monte carlo error

Probability and Inference — Deliverable 02

Monte Carlo Error

Rastko Stojsin

Simulations are commonly used when the math required to answer a problem is complex and would take a lot of time to work out. Simulations can be used and may be quicker to run than figuring out how to setup and solve complex math problems.

One issue with simulations is that there is degree of error within them due to the randomness needed for them to exist. Thus a simulation generates approximate answers; there is some degree of error in a quantity estimated by Monte Carlo simulation. Intuitively, it seems that the degree of error should get smaller as the number of simulation replicates increases. I will test this and explain why the concept is important through new introduced key vocabulary terms and demonstrate these errors in action.

Absolute Error

I will look at absolute errors of the calculation that is coded down below. Absolute error is simply the absolute difference of the simulations estimate and the true underlying probability.

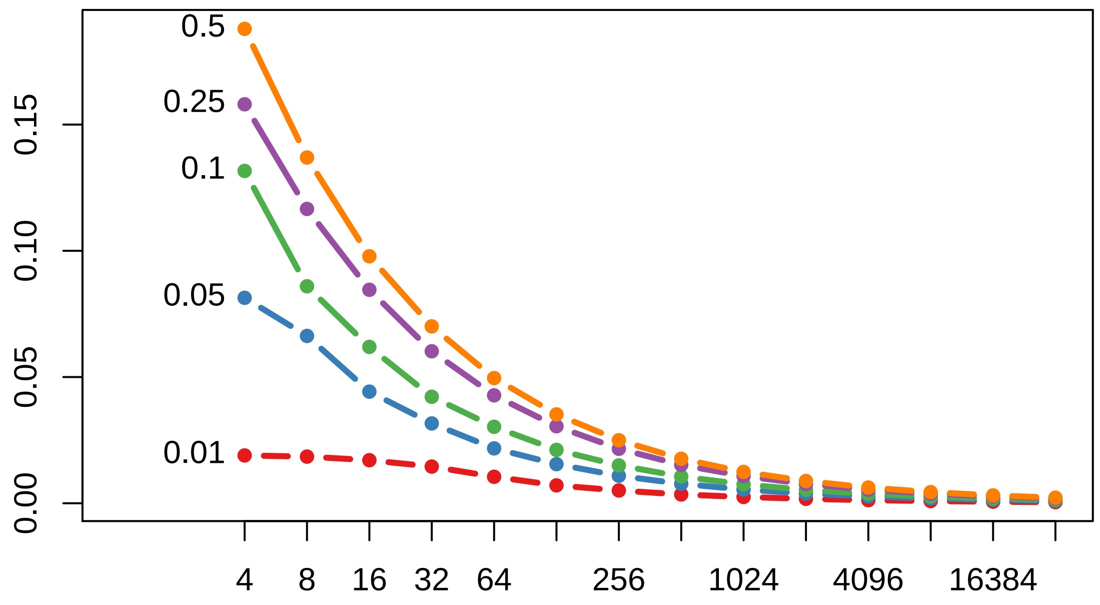
Reletive Error

This is another measure of error. The relative error is just a ratio of the absolute error of a measurment to the measurement itself. These two errors help us explore how close to reality the simulation can get based on changing parameters, in this case p (probabilty) and N (number of runs).

```r
calc_binoms <- function(N,p) {
  Y <- rbinom(100000, N, p)
  p_hat <- Y/N
  absolute_error <- abs(p_hat - p)
  relative_error <- absolute_error/p
  c(mean(absolute_error), mean(relative_error))
}
```

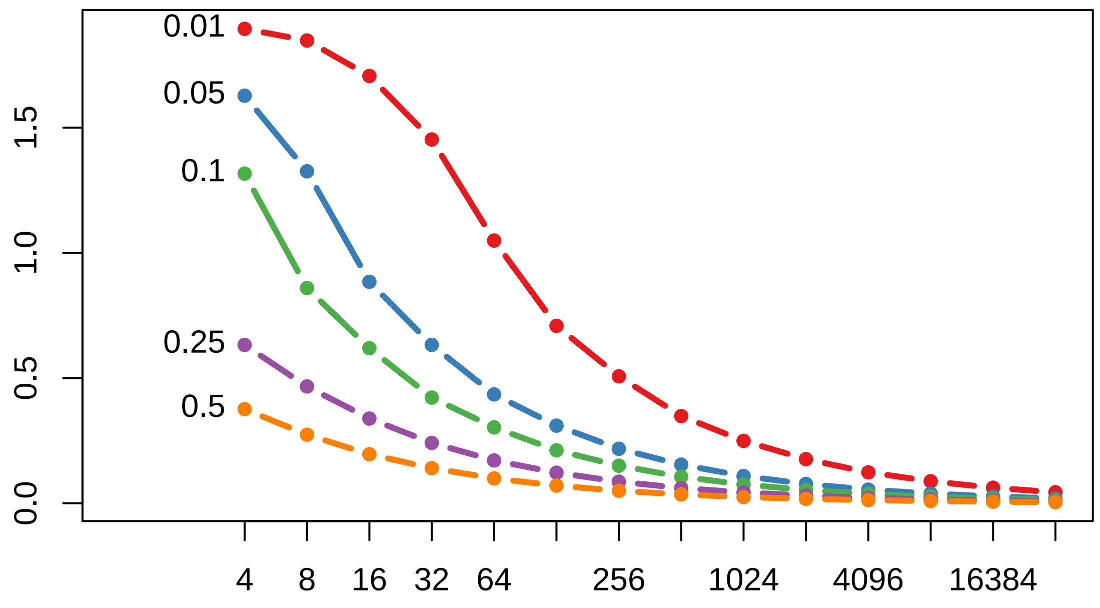
Absolute Error — Log2 Scale

```r
full_grid$xlog2 <- log2(full_grid$N)
palette(RColorBrewer::brewer.pal(5, "Set1"))
plot.new()
plot.window(xlim=c(0,15), ylim=range(full_grid$mae))
l1 <- full_grid %>% split(.$p)
for (j in seq_along(l1)){
  lines(l1[[j]]$xlog2, l1[[j]]$mae, type = "b", pch = 16, col = j, lwd = 3)
  text(2, l1[[j]]$mae[1], l1[[j]]$p[1], pos = 2)
}
axis(1, 2:15, 2^(2:15))
axis(2)
box()
```



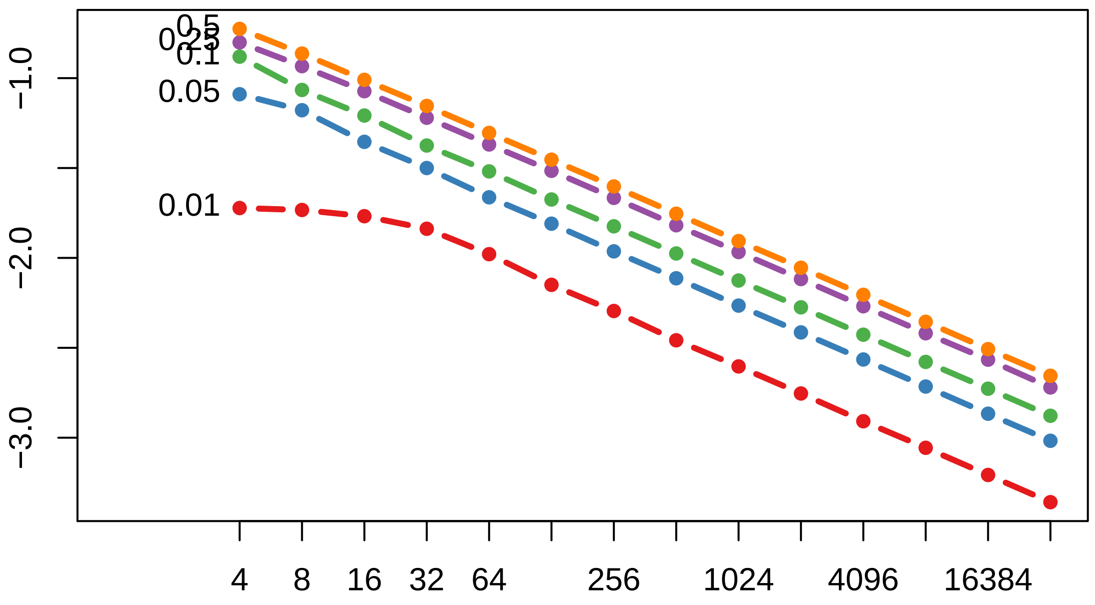
Relative Error — Log2 Scale

```
full_grid$xlog2 <- log2(full_grid$N)
palette(RColorBrewer::brewer.pal(5, "Set1"))
plot.new()
plot.window(xlim=c(0,15), ylim=range(full_grid$mre))
l1 <- full_grid %>% split(.$p)
for (j in seq_along(l1)){
  lines(l1[[j]]$xlog2, l1[[j]]$mre, type = "b", pch = 16, col = j, lwd = 3)
  text(2, l1[[j]]$mre[1], l1[[j]]$p[1], pos = 2)
}
axis(1, 2:15, 2^(2:15))
axis(2)
box()
```



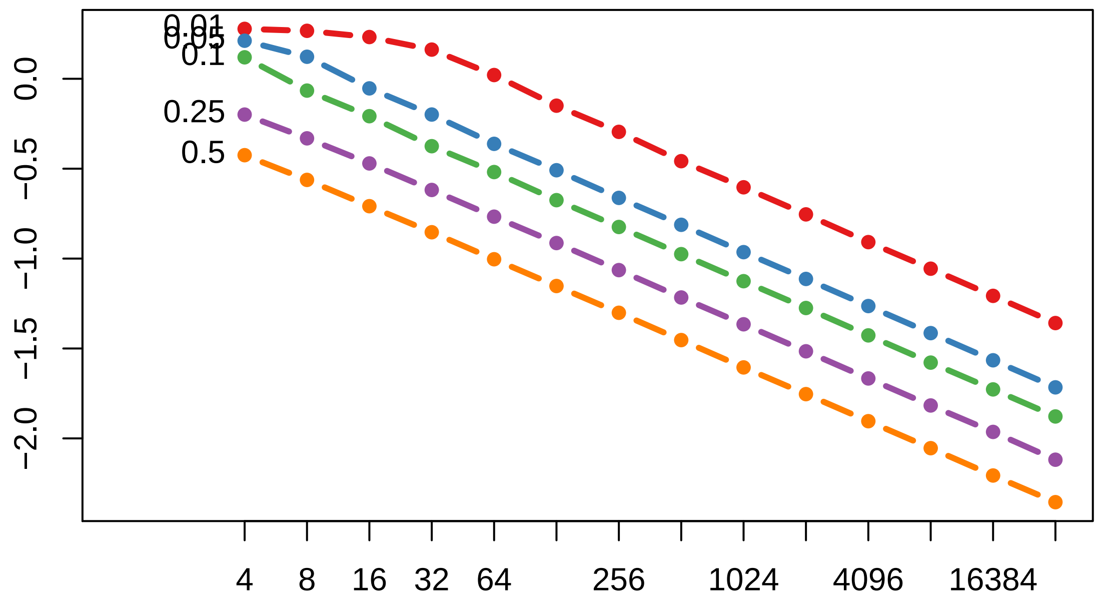
Absolute Error — Log10 Scale

```
full_grid$xlog2 <- log2(full_grid$N)
palette(RColorBrewer::brewer.pal(5, "Set1"))
plot.new()
plot.window(xlim=c(0,15), ylim=range(full_grid$log10_mae))
l1 <- full_grid %>% split(.$p)
for (j in seq_along(l1)){
  lines(l1[[j]]$xlog2, l1[[j]]$log10_mae, type = "b", pch = 16, col = j, lwd = 3)
  text(2, l1[[j]]$log10_mae[1], l1[[j]]$p[1], pos = 2)
}
axis(1, 2:15, 2^(2:15))
axis(2)
box()
```



Relative Error — Log10 Scale

```
full_grid$xlog2 <- log2(full_grid$N)
palette(RColorBrewer::brewer.pal(5, "Set1"))
plot.new()
plot.window(xlim=c(0,15), ylim=range(full_grid$log10_mre))
l1 <- full_grid %>% split(.$p)
for (j in seq_along(l1)){
  lines(l1[[j]]$xlog2, l1[[j]]$log10_mre, type = "b", pch = 16, col = j, lwd = 3)
  text(2, l1[[j]]$log10_mre[1], l1[[j]]$p[1], pos = 2)
}
axis(1, 2:15, 2^(2:15))
axis(2)
box()
```



Conclusion

Simulations in their nature will have error! This is inevitable and important to understand. We can mitigate this error through many repetitions in simulations.