Lab5-R

Clayton Glenn

February 16, 2018

Task1

Get working directory

```
getwd()
```

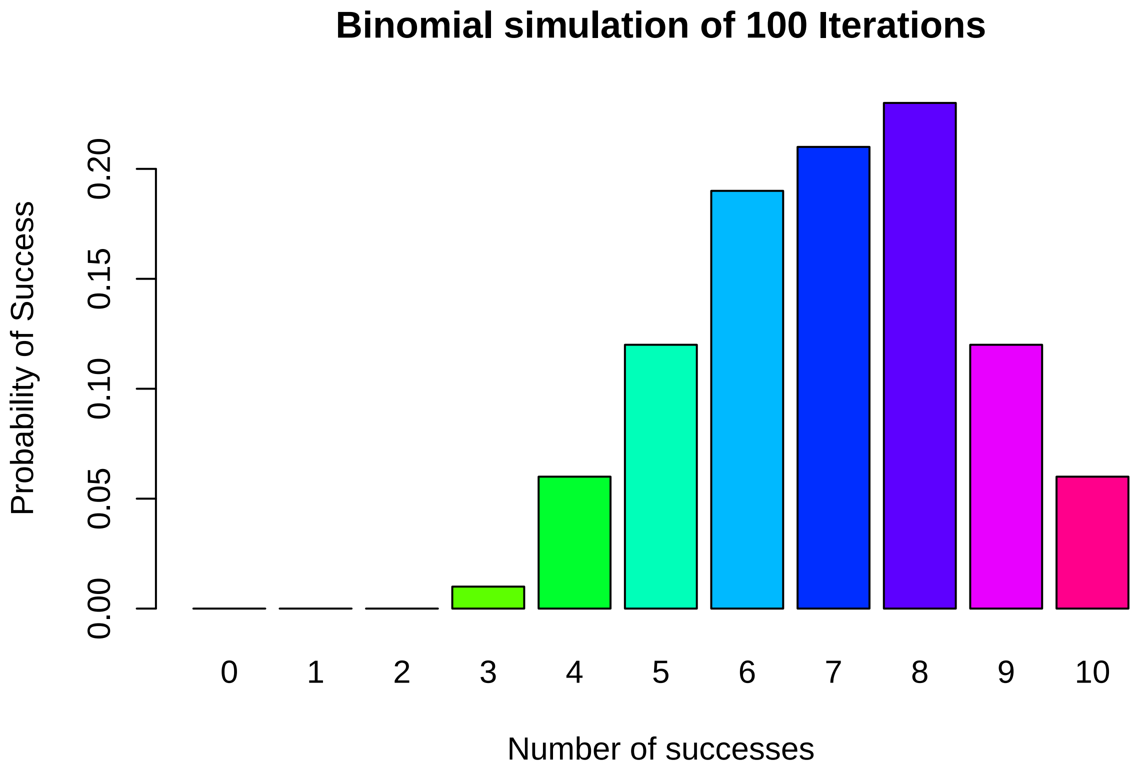
```
## [1] "C:/Users/cglen/Documents/Stat Methods/Labs/LAB5"
```

Task2

Show probability

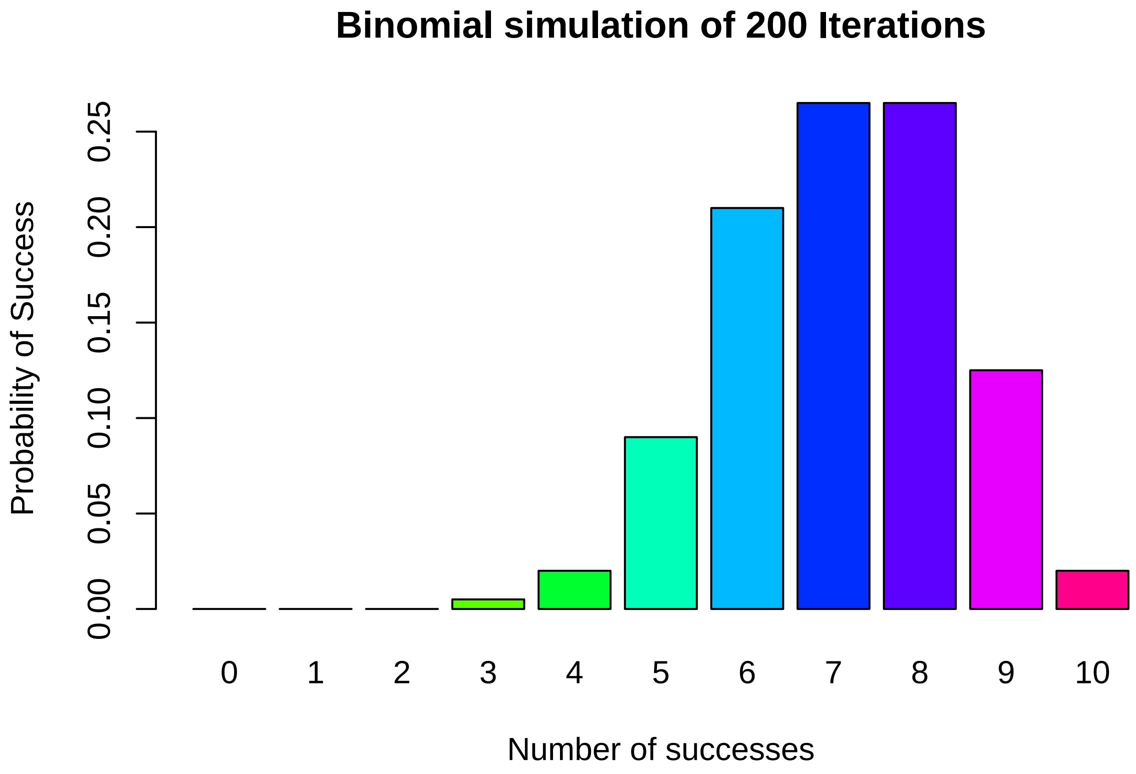
$$P(X = x) = \binom{n}{r} p^r q^{n-r}$$

```
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)
```



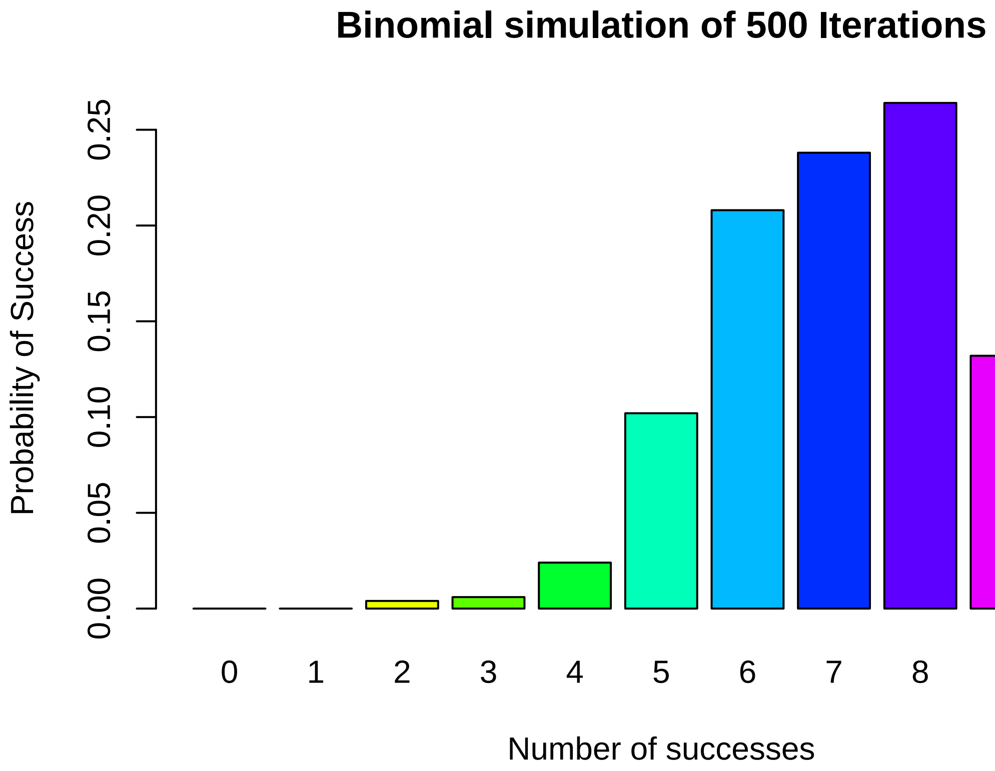
```
##
##    0    1    2    3    4    5    6    7    8    9   10
## 0.00 0.00 0.00 0.01 0.06 0.12 0.19 0.21 0.23 0.12 0.06
```

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



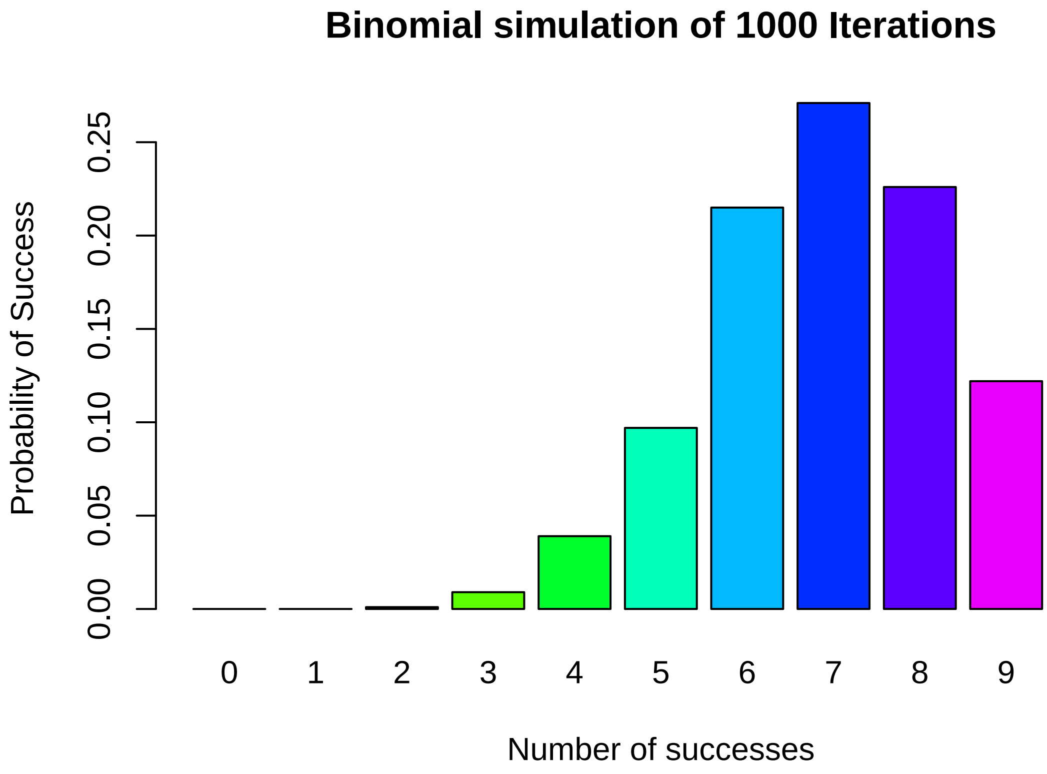
```
##
##     0     1     2     3     4     5     6     7     8     9    10
## 0.000 0.000 0.000 0.005 0.020 0.090 0.210 0.265 0.265 0.125 0.020
```

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



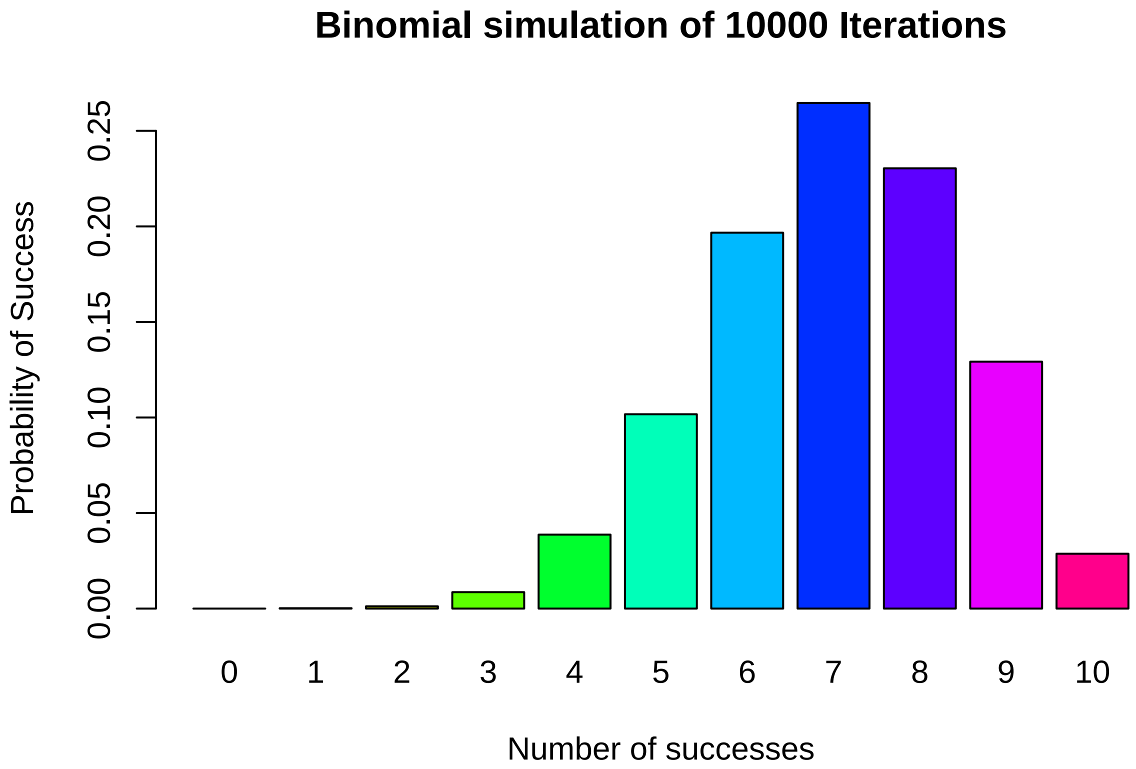
```
##
##     0     1     2     3     4     5     6     7     8     9    10
## 0.000 0.000 0.004 0.006 0.024 0.102 0.208 0.238 0.264 0.132 0.022
```

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



```
##
##     0     1     2     3     4     5     6     7     8     9    10
## 0.000 0.000 0.001 0.009 0.039 0.097 0.215 0.271 0.226 0.122 0.020
```

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



```
##
##      0      1      2      3      4      5      6      7      8      9
## 0.0000 0.0002 0.0012 0.0086 0.0387 0.1017 0.1967 0.2646 0.2304 0.1292
##     10
## 0.0287
```

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 1 1 1 0
```

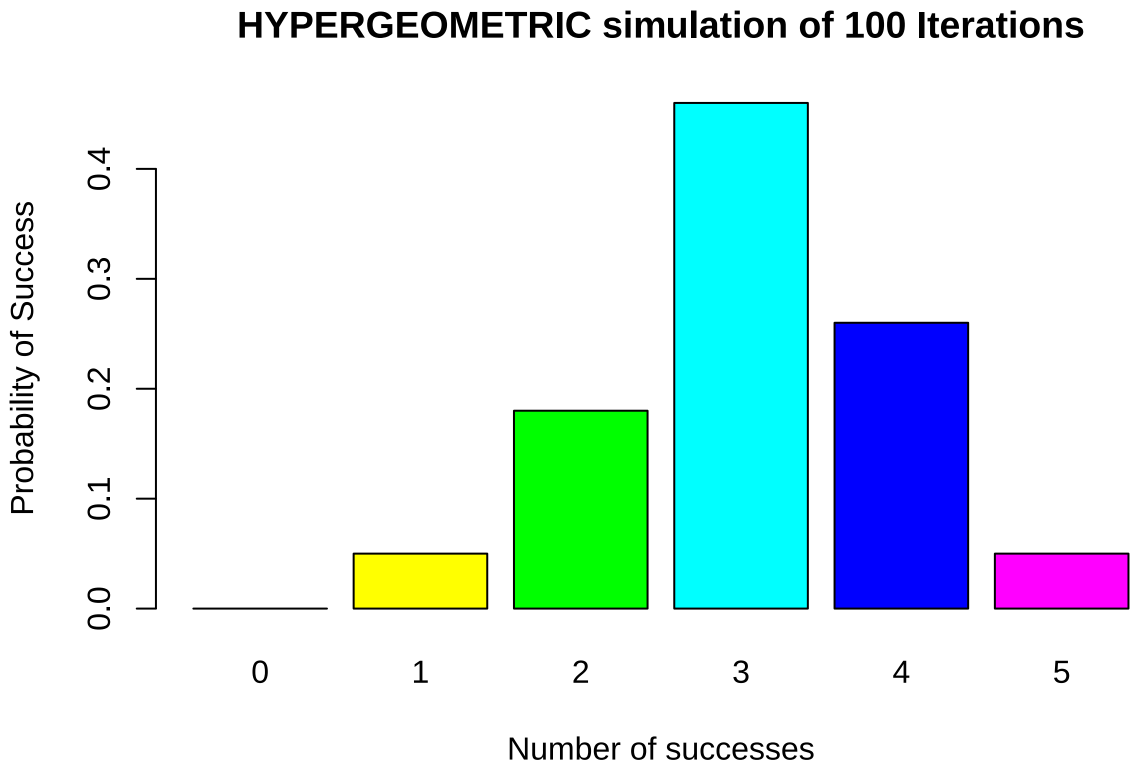
```
sample(rep(c(1,0),c(8,12)),5,replace=TRUE)
```

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

Hypergeometric function

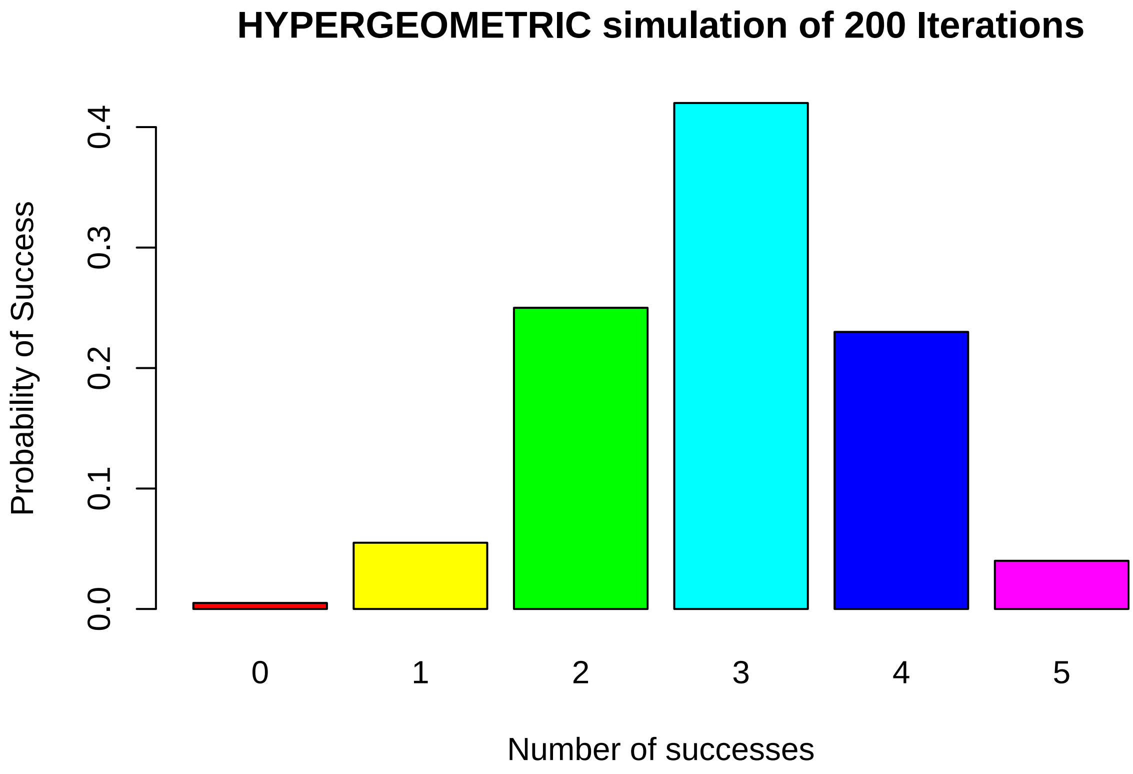
$$P(X=k)=\frac{\binom{m}{k}\binom{N-m}{n-k}}{\binom{N}{n}}$$

```
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)
```



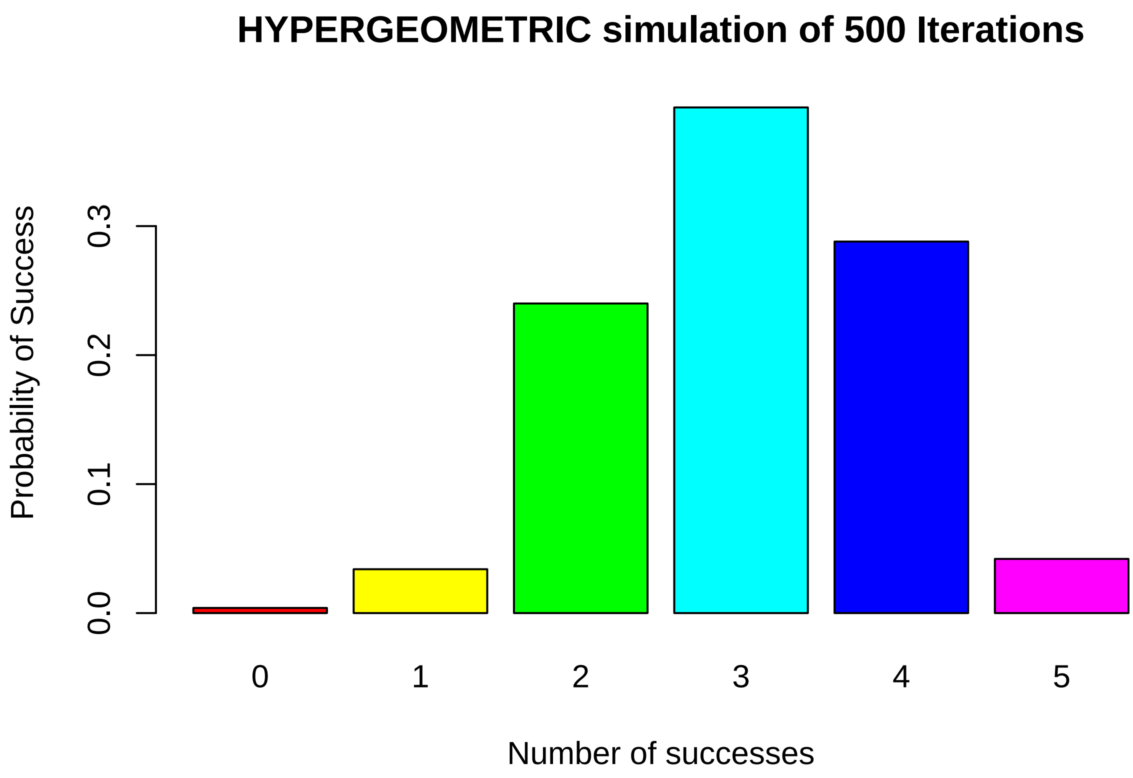
```
##
##    0    1    2    3    4    5
## 0.00 0.05 0.18 0.46 0.26 0.05
```

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



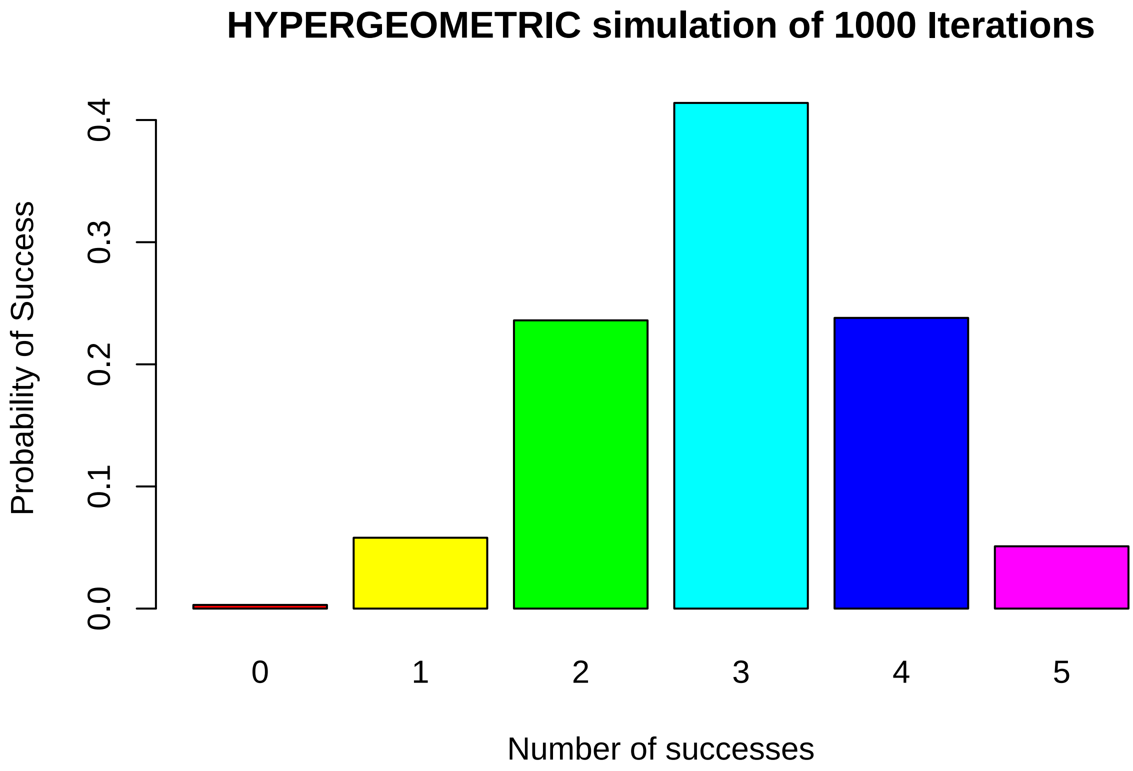
```
##
##     0     1     2     3     4     5
## 0.005 0.055 0.250 0.420 0.230 0.040
```

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



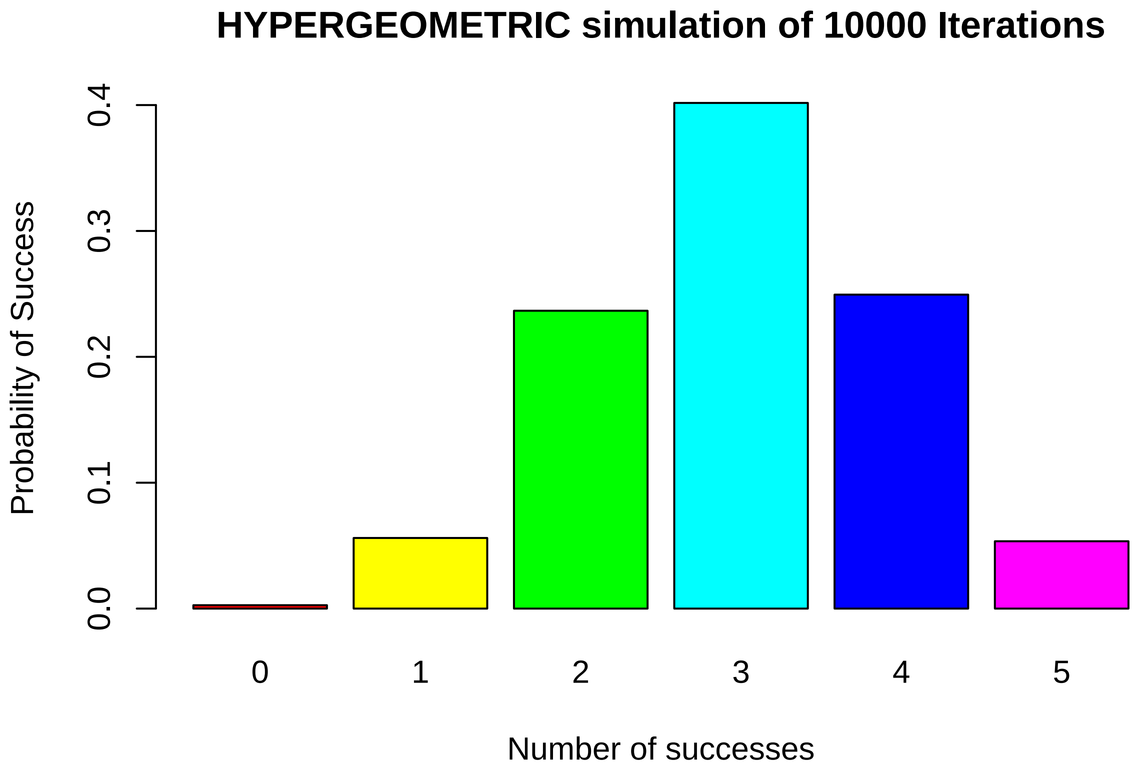
```
##
##     0     1     2     3     4     5
## 0.004 0.034 0.240 0.392 0.288 0.042
```

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



```
##
##     0     1     2     3     4     5
## 0.003 0.058 0.236 0.414 0.238 0.051
```

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



```
##
##      0      1      2      3      4      5
## 0.0027 0.0561 0.2366 0.4017 0.2494 0.0535
```

Check the HyperGeom Plots

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

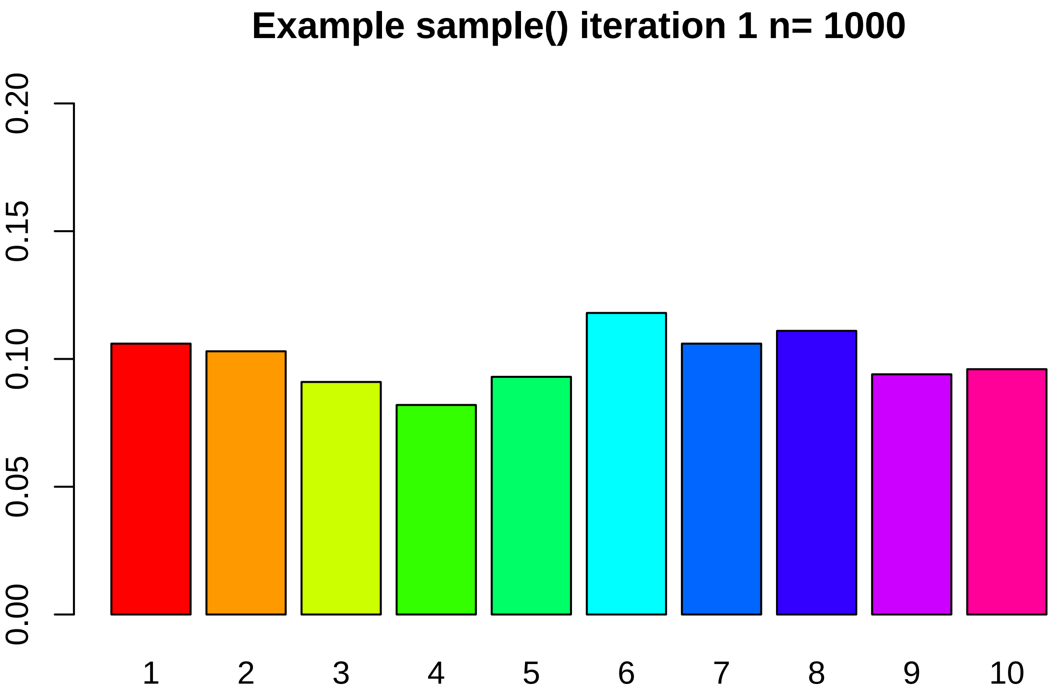
```
## [1] 0.003611971 0.054179567 0.238390093 0.397316821 0.255417957 0.051083591
```

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

$$\binom{n}{r} = \frac{n!}{r!\,(n-r)!}$$

```
choose(8,4)
```

```
## [1] 70
```

Poisson function

$$P\left(X=x\right)=\frac{\lambda^{x}e^{-\lambda}}{x!}$$

```
ppois(4, lambda=2)
```

```
## [1] 0.947347
```

neg binom function

$$P(y-r)=\binom{y-1}{r-1}\cdot p^{r}\cdot(1-p)^{y-r}$$

```
dnbinom(10,3,0.4)
```

```
## [1] 0.02554091
```

sum of vectors 0:8 of binom function

$$P(X\leq 8)=\sum_{x=0}^{8}\binom{n}{r}p^{r}q^{n-r}$$

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

```
## [1] 0.9049526
```

Task6

Negative Binomial Function without any generic function calls

$$P(y-r)=\frac{(y-1)!\cdot p^{r}\cdot(1-p)^{y-r}}{(r-1)!(y-r)!}=\frac{num\cdot pr\cdot tail}{denum1\cdot denum2}$$

```
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
  }
```

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

```
## [1] 0.06449725
```