Binomial vs Hypergeometric vs Simulation by: leta199

2025-10-06

#HOW WELL DOES BINOMIAL APPROXIMATE HYPERGEOMETRIC?——————————- #LEARNING MORE ABOUT THE FUNCTIONS PBINOM AND PHYPER

```
help(pbinom)
help(phyper)
help(sample)
```

#PART #1 - BINOMIAL AND HYPERGEOMETRIC ———————————————-

#COMPARING BINOMIAL TO HYPER GEOMETRIC AT n=100, 1000, 5000, 10000

```
n_pop<- c(100,1000,5000,10000)

#PROBABILITY COMPARISON
binom_prob<- numeric(length(n_pop)) #function that calculates bimonial probability
hyper_prob<- numeric(length(n_pop)) #function that calculates hypergeometric probability

for (i in seq_along(n_pop)) {
  n<-n_pop[i]
  red<- n*0.25
  non_red<-n*0.75
  sample<-n*0.1
  threshold<- 0.02*n

  binom_prob[i]<-pbinom(threshold, sample, 0.25)
  hyper_prob[i]<-phyper(threshold, red, non_red,sample)
}
```

#PART #2 - BY SIMULATION———————————————————-

#I want to see how our simulation compares with our functions.

```
set.seed(8909)                    #reproducibility
reps<-5000                        #number of repetitions for our internal loop with index
n_pop_2<- c(100,1000,5000,10000)  #population sizes

overall_prob<-numeric(length(n_pop_2)) #storing probability results at 4 different populations


for(j in 1:4) {
#defining counter for outer for loop
  n<-n_pop_2[j]

#partitioning our balls into a vector of red and non-red balls
```

```
  n_red<-as.integer(0.25*n)
  red_balls <-sample(1:n, n_red, replace = FALSE) #sampling 25% into red group
  colours<- rep("non red ball",n) #repeats non red ball 100 times
  colours[red_balls]<-"red ball"  #balls with index red_balls have string "red ball"

#creating a vector to store the binary outcome of  experiment to get (probability)
  colour_proportion<-numeric(reps)

  sample_size<-as.integer(0.1*n ) #sample size we select red balls from
  print(paste("Running for j =", j, "n =", n, "sample size =", sample_size))

for(i in 1:reps) {
  balls<- sample(1:n, sample_size, replace = FALSE)
  sum_of_red_balls<- sum(colours[balls]== "red ball")

#if samples less than or equla to 0.02*n red balls, success and 1
  if(sum_of_red_balls <= 0.02*n) {
    colour_proportion[i]<-1  #assigning success or failure to colour_proportion vector
  } else {
    colour_proportion[i]<-0
  }
}
  overall_prob[j]<-mean(colour_proportion) }
```

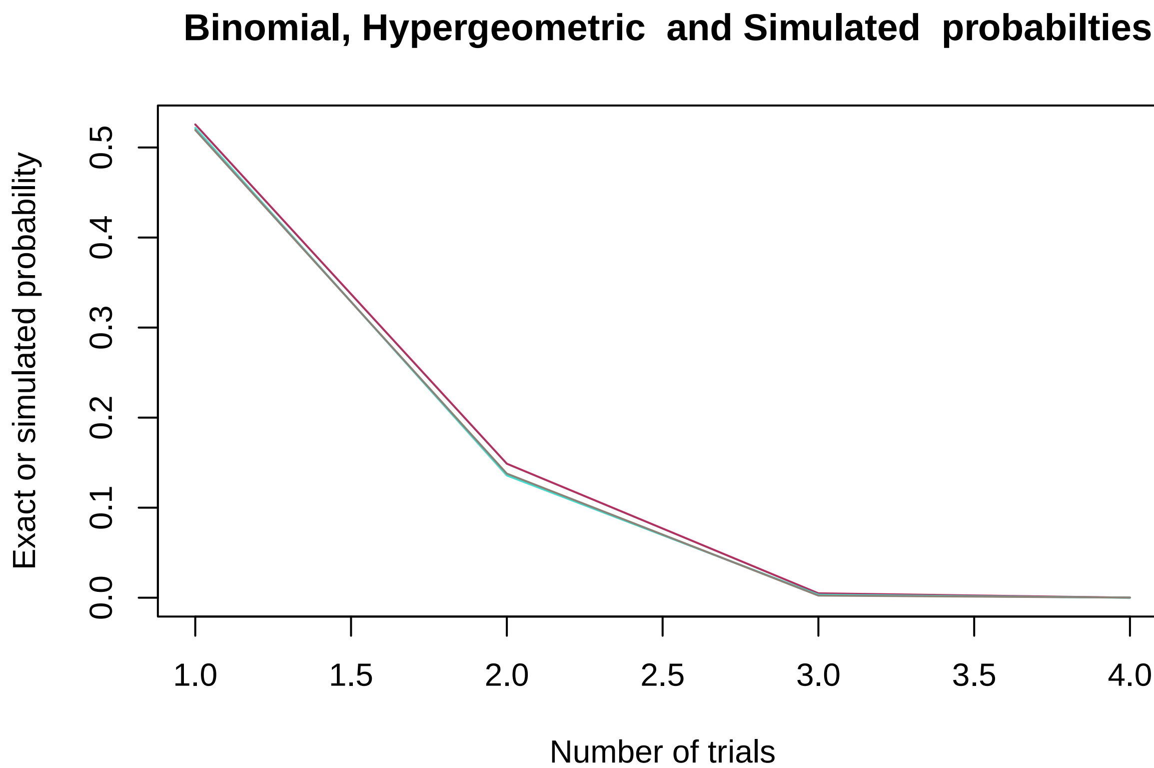
```
## [1] "Running for j = 1 n = 100 sample size = 10"
## [1] "Running for j = 2 n = 1000 sample size = 100"
## [1] "Running for j = 3 n = 5000 sample size = 500"
## [1] "Running for j = 4 n = 10000 sample size = 1000"
```

#VISUALISATIONS———————————————————

#We will create a data frame to display the numerical value of all three methods and their probabilities

```
results<- data.frame(
  n= n_pop,
  Bimonial_Probaility = round(binom_prob,6),
  Hypergeomtric_Probability = round(hyper_prob,6),
  Simulated_Probability= round(overall_prob,6)
)
```

#Let us also create a combined line graph to show how close the porbabilities are to each other at each trial



#CONCULSION——————————————————————

#Of course as our population increases the probability of having less than or equal to 0.02*n decreases. #Interestingly, our simulation at 5000 repetitions models our phyper() function very closely as all population sizes. #As our population size increases (N»n) our binomial probability apporximates our hypergeometric probability very well.