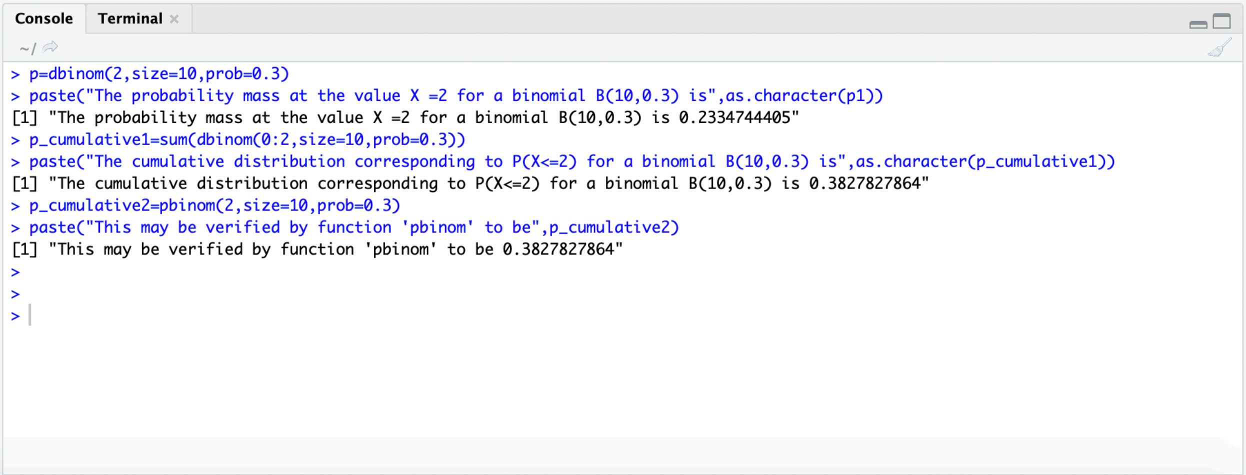
1.2



```
> p=dbinom(2,size=10,prob=0.3)
> paste("The probability mass at the value X =2 for a binomial B(10,0.3) is",as.character(p1))
[1] "The probability mass at the value X =2 for a binomial B(10,0.3) is 0.2334744405"
> p_cumulative1=sum(dbinom(0:2,size=10,prob=0.3))
> paste("The cumulative distribution corresponding to P(X<=2) for a binomial B(10,0.3) is",as.character(p_cumulative1))
[1] "The cumulative distribution corresponding to P(X<=2) for a binomial B(10,0.3) is 0.3827827864"
> p_cumulative2=pbinom(2,size=10,prob=0.3)
> paste("This may be verified by function 'pbinom' to be",p_cumulative2)
[1] "This may be verified by function 'pbinom' to be 0.3827827864"
>
>
> 
```

R code:

```
p=dbinom(2,size=10,prob=0.3)
paste("The probability mass at the value X =2 for a binomial B(10,0.3) is",as.character(p1))
p_cumulative1=sum(dbinom(0:2,size=10,prob=0.3))
paste("The cumulative distribution corresponding to P(X<=2) for a binomial B(10,0.3) is",as.character(p_cumulative1))
p_cumulative2=pbinom(2,size=10,prob=0.3)
paste("This may be verified by function 'pbinom' to be",p_cumulative2)
```

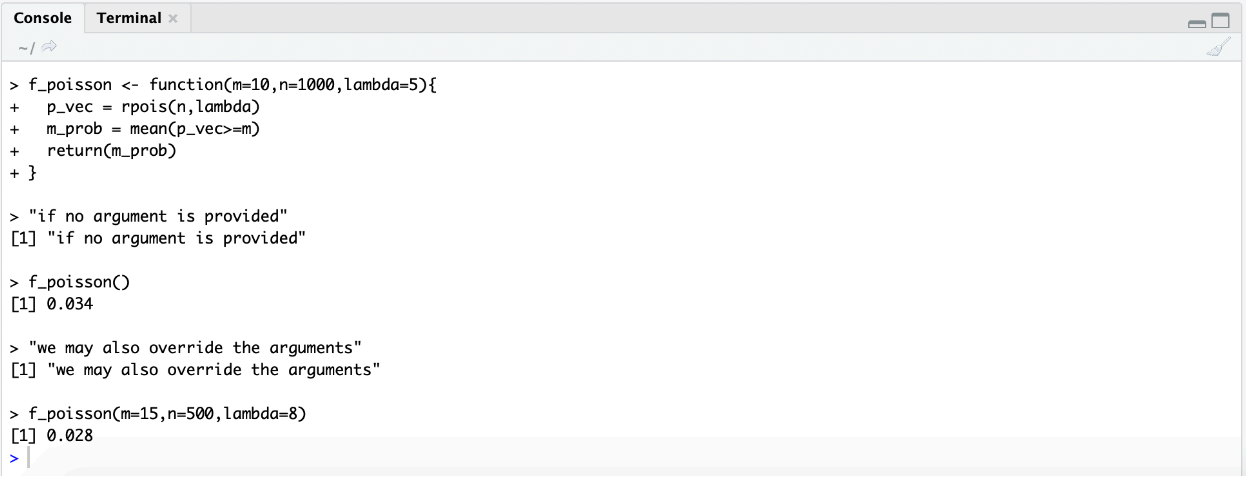
1.3

This function returns the portion(probability) of Poisson variables that are larger than or equal to 'm' amongst all 'n' variables characterized by the same 'lambda'.

R code:

```
f_poisson <- function(m,n,lambda){
  p_vec = rpois(n,lamba)
  m_prob = mean(p_vec>=m)
  return(m_prob)
}
```

1.4



```
> f_poisson <- function(m=10,n=1000,lambda=5){
+   p_vec = rpois(n,lambda)
+   m_prob = mean(p_vec>=m)
+   return(m_prob)
+ }

> "if no argument is provided"
[1] "if no argument is provided"

> f_poisson()
[1] 0.034

> "we may also override the arguments"
[1] "we may also override the arguments"

> f_poisson(m=15,n=500,lambda=8)
[1] 0.028
>
```

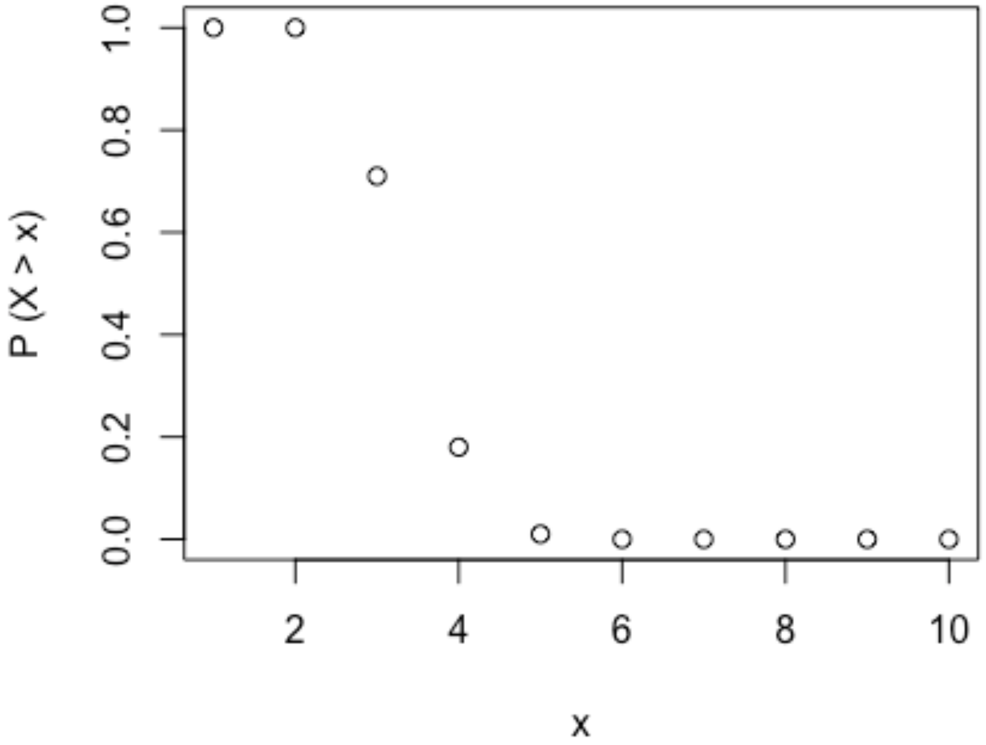
R code:

```
f_poisson <- function(m=10,n=1000,lambda=5){
  p_vec = rpois(n,lambda)
  m_prob = mean(p_vec>=m)
  return(m_prob)
}

"If no argument is provided, then..."
f_poisson()

"We may also override the arguments..."
f_poisson(m=15,n=500,lambda=8)
```

1.5

If we assume that λ stays 0.5, then for 100 trials of Poisson simulation we can see that $P(X > 9) \approx 0$.

If we want to prove that the probability is smaller than 10^{-6}, then we need to increase granularity of the simulation, with at least 10^6 trials. The user may feel free to change the “*trials*” variable below to 10^6, but the previous statement $P(X > 9) \approx 0$ remains true.



R Code:

```
#Assume that the false positive rate 'r' stays the same at 0.01
p <- 0.01
#Assume that the number of patient samples stays the same at 50
n <- 50
l_protein_position <- 100
```

```
#The 'lambda' parameter for the Poisson distribution is thus 0.5
lambda <- n*p
#If we simulate 100 trials, then the probability of finding a maximum greater than or equal to 9 is thus...
trials <- 100
maxes <- replicate(trials,{max(rpois(l_protein_position,lambda))})
prob <- formatC(mean(maxes>=9),format = "e", digits=8)
prob

#Let's investigate the trend, part I
X <- c(1:10)
Y <- c()
for (x in X)
{
y=as.double(mean(maxes>=x),format="e",digit=4)
Y <- c(Y,y)
}
plot(X,Y,xlab = "x",ylab = "P (X > x)")
```

1.8

a.

We can use the *letterFrequency* function in Biostrings to obtain

A	C	G	T
4335	1225	2055	6179

b.

Under the equal probability assumption, the χ^2 distribution with 3 degrees of freedom should have a distribution with the following characteristics from a 10^6 simulation:

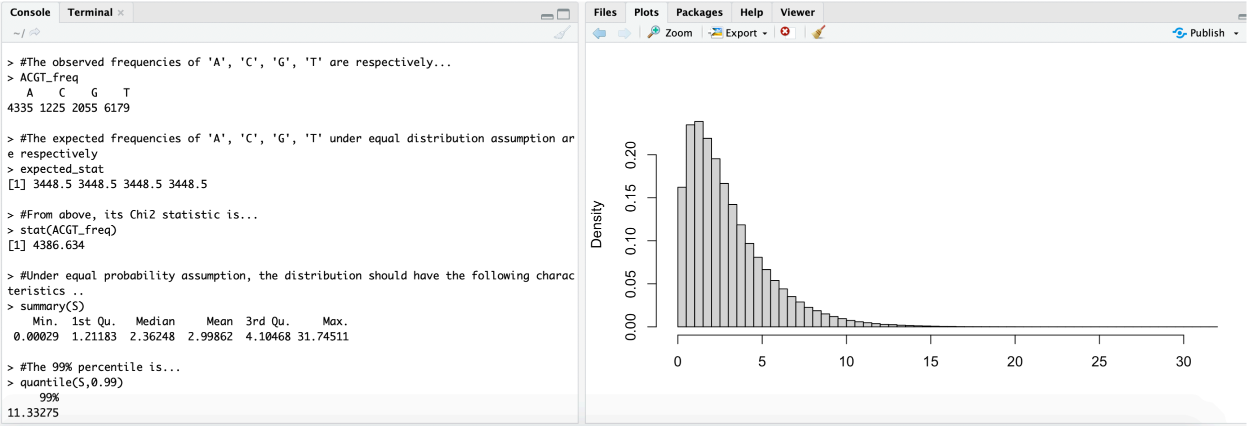
Min.	1^{st} Q	Median	Mean	3^{rd} Q	Max.
0.00029	1.21009	2.36654	3.00133	4.10932	29.42178

The theoretical results from using the *dchisq()* function also comfirmed the correctness of the simulation above, giving a median of approximately 2.3814 and a mean of 3.

Given that the 99% percentile of the distribution is around 11.3351. The probability that the *C.elegans* data is consistent with the uniform model is close to zero.

$$P_{\chi^2}(4386) \approx 0$$

In conclusion, we can be fairly certain the *C.elegans* data did not come from a uniform distribution.



```
> #The observed frequencies of 'A', 'C', 'G', 'T' are respectively...
> ACGT_freq
   A    C    G    T 
4335 1225 2055 6179 

> #The expected frequencies of 'A', 'C', 'G', 'T' under equal distribution assumption ar
e respectively
> expected_stat
[1] 3448.5 3448.5 3448.5 3448.5

> #From above, its Chi2 statistic is...
> stat(ACGT_freq)
[1] 4386.634

> #Under equal probability assumption, the distribution should have the following charac
teristics ..
> summary(S)
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max. 
 0.00029  1.21183  2.36248  2.99862  4.10468 31.74511 

> #The 99% percentile is...
> quantile(S,0.99)
     99% 
11.33275 
```

R Code:

```
seqnames(BSgenome.Celegans.UCSC.ce2)
M <- BSgenome.Celegans.UCSC.ce2[["chrM"]]
ACGT_freq                                                    <-
Biostrings::letterFrequency(letters=c("A","C","G","T"),M)
s = sum(ACGT_freq)

#Obtain the expectation value assuming As Cs Gs and Ts are equally
distributed
```

```
pvec = rep(1/4, 4)
expected_stat = pvec*s

equal_distribution = rmultinom(1000000, prob = pvec, size = s)

#Chi2 statistics
stat = function(observation, expectation = expected_stat){
  return(sum((observation-expectation)^2/expectation))
}
S = apply(equal_distribution,2,stat)

#The observed frequencies of 'A', 'C', 'G', 'T' are respectively...
ACGT_freq
#The expected frequencies of 'A', 'C', 'G', 'T' under equal
distribution assumption are respectively
expected_stat
#From above, its Chi2 statistic is...
stat(ACGT_freq)
#Under equal probability assumption, the distribution should have
the following characteristics ..
summary(S)
#The 99% percentile is...
quantile(S,0.99)
hist(S, breaks = 50, main="",freq=FALSE, xlab="")
abline(v = stat(ACGT_freq), col = "red")
```