Sum of Binomials

Michael LeFew

December 27, 2015

We take a set of projects, each having a certain number of dollars and a certain ratio of the area of overlap with our regioin of to the total area of the project. For each project, we calcuate a binomial distribution with parameters n = number of dollars and p = area ratio, which describes the probability of any number of dollars from 0 to the total number of dollars landing in the area of interest. (For the sake of efficiency, we won't use the full number of dollars per se, but bundles of $10,000, so for a 1 million dollar project, n = 100.) We want to describe the distribution of the sum of dollars from all the projects landing in the area of interest, which we will calculate explicitly.

For more background see this paper: https://statistics.stanford.edu/sites/default/files/SOL%20ONR%20467.pdf

and this Stack Exchange post: http://math.stackexchange.com/questions/29998/sum-of-independent-binomial-random-variables-with-different-probabilities

First we need our function that calculates the density distribution of the sum of two random variables. What it's doing is for each i in 0:dollars in project a and each j in 0:dollars in project b, multiplying p(i dollars from a lands in region of interest) x p(j dollars from b lands in area of interest), and then adding up the probability of all the situations where the sum k = i+j, i.e. for k = 5, p(k=5) = p(i=0) x p(j=5) + p(i=1) x p(j=4) + … + p(i=5) x p(j=0).

```
# explicitly calculate distribution of sum of discrete random variables
# courtesy of Michael Kuhn on Math Stack Exchange
# http://math.stackexchange.com/questions/29998/sum-of-independent-binomial-random-variables-with-different-probabilities
sumof.distributions <- function(a, b) {

  # because of the following computation, make a matrix with more columns than rows
  if (length(a) < length(b)) {
```

```
    t <- a
    a <- b
    b <- t
  }

  # explicitly multiply the probability distributions
  m <- a %*% t(b)

  # initialized the final result, element 1 = count 0
  result <- rep(0, length(a)+length(b)-1)

  # add the probabilities, always adding to the next subsequent slice
  # of the result vector
  for (i in 1:nrow(m)) {
    result[i:(ncol(m)+i-1)] <- result[i:(ncol(m)+i-1)] + m[i,]
  }

  result

}
```

Next, we'll make up some projects. The projects range in size from $100k to $1.5m, and ratio of the area of overlap ranges from 0 to 1.

```
#number of projects
projects <- 100

#number of dollars in each project
lengths <- sample(10:150,projects)

#ratio of area of interest overlap to area of project
```

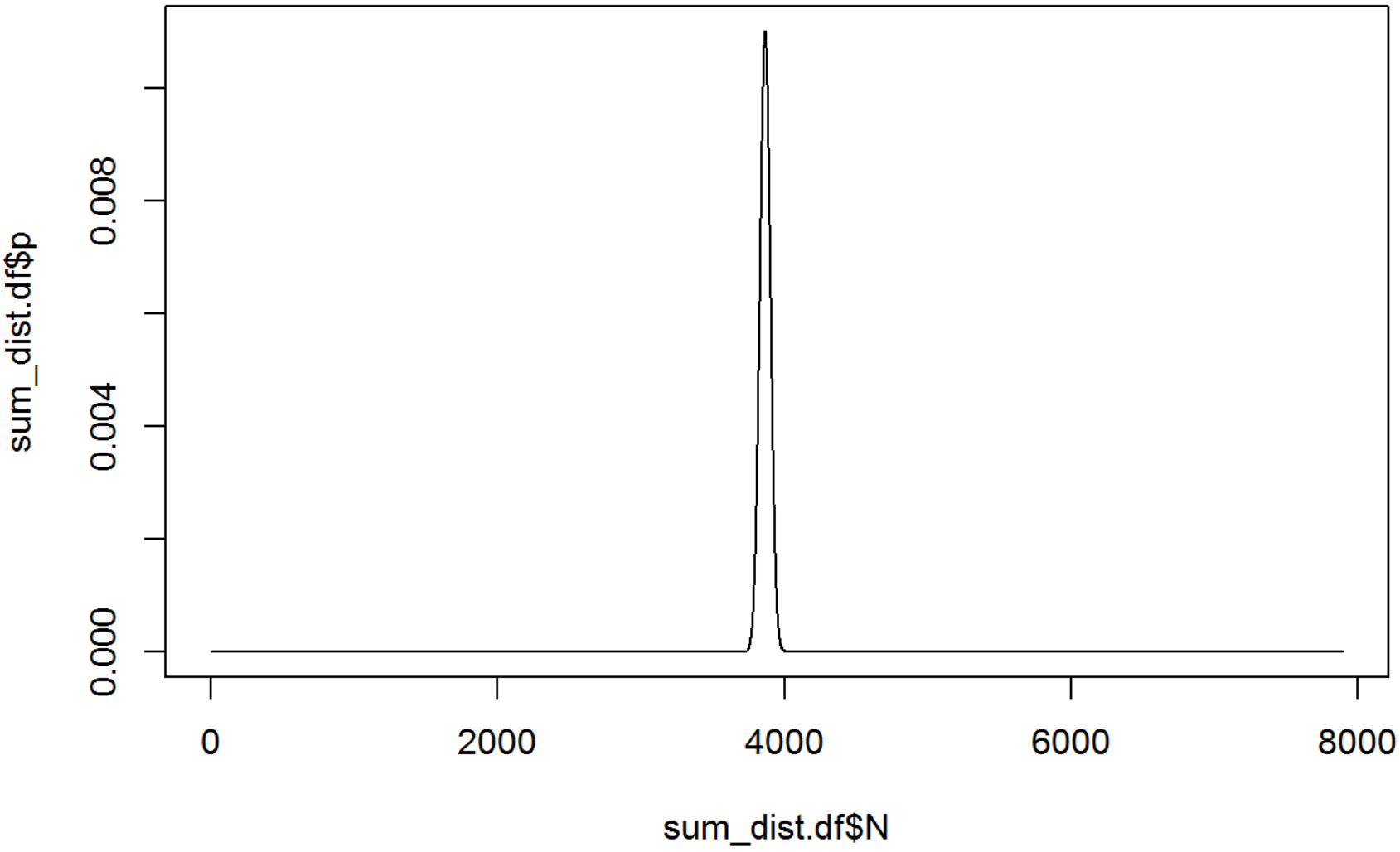
```
probs <- runif(projects,min=0,max=1)
```

Finally, add up all the distributions to find the distribution of sums. For reference, we'll keep track of the time it takes.

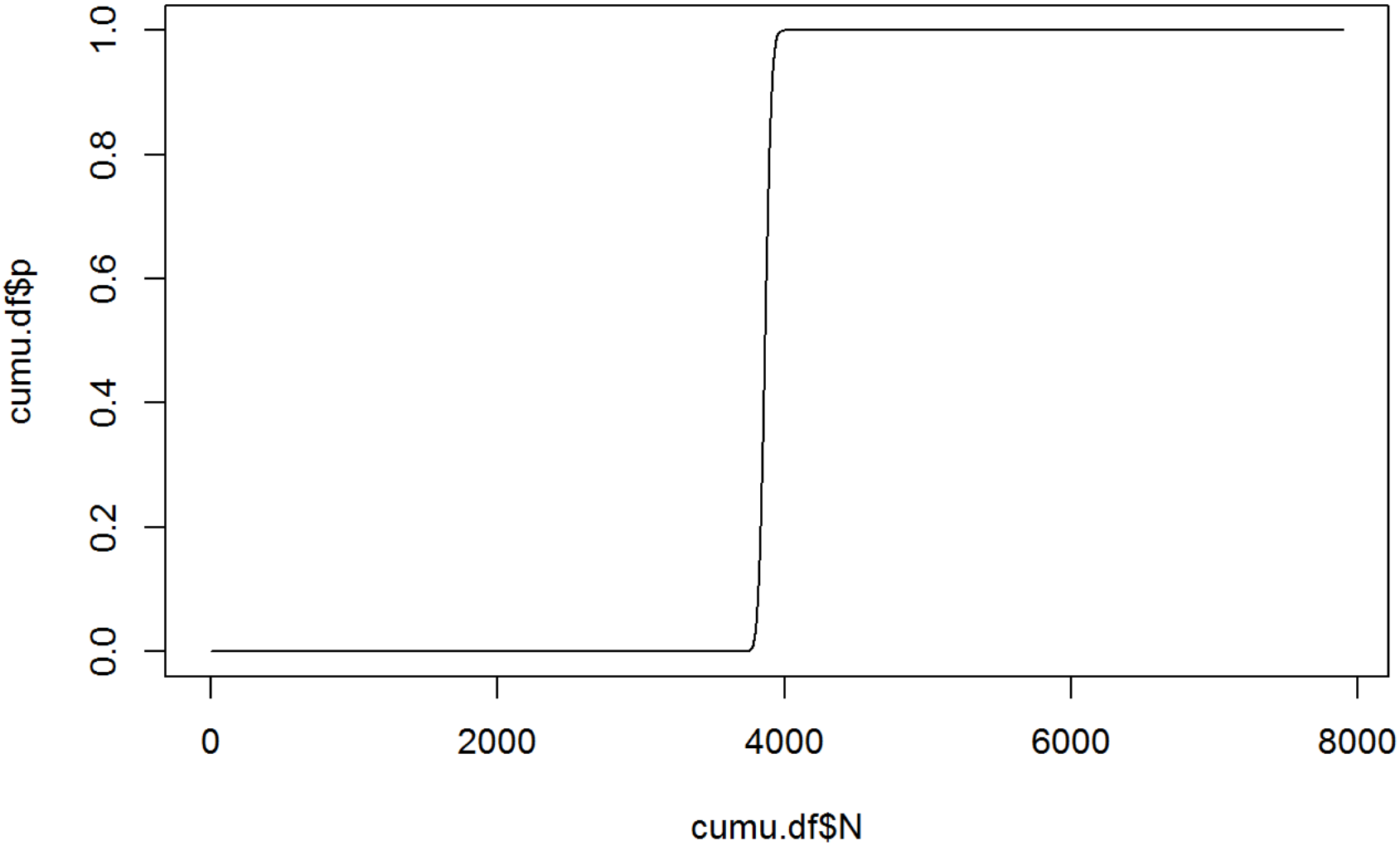
```
#start the clock!
ptm <- proc.time()

#initialize distribution of sums
#with distribution of first project
sum_dist <- dbinom(0:lengths[1],lengths[1],probs[1])

#add distribution of each project in turn
for (proj in 2:projects){
  proj_dist <- dbinom(0:lengths[proj],lengths[proj],probs[proj])
  sum_dist <- sumof.distributions(sum_dist,proj_dist)
}

#density distibution, N are sums of dollars, p are corresponding probabilities
sum_dist.df <- data.frame(N=0:(length(sum_dist)-1),p=sum_dist)

#cumulative distribution
cumu <- sum_dist
for(i in 2:length(cumu)){
  cumu[i] <- cumu[i] + cumu[i-1]
}

cumu.df <- data.frame(N=0:(length(cumu)-1),p=cumu)

#time
timetaken <- proc.time() - ptm
```

And display result. (Time is in seconds.)

```
plot(sum_dist.df$N,sum_dist.df$p,type="l")
```



```
plot(cumu.df$N,cumu.df$p,type="l")
```




```
print(timetaken)
```

```
##    user  system elapsed
##   14.06    0.18   14.95
```

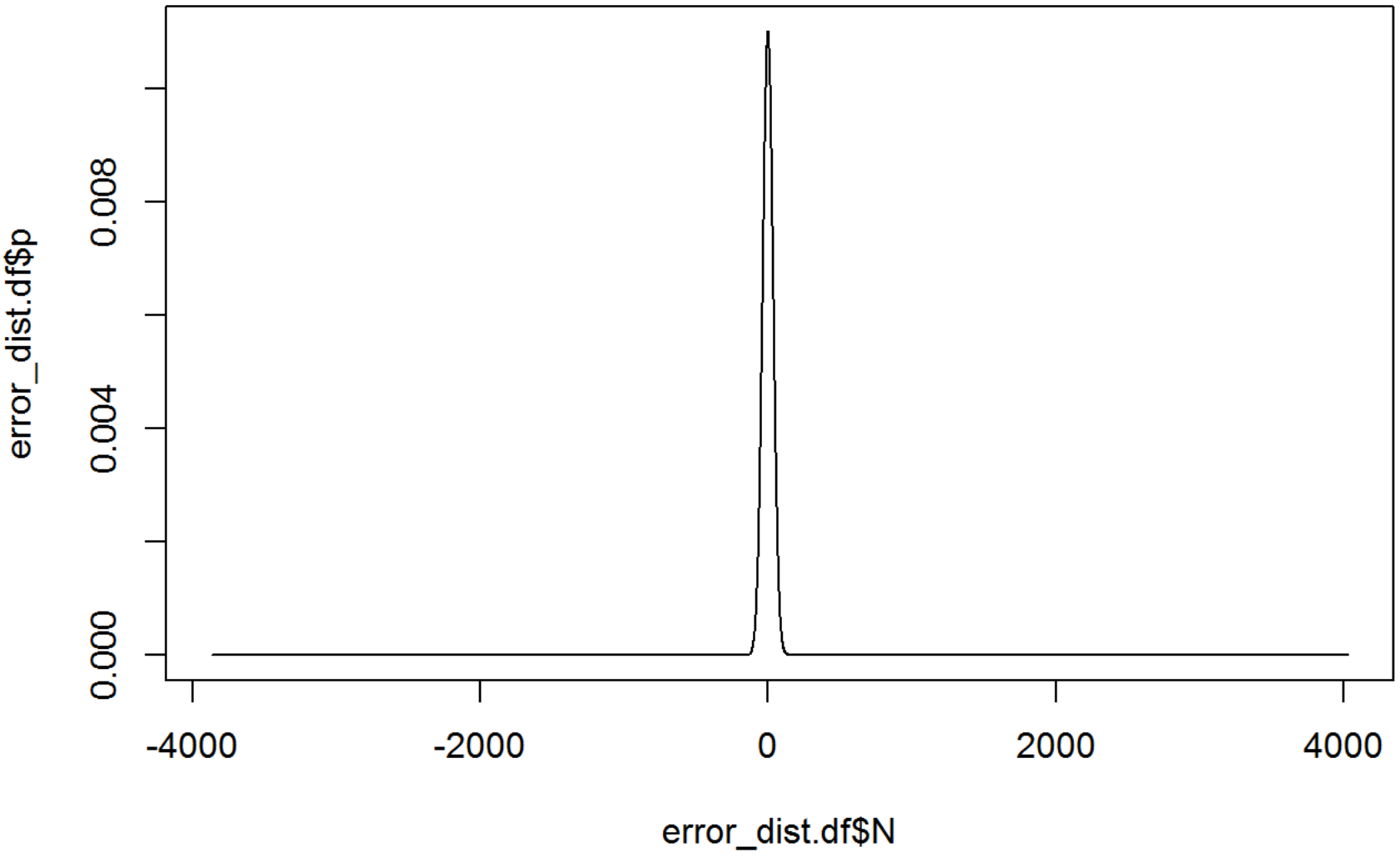
The above describes the density and cumulative distributions for the sum of dollars (sum of $10,000 bundles) landing in the area of interest from all projects. We'll use the expected value for this distribution as our assumed sum and multiply by $10,000.

```
expected_val <- sum(sum_dist.df$N * sum_dist.df$p)
dollars_in_roi <- expected_val * 10000

print(dollars_in_roi)
```

```
## [1] 38677015
```

For SIMEX we need to describe the distribution of errors. Our measured values are the expected value of the distribution of sums. We can think of the distribution of errors as the same distribution as the distribution of sums, shifted down the axis of possible values so that the expected error value is 0 and the distribution has suppport -(expected dollars):(highest possible sum - expected dollars). This is easily accomplished by subtracting expected dollars from each value in the distribution.

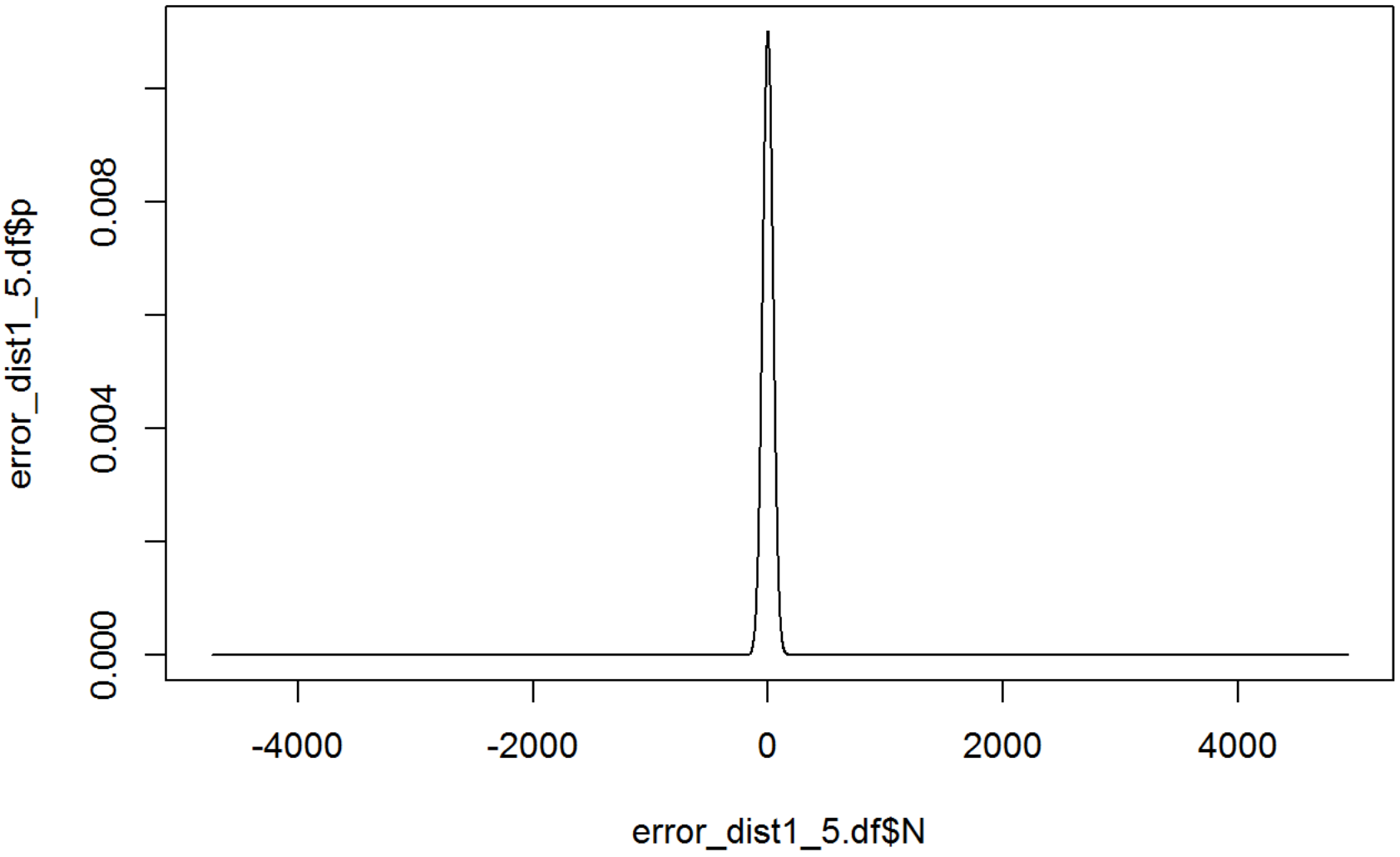
```
error_dist.df <- data.frame(N = sum_dist.df$N - expected_val, p = sum_dist.df$p)

plot(error_dist.df$N,error_dist.df$p,type="l")
```



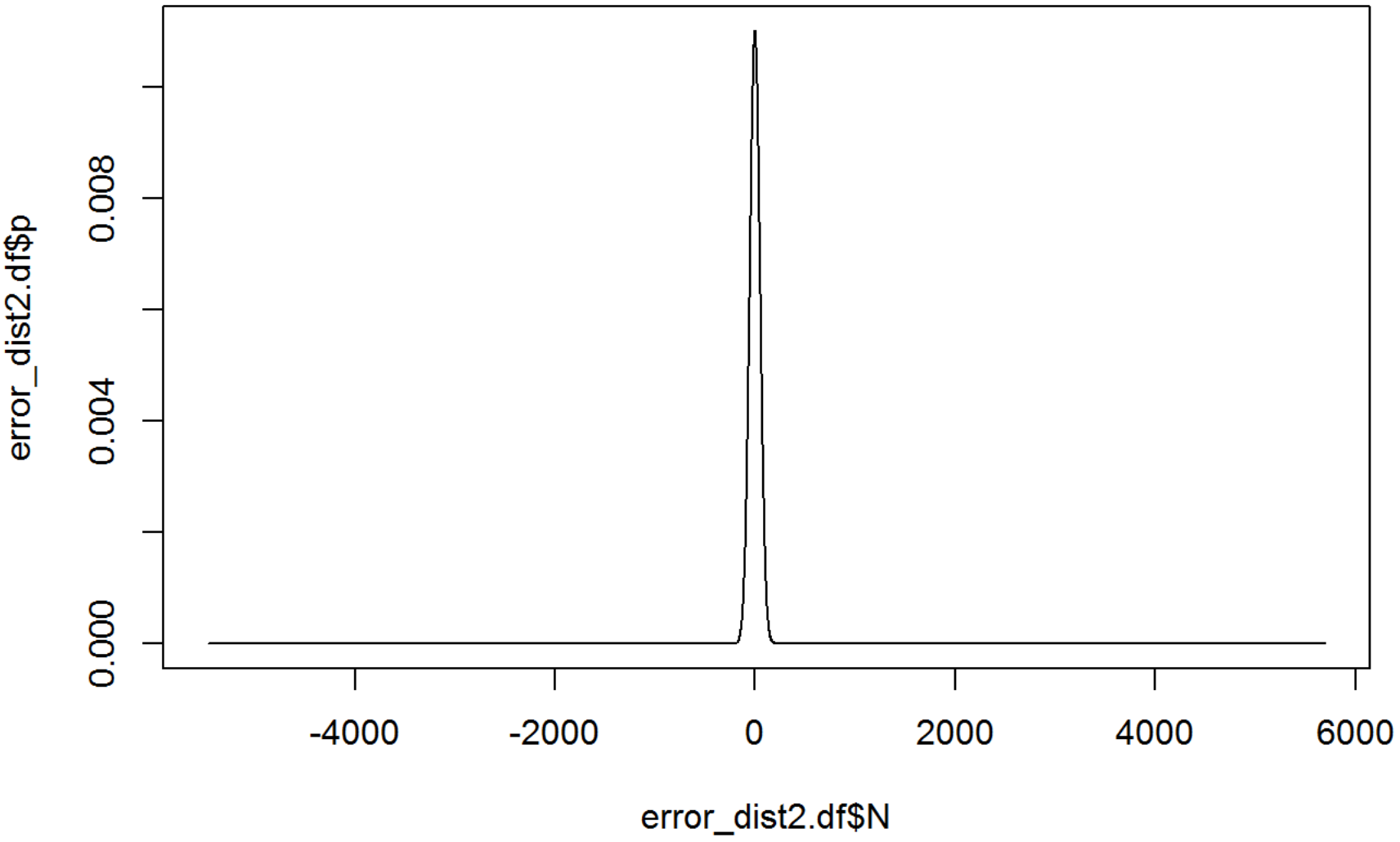
Lastly, we'll need to be able to increase the variance of the error distribution by factors lamda = {1.5,2,2.5,3} for SIMEX. Recall that the formula for variance in the discrete case is var = sum[(error-expected error)^2 x p(error)] = sum[error^2 x p(error)] in this case, since expected error is 0. To widen the distribution such that the variance increases by a factor lamda, we multipy each error value in the distribution by sqrt(lambda), such that we have altered error distribution with lambda x var

= lambda x sum[error^2 x p(error)] = sum[(sqtr(lambda)xerror)^2 x p(error)].

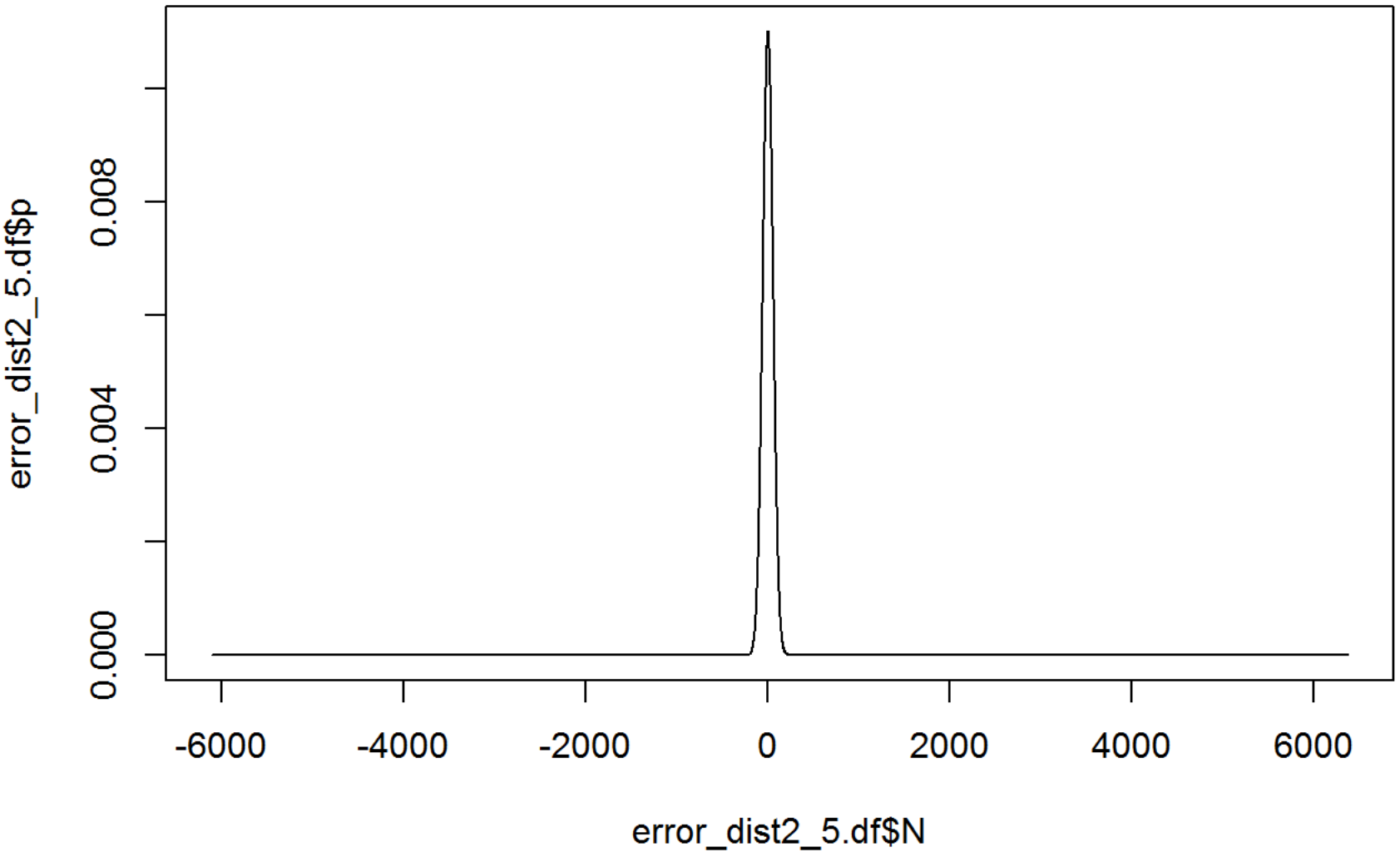
```
error_dist1_5.df <- data.frame(N=error_dist.df$N * sqrt(1.5), p=error_dist.df$p)

error_dist2.df <- data.frame(N=error_dist.df$N * sqrt(2), p=error_dist.df$p)

error_dist2_5.df <- data.frame(N=error_dist.df$N * sqrt(2.5), p=error_dist.df$p)

error_dist3.df <- data.frame(N=error_dist.df$N * sqrt(3), p=error_dist.df$p)

plot(error_dist1_5.df$N,error_dist1_5.df$p,type="l")
```



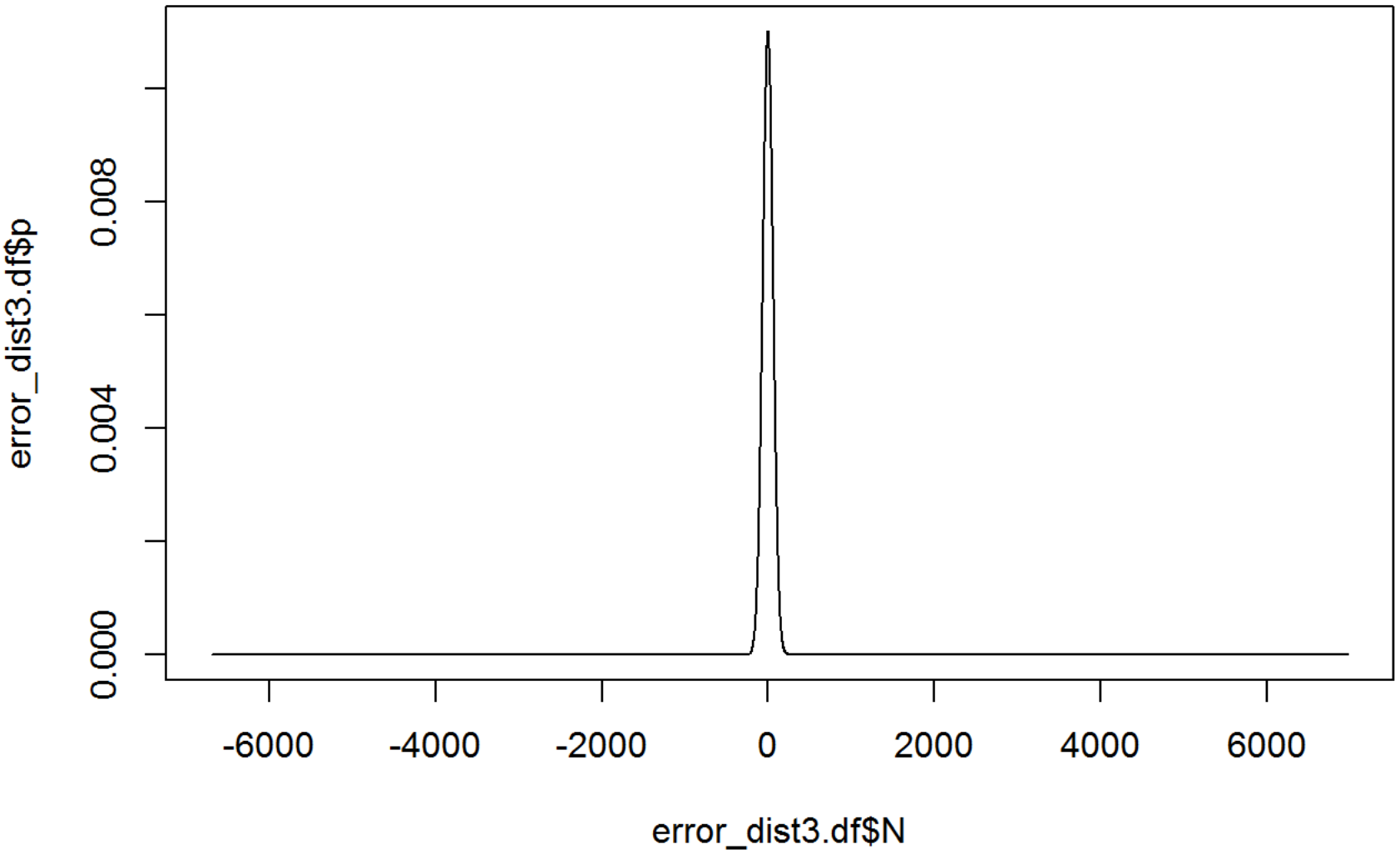

```
plot(error_dist2.df$N,error_dist2.df$p,type="l")
```




```
plot(error_dist2_5.df$N,error_dist2_5.df$p,type="l")
```




```
plot(error_dist3.df$N,error_dist3.df$p,type="l")
```



Boom.