with the posterior probability distribution. Two things, which

I will discuss briefly and refer you to the web to learn more about, are multi-model inference and prediction. Bayesian multi-model inference is very similar to frequentist based multi-model inference (often referred to as AIC for the most widely used information criterion, Akaike's Information Criterion) except that the posterior probabilities would be used to compute estimates of relative model performance. Also, the posterior probability distribution can be used to generate predictions about our parameters of interest. This allows one to use all available information at hand in a systematic analytical framework to predict future values, which is especially useful for reducing variance in estimates by using prior information to augment noisy data.

For a great introduction to Bayesian applications in ecology, read either (or both) of these books:

*Bayesian Methods for Ecology* by Michael A. McCarthy

*A Primer of Ecological Statistics* by Nicholas Gotelli and Aaron Ellison