Lab5-R

Clayton Glenn

February 16, 2018

Table of Contents

# Task1

## Get working directory

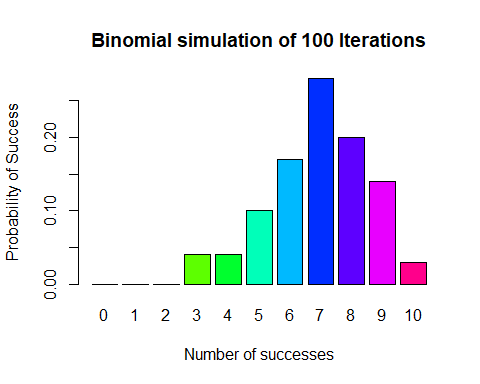
getwd()

## [1] "C:/Users/cglen/Documents/Stat Methods/Labs/LAB5"

# Task2

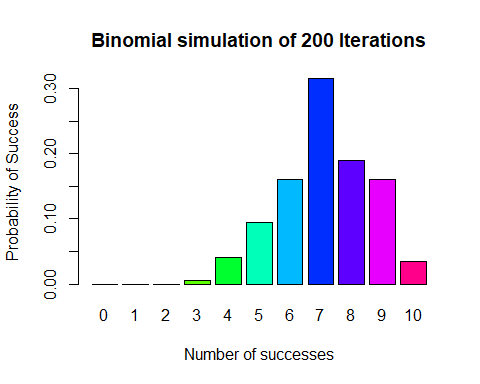
## Show probability

mybin = function(iter = 100, n = 10, p = 0.7){   
 #make a matrix to hold the samples  
 #initially filled with NA's  
 sam.mat = matrix(NA, nr = n, nc = iter, byrow = TRUE)  
 #Make a vector to hold the number of successes in each trial  
 succ = c()  
 for( i in 1:iter){  
 #Fill each column with a new sample  
 sam.mat[, i] = sample(c(1, 0), n, replace = TRUE, prob = c(p, 1-p))  
 #Calculate a statistic from the sample (this case it is the sum)  
 succ[i] = sum(sam.mat[, i])  
 }  
 #Make a table of successes  
 succ.tab = table(factor(succ, levels = 0:n))  
 #Make a barplot of the proportions  
 barplot(succ.tab / (iter), col = rainbow(n+1),   
 main = sprintf("Binomial simulation of %d Iterations", iter),   
 xlab = "Number of successes", ylab = "Probability of Success")  
 succ.tab / iter  
}  
mybin(iter = 100, n = 10, p = 0.7)



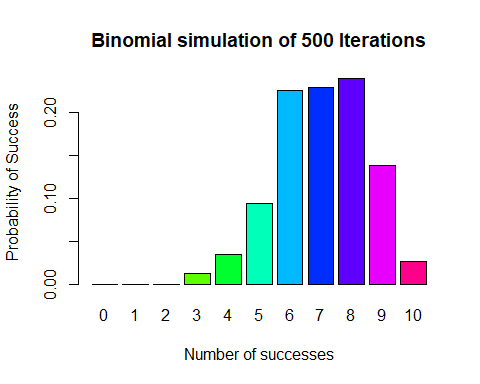
##   
## 0 1 2 3 4 5 6 7 8 9 10   
## 0.00 0.00 0.00 0.04 0.04 0.10 0.17 0.28 0.20 0.14 0.03

mybin(iter = 200, n = 10, p = 0.7)



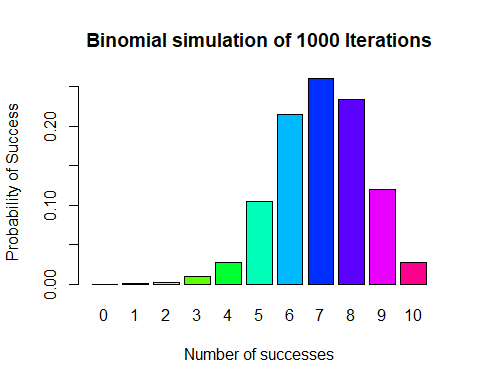
##   
## 0 1 2 3 4 5 6 7 8 9 10   
## 0.000 0.000 0.000 0.005 0.040 0.095 0.160 0.315 0.190 0.160 0.035

mybin(iter = 500, n = 10, p = 0.7)



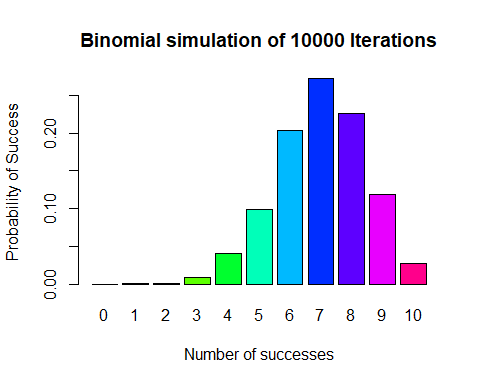
##   
## 0 1 2 3 4 5 6 7 8 9 10   
## 0.000 0.000 0.000 0.012 0.034 0.094 0.226 0.230 0.240 0.138 0.026

mybin(iter = 1000, n = 10, p = 0.7)



##   
## 0 1 2 3 4 5 6 7 8 9 10   
## 0.000 0.001 0.002 0.010 0.027 0.104 0.215 0.260 0.234 0.120 0.027

mybin(iter = 10000, n = 10, p = 0.7)



##   
## 0 1 2 3 4 5 6 7 8 9   
## 0.0000 0.0004 0.0013 0.0092 0.0411 0.0992 0.2034 0.2724 0.2266 0.1188   
## 10   
## 0.0276

## Check binomial plots for accuracy

dbinom(x = 0:10, size = 10, prob = 0.7)

## [1] 0.0000059049 0.0001377810 0.0014467005 0.0090016920 0.0367569090  
## [6] 0.1029193452 0.2001209490 0.2668279320 0.2334744405 0.1210608210  
## [11] 0.0282475249

The plots correspond to the binomial function that 7 is the peak, and rest of graph tends down.

# Task3

## Use sample to create a 12:8 marble scenario

sample(rep(c(1,0),c(8,12)),5,replace=FALSE)

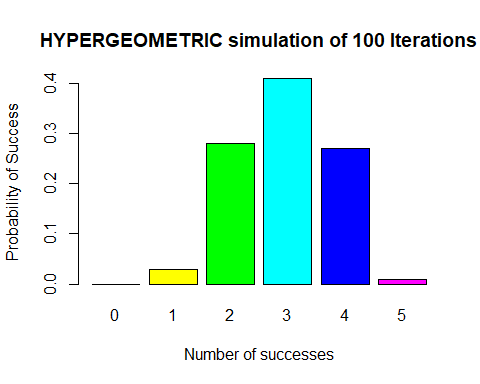
## [1] 0 0 0 0 1

sample(rep(c(1,0),c(8,12)),5,replace=TRUE)

## [1] 0 0 1 1 1

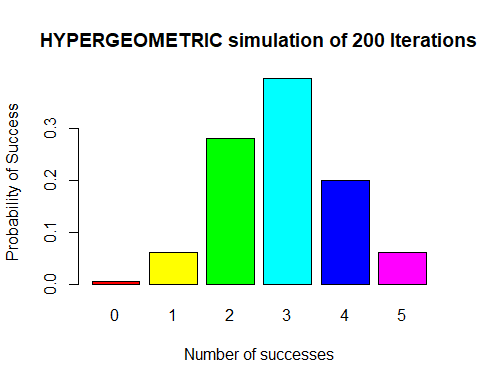
## Hypergeometric function

myhyper=function(iter=100,N=20,r=12,n=5){  
 # make a matrix to hold the samples  
 #initially filled with NA's  
 sam.mat=matrix(NA,nr=n,nc=iter, byrow=TRUE)  
 #Make a vector to hold the number of successes over the trials  
 succ=c()  
 for( i in 1:iter){  
 #Fill each column with a new sample  
 sam.mat[,i]=sample(rep(c(1,0),c(r,N-r)),n,replace=FALSE)  
 #Calculate a statistic from the sample (this case it is the sum)  
 succ[i]=sum(sam.mat[,i])  
 }  
 #Make a table of successes  
 succ.tab=table(factor(succ,levels=0:n))  
 #Make a barplot of the proportions  
 barplot(succ.tab/(iter), col=rainbow(n+1),   
 main=sprintf("HYPERGEOMETRIC simulation of %d Iterations", iter),   
 xlab="Number of successes", ylab = "Probability of Success")  
 succ.tab/iter  
}  
myhyper(iter=100,n=5, N=20,r=12)



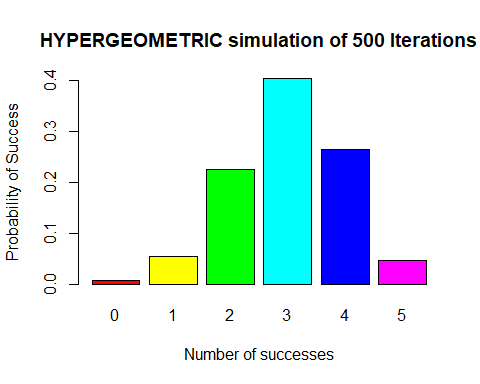
##   
## 0 1 2 3 4 5   
## 0.00 0.03 0.28 0.41 0.27 0.01

myhyper(iter=200,n=5, N=20,r=12)



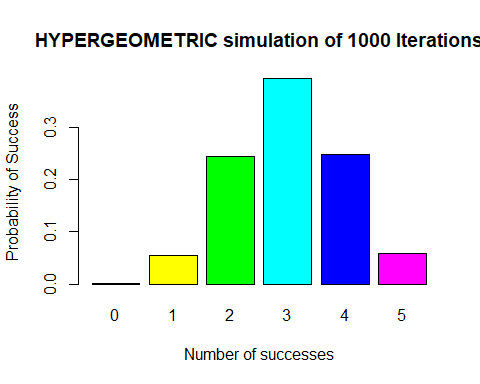
##   
## 0 1 2 3 4 5   
## 0.005 0.060 0.280 0.395 0.200 0.060

myhyper(iter=500,n=5, N=20,r=12)



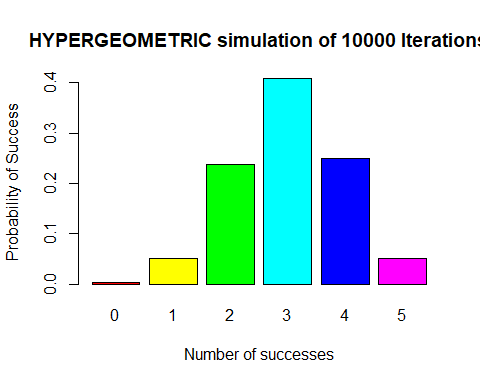
##   
## 0 1 2 3 4 5   
## 0.006 0.054 0.226 0.404 0.264 0.046

myhyper(iter=1000,n=5, N=20,r=12)



##   
## 0 1 2 3 4 5   
## 0.001 0.054 0.244 0.394 0.248 0.059

myhyper(iter=10000,n=5, N=20,r=12)



##   
## 0 1 2 3 4 5   
## 0.0037 0.0511 0.2370 0.4082 0.2501 0.0499

## Check the HyperGeom Plots

dhyper(x=0:5, m=12, n=8, k=5)

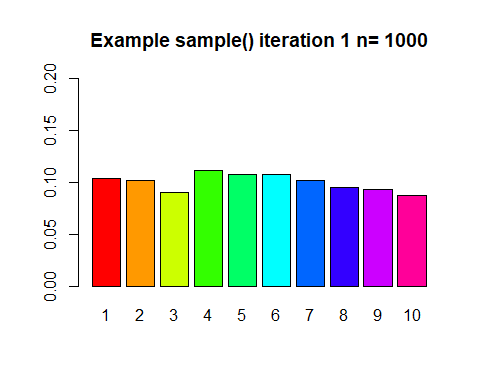
## [1] 0.003611971 0.054179567 0.238390093 0.397316821 0.255417957 0.051083591

The Hypergeometric plot does follow the path of the Hypergeometric function for n 0:5 for the 10000 iterations.

# Task 4

## Show 30 iterations of plots with 1 sec time lag

mysample=function(n, iter=10,time=0.5){  
 for( i in 1:iter){  
 #make a sample  
 s=sample(1:10,n,replace=TRUE)  
 # turn the sample into a factor  
 sf=factor(s,levels=1:10)  
 #make a barplot  
 barplot(table(sf)/n,beside=TRUE,col=rainbow(10),   
 main=paste("Example sample()", " iteration ", i, " n= ", n,sep="") ,  
 ylim=c(0,0.2)  
 )  
 #release the table  
 Sys.sleep(time)  
 }  
}  
mysample(n=1000, iter=1, time=1)



# Task5

## 8 choose 4

choose(8,4)

## [1] 70

## Poisson function

ppois(4, lambda=2)

## [1] 0.947347

## neg binom function

dnbinom(10,3,0.4)

## [1] 0.02554091

## sum of vectors 0:8 of binom function

sum(dbinom(0:8,15,0.4))

## [1] 0.9049526

# Task6

## Negative Binomial Function without any generic function calls

mynbin=function(y,r,p){  
 num=1  
 denum1=1  
 denum2=1  
 pr = 1  
 tail = 1  
 for(i in 1:(y-1)){  
 num = num \* i  
 }  
 for(i in 1:(r-1)){  
 denum1 = denum1 \* i  
 }  
 for(i in 1:((y-1)-(r-1))){  
 denum2 = denum2 \* i  
 }  
 for(i in 1:r){  
 pr = pr \* p  
 }  
 for(i in 1:(y-r)){  
 tail = tail \* (1-p)  
 }  
 num/denum1/denum2\*pr\*tail  
}  
mynbin(10,3,0.4)

## [1] 0.06449725