# R-Project_501

Brock Akerman

11/16/2020

**Part 1: “Visualizing Convergence in Probability”**

Show that the minimum order statistic converges in probability to 0. Hint: We know the CDF of an exponential and how to find the CDF of the minimum order statistic Y(1). Start with the probability you want to show from the definition of convergence in probability to 0, i.e., P(|Y(1) − 0| < ε) and take the limit as n goes to infinity and show that this probability converges to 1.

To visualize this we’ll simulate data and approximate the probability statement proven in the previous part. – For a sample size of n = 1, generate N = 1000 data sets from an exp(1) distribution – For each data set, find the minimum value (for a sample of size 1 that will just be the value itself) – Save these minimum values for plotting

```r
set.seed(841)
n <- 1
N <- 1000
minmat <- matrix(ncol=1,nrow=1000)
for (i in 1:N){
    ds <- replicate(i,(rexp(n,rate = 1)))
    minmat[i] <- min(ds)
    }
i <- 1
```

Now set ε = 0.05. Next approximate the probability of interest P(|Y(1) − 0| <= ε) using the N = 1000 simulated minimum values. (This is a Monte Carlo estimate of the probability.) Save this probability

```r
ε <- 0.05
TF_values <- (abs(minmat-0) <= ε) #Assign Boolean values to a vector
dsprob <- sum(TF_values)/N #Probability of true values from our 1000 dataset minimums.
```

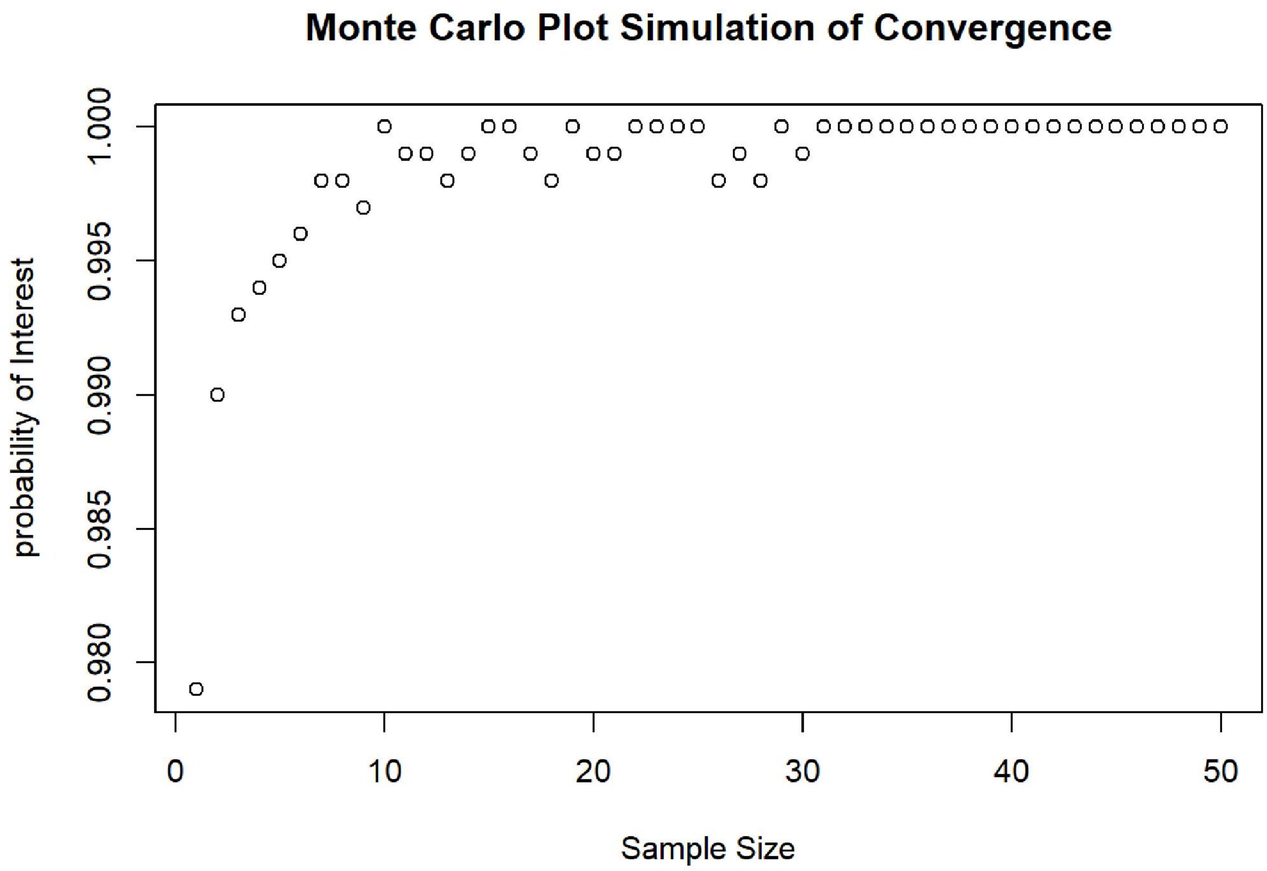
Repeat the above simulation and approximation of the probability of interest for n = 2, 3, …, 50.

```r
fullminmat <- matrix(ncol=50,nrow=1000)
for (j in 1:50){
  for (k in 1:N) {
  fullds <- replicate(k,(rexp(j,rate = 1)))
  fullminmat[k,j] <- min(fullds) } #Creates our
}
```

```
TFmat <- abs(fullminmat-0)<= ε

fulldsprob <- matrix(colSums(TFmat, na.rm=FALSE, dims=1),ncol=50,nrow=1)

probmatrix <- fulldsprob/N
```

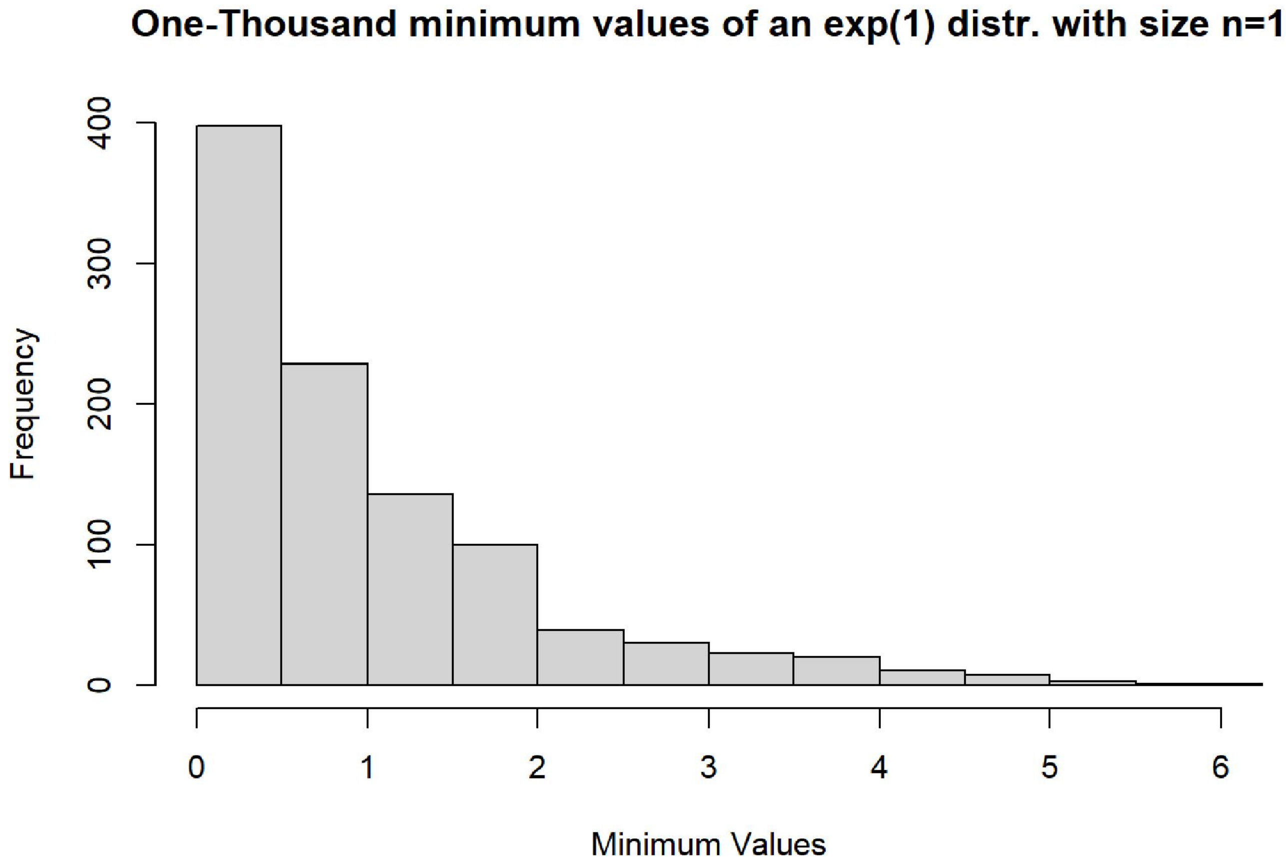
Now create a plot with the sample size on the x-axis and the probability of interest on the y-axis. The plot should have an appropriate title and appropriate axis labels. In a comment explain how this plot can help someone understand convergence in probability to a constant.

```
plot(x = 1:50,

     y=probmatrix,

     main="Monte Carlo Plot Simulation of Convergence",

     xlab="Sample Size",

     ylab="probability of Interest",

     )
```

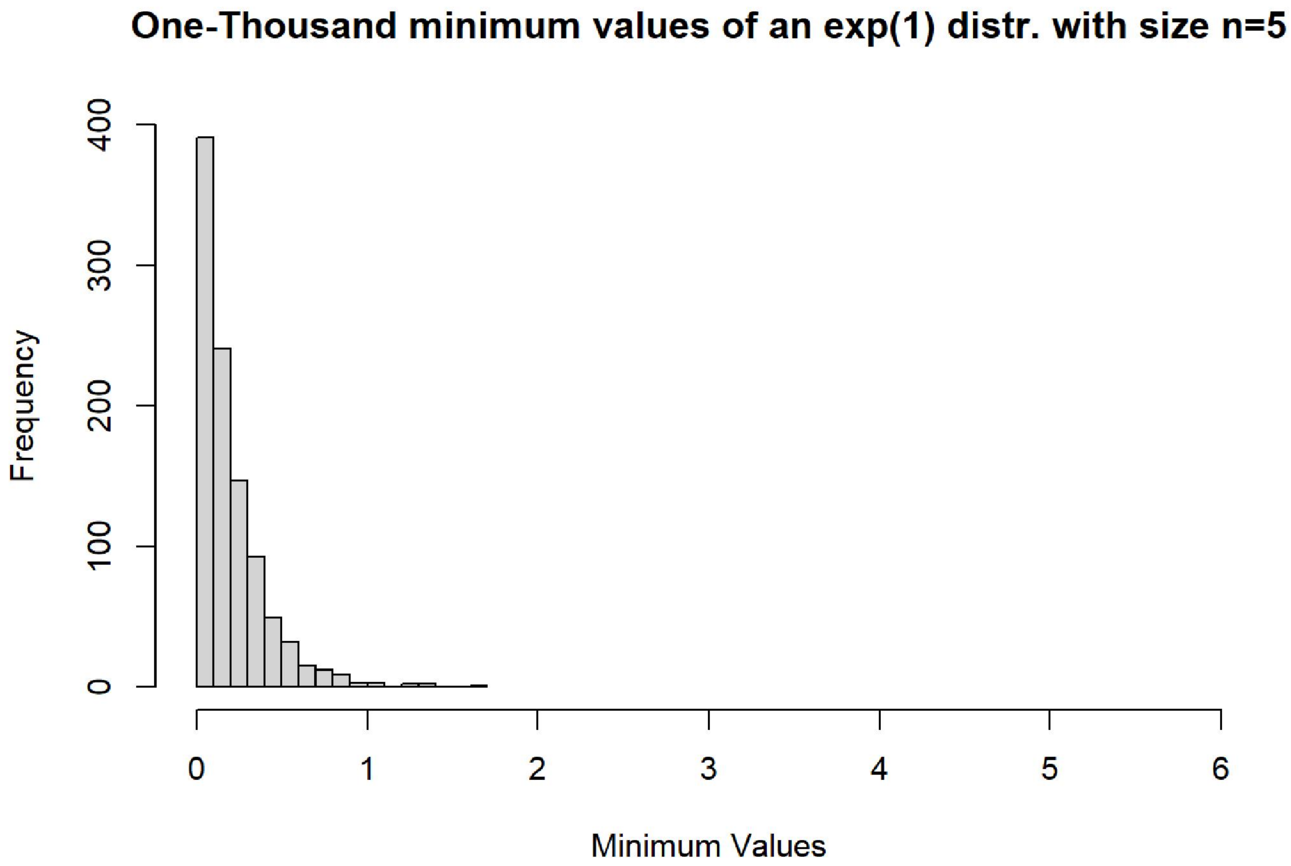


Now for each value of n{1, 5, 10, 25, 50}, draw one histogram of the minimum values for a sample of size n. You will thus have 5 histogram plots and, for example, the histogram plot for n = 10 will be a histogram plot of the N = 1000 minimum values for the N = 1000 samples of size n = 10. In a comment, explain how these histogram plots (for n changing), can help someone understand convergence in probability to a constant.

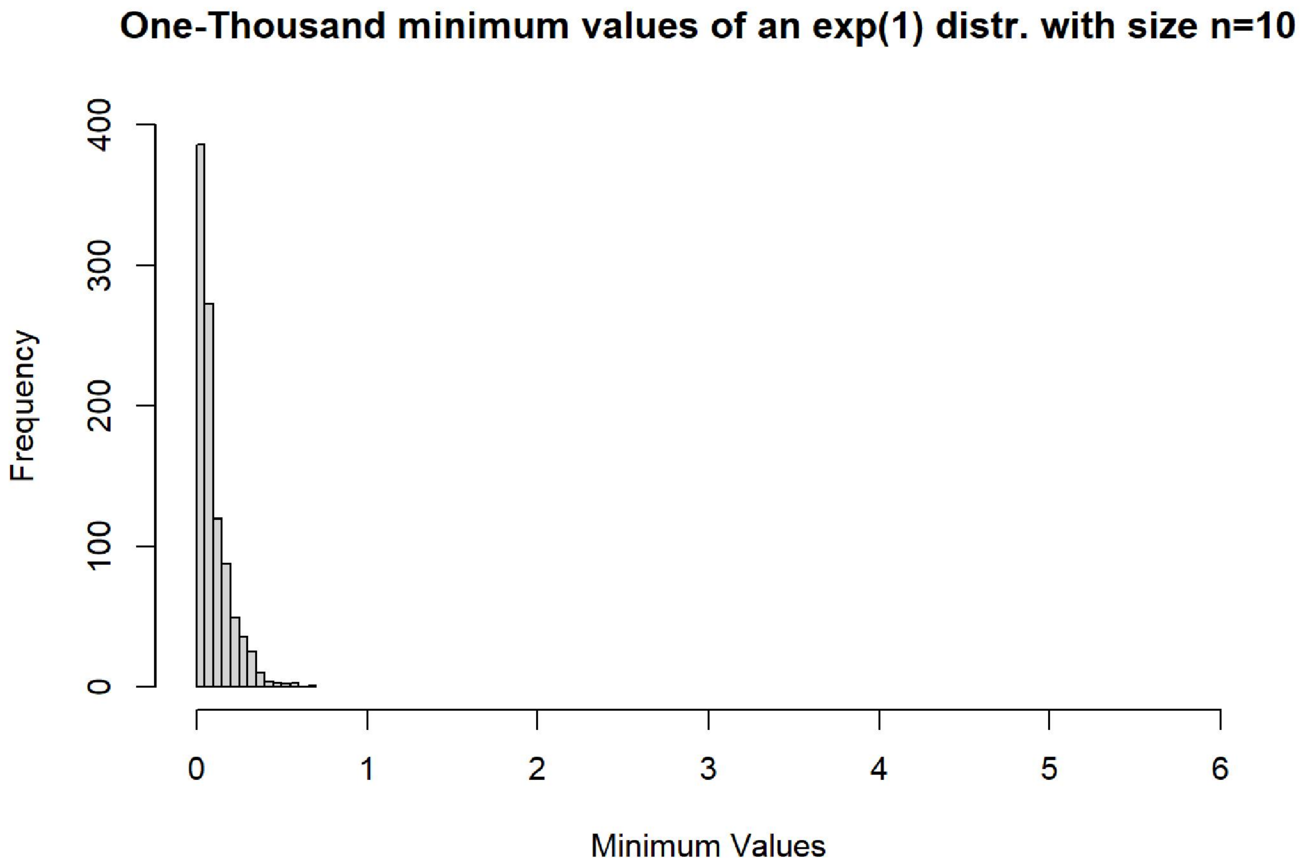
```
F01 <- sort(replicate(N, min(rexp(1,rate = 1))))

FM01 <- matrix(F01,nrow=N,ncol=1)

probability01 <- sum(FM01<=ε)/N

hist(FM01, breaks=20, prob=FALSE, xlim=c(0,6), ylim=c(0,400), main="One-Thousand minimum values of an exp(1) distr. with size n=1", xlab="Minimum Values", ylab="Frequency")
```



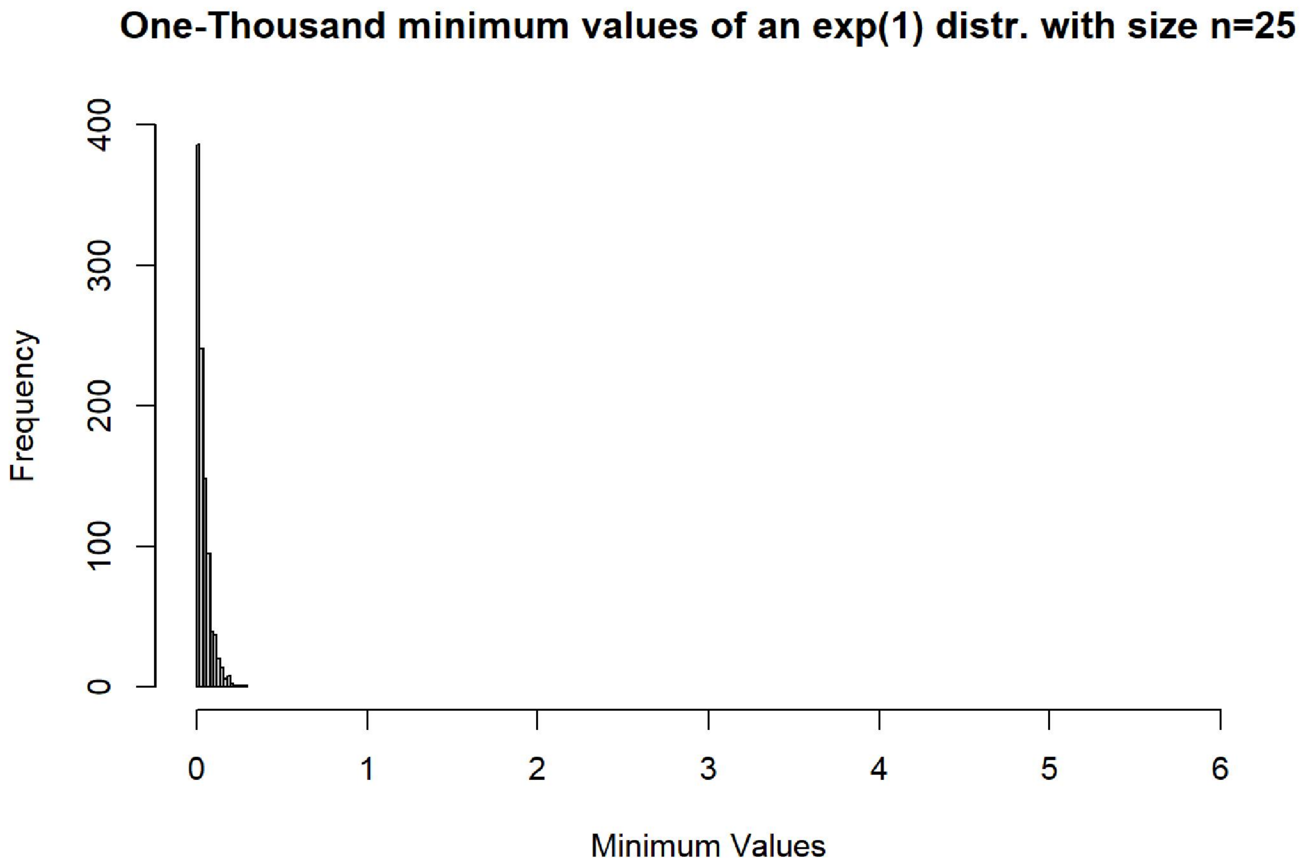
```
F05 <- sort(replicate(N, min(rexp(5,rate = 1))))

FM05 <- matrix(F05,nrow=N,ncol=1)

probability05 <- sum(FM05<=ε)/N

hist(FM05, breaks=20, prob=FALSE, xlim=c(0,6), ylim=c(0,400), main="One-Thousand minimum values of an exp(1) distr. with size n=5", xlab="Minimum Values", ylab="Frequency")
```



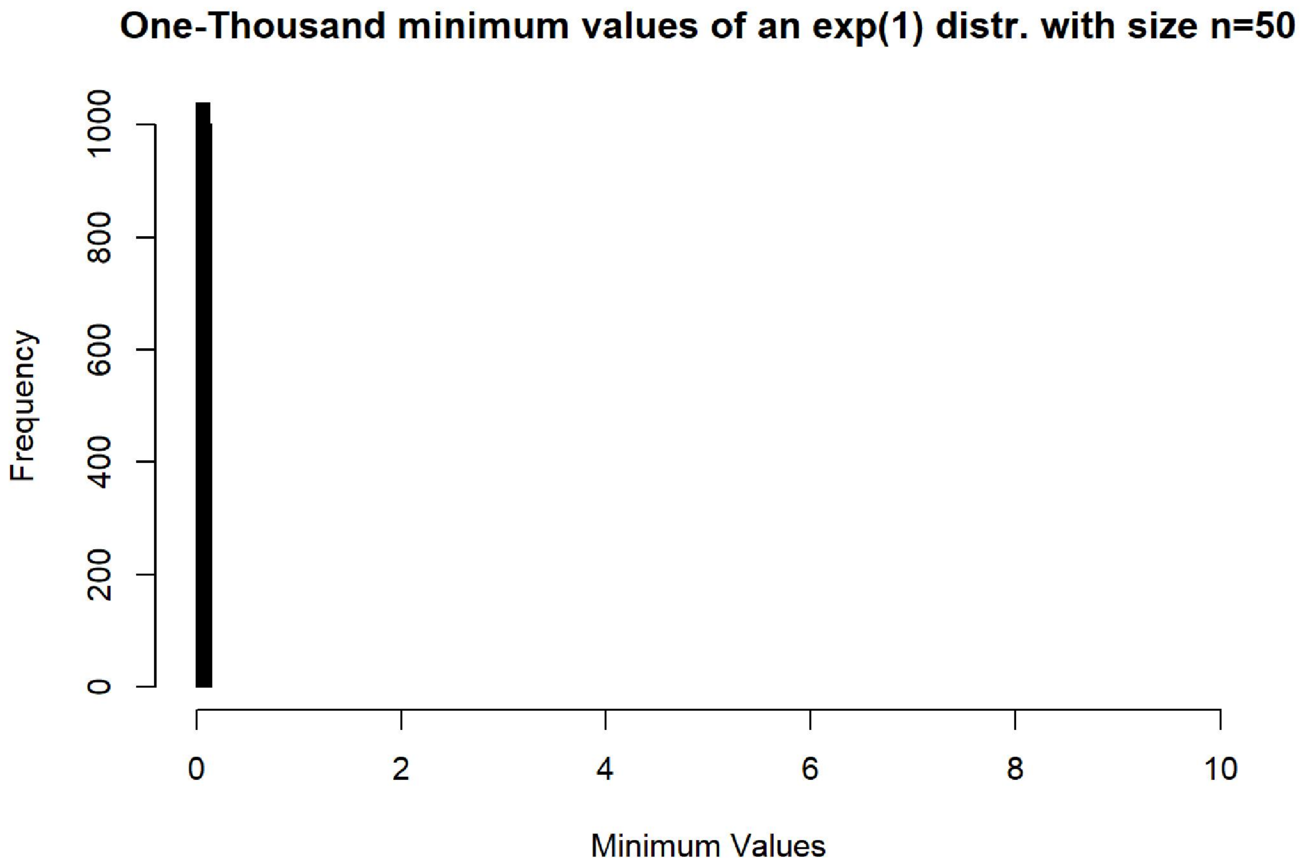
```r
F10 <- sort(replicate(N, min(rexp(10,rate = 1))))

FM10 <- matrix(F10,nrow=N,ncol=1)

probability10 <- sum(FM10<=ε)/N

hist(FM10, breaks=20, prob=FALSE, xlim=c(0,6), ylim=c(0,400), main="One-Thousand minimum values of an exp(1) distr. with size n=10", xlab="Minimum Values", ylab="Frequency")
```



```
F25 <- sort(replicate(N, min(rexp(25,rate = 1))))

FM25 <- matrix(F25,nrow=N,ncol=1)

probability25 <- sum(FM25<=ε)/N

hist(FM25, breaks=20, prob=FALSE, xlim=c(0,6), ylim=c(0,400), main="One-Thousand minim
um values of an exp(1) distr. with size n=25", xlab="Minimum Values", ylab="Frequency"
)
```



```
F50 <- sort(replicate(N, min(rexp(50,rate = 1))))

FM50 <- matrix(F50,nrow=N,ncol=1000)

probability50 <- sum(FM50<=ε)/N

hist(FM50, breaks=20, prob=FALSE, xlim=c(0,10), ylim=c(0,1000), main="One-Thousand min
imum values of an exp(1) distr. with size n=50", xlab="Minimum Values", ylab="Frequenc
y")
```



**Discussion Question:** With a small sampling of data, the Exp(1) shows some variability which manifests itself as a right-tail in the distribution. However, as sample size increases the Exp(1) distirbution begins to converge on zero. With enough samples, such is the case when n = 50, the distribution nearly resembles a vertical line about zero. There are values within the minimum value dataset that are greater than zero. There are simply too many values at or nearer to zero that the exp(1) looks almost tail-less. In fact, that is precisely what is occuring. In a convergence in probability to a constant–in the case of Exp(1) is zero–the Exp(1) is converging to a number while its variance collapses to zero. When looking at our ordered datasets, from n=1 to n=50, the distribution is consistently becoming thinner and "squishing" toward the y-axis.

**Part 2: "Visualizing Convergence in Distribution"** This part will consider how well the Central Limit Theorem applies to sample means from Poisson data

Consider a sample size of n = 5 from a Poisson distribution with rate parameter λλ = 1. Generate N = 50000 data sets of size n from the Poisson distribution.

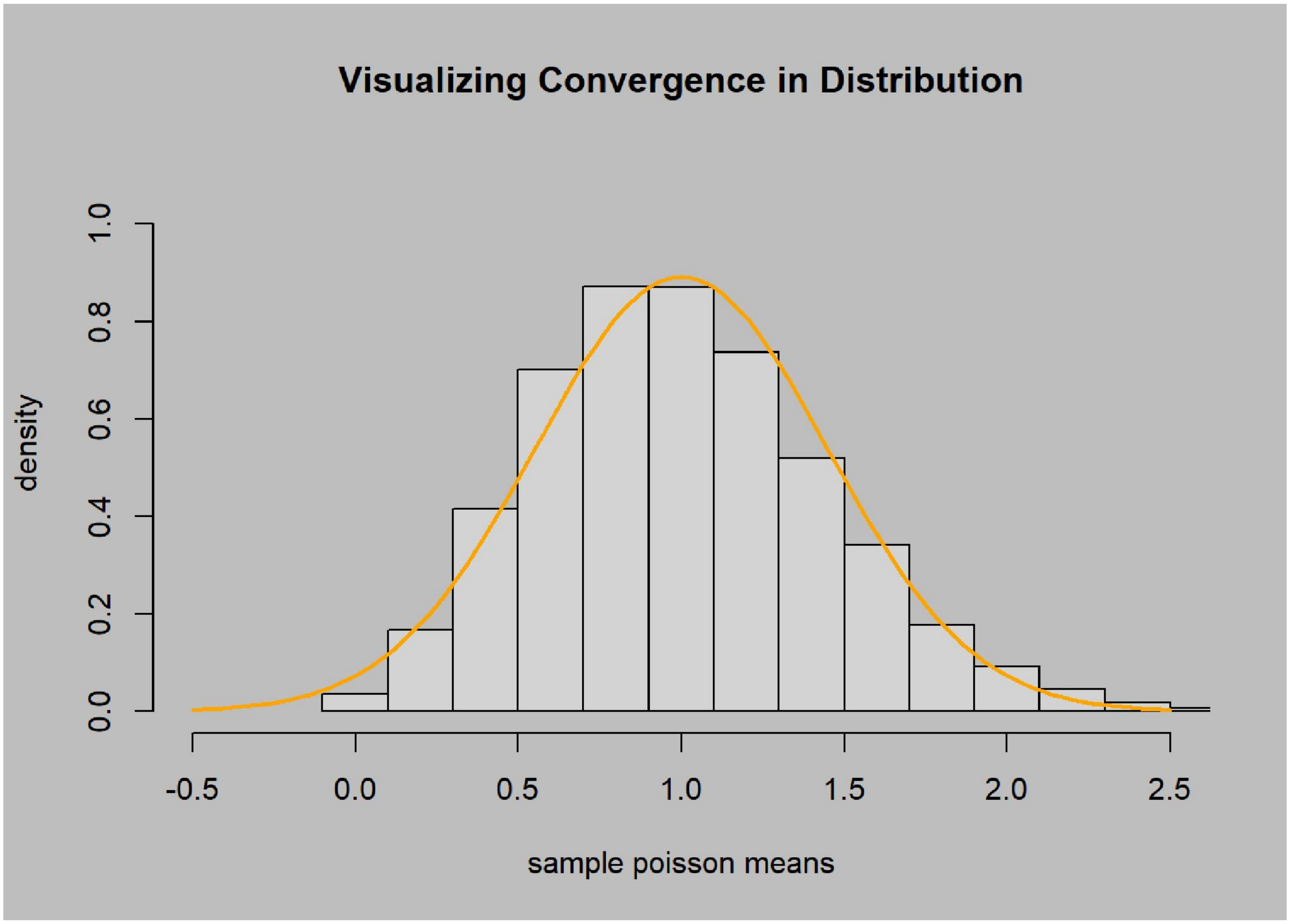
```
m = 5

lambda = 1

M = 50000

ds2 <- matrix(replicate(M,(rpois(m,1))),nrow=50000,ncol=5)
```

For each data set, find the sample mean value. Hint If you saved the above data in a large matrix the colMeans or rowMeans functions can be handy here), e.g., n <- 5 N <- 50000 A matrix of N = 50000 rows, each rows containing n = 5 sample from a Poisson distribution with rate 1 X <- matrix(rpois(n*N, λλ = 1), nrow = N), rowMeans(X) - This will give you a vector of N = 50000 sample means.

```
rMeans <- matrix(rowMeans(ds2),nrow=50000,ncol=1);
```

Create a histogram of the sample means. Make the bins of appropriate width so that each bin only has one value of the support. For instance, the possible values for the sample mean here are 0, 0.2, 0.4, 0.6, . . .. Make sure that each bar only has one of these values included (so the bins would go from say −0.1 to 0.1, 0.1 to 0.3, 0.3 to 0.5. The use of the breaks argument for the histogram function hist will be helpful here. The central limit theorem says that X¯X¯ ~ N(λλ, λλ/n) when X¯X¯ is the sample mean of n i.i.d Pois(λλ) random variables. Overlay this large-sample distribution on the histogram (Hint:use freq = FALSE in your histogram and the curve function with add = TRUE to overlay the normal distribution). All plots should have appropriate titles and axis labels.

```
par(bg="gray")

binlen <- c(-0.1,0.1,0.3,0.5,0.7,0.9,1.1,1.3,1.5,1.7,1.9,2.1,2.3,2.5,2.7,2.9,3.1,3.3,3.5,3.7,3.9,4.1);

hist(rMeans ,breaks=binlen ,main="Visualizing Convergence in Distribution" ,xlab="sample poisson means" ,xlim=c(-0.5,2.5) ,ylab="density" ,ylim=c(0,1.1) ,freq = FALSE)

x <- seq(-4,5, by = 0.1)

y <- dnorm(x, mean = lambda, sd = lambda/sqrt(m))

curve(dnorm(x, mean=lambda, sd=lambda/sqrt(m)), add=TRUE, col="orange", lwd=2)
```



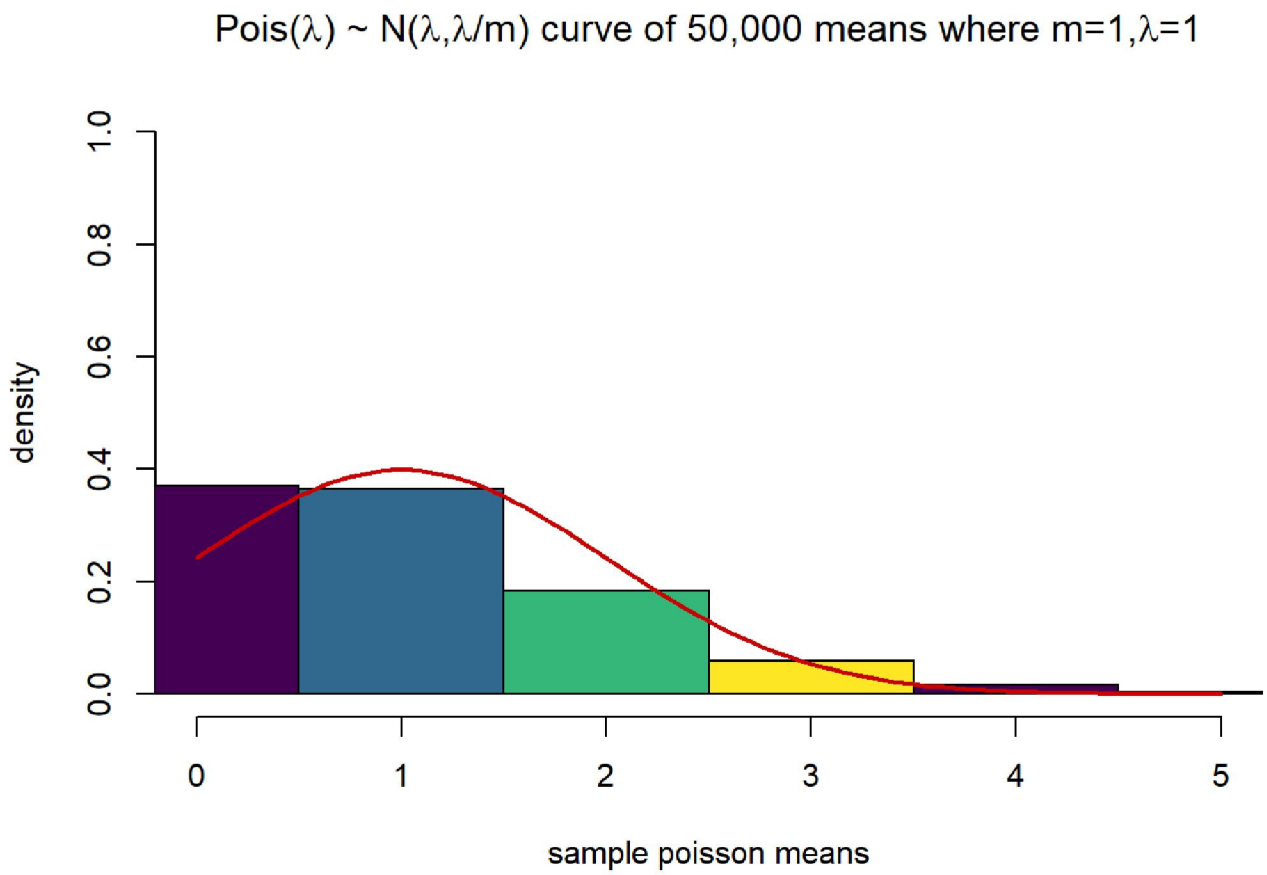
Use the N = 50000 [mean] values to approximate the probability that X¯X¯ is greater than or equal to λλ + 2λλ/sqrt(n). Also report this probability as approximated by the normal distribution.

```
TF_values.2 <- rMeans >= (lambda+((2*lambda)/sqrt(m)))

TFprob <- sum(TF_values.2)/M

print(1-TFprob)

## [1] 0.967

NormProb <- 1-dnorm((lambda+((2*lambda)/sqrt(m))),mean=lambda,sd=(lambda/sqrt(m)))

print(NormProb)

## [1] 0.8792725
```

Repeat the above for n = 10, n = 30, and n = 100. FOR m = 1,λλ = 1

```
m1 = 1; lambda.1 = 1; M = 50000

ds1 <- matrix(replicate(M,(rpois(m1,lambda.1))),nrow=50000,ncol=5)

rMeans1 <- matrix(rowMeans(ds1),nrow=50000,ncol=1)

binlen <- c(-0.5,0.5,1.5,2.5,3.5,4.5,5.5,6.5,7.5,8.5,9.5,10.5,11.5,12.5,13.5,14.5,15.5
,16.5,17.5,18.5,19.5,20.5,21.5,22.5);
```

```
hist(rMeans1,breaks=binlen ,main=expression(paste("Pois(",lambda,") ~ N(",lambda,",",lambda,"/m) curve of 50,000 means where m=1,",lambda,"=1"))  ,xlab="sample poisson means" ,xlim=c(0,5) ,ylab="density" ,ylim=c(0,1) ,freq = FALSE, col=viridis(4))

x <- seq(0,2, by = 0.2)

y1 <- dnorm(x, mean=lambda.1, sd=lambda.1/(sqrt(m1)))

curve(dnorm(x, mean=lambda.1, sd=(lambda.1/sqrt(m1))), add=TRUE, col="#cc0000", lwd=2)
```

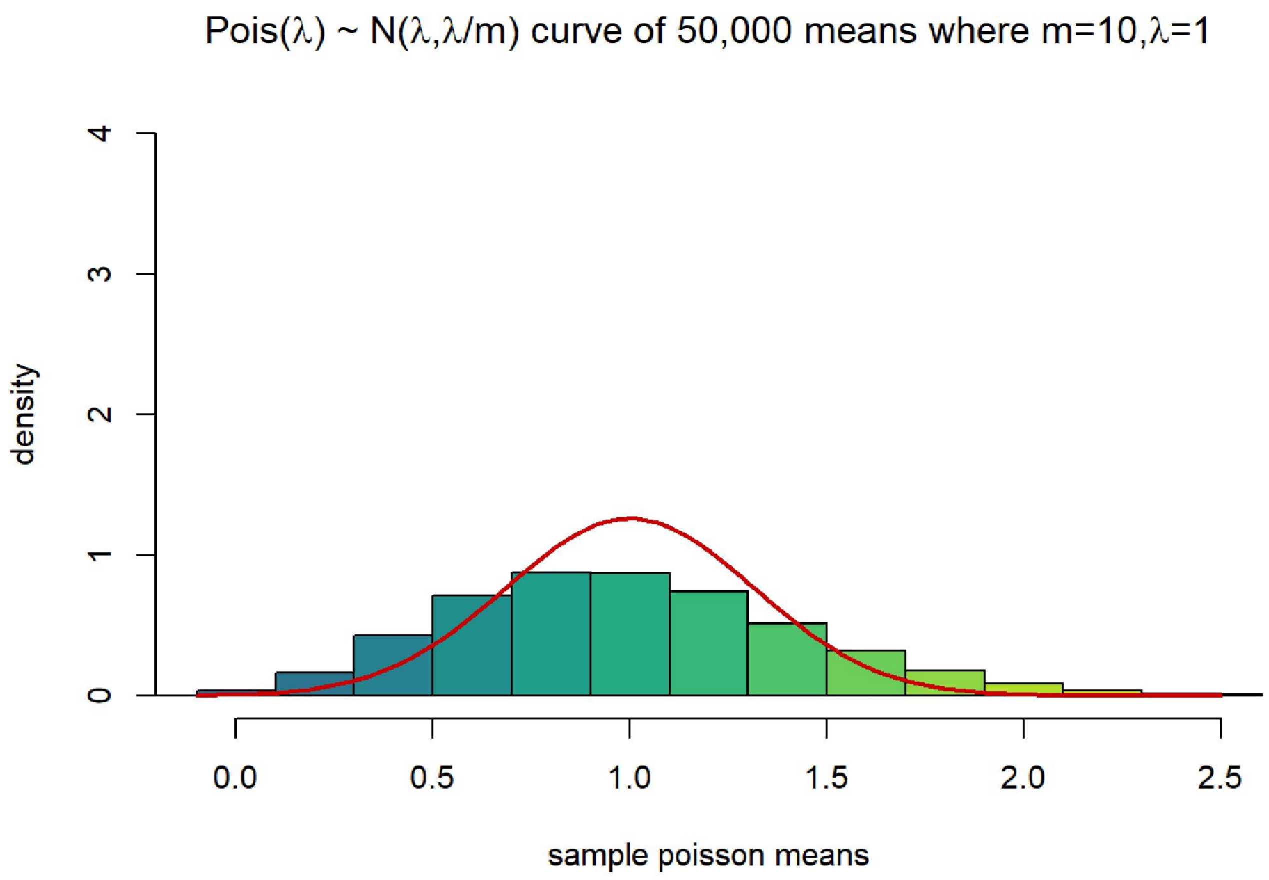


```
TF_values.1 <- rMeans1 >= (lambda.1+((2*lambda.1)/sqrt(m1)))

TFprob1 <- sum(TF_values.1)/M; print(TFprob1)

## [1] 0.07924

NormProb1 <- dnorm((lambda.1+((2*lambda.1)/sqrt(m1))),mean=lambda.1,sd=(lambda.1/sqrt(m1))); print(NormProb1)

## [1] 0.05399097
```

FOR m = 10,λλ = 1

```
m10 = 10; lambda.1 = 1; M = 50000

ds10 <- matrix(replicate(M,(rpois(m10,1))),nrow=50000,ncol=5)
```

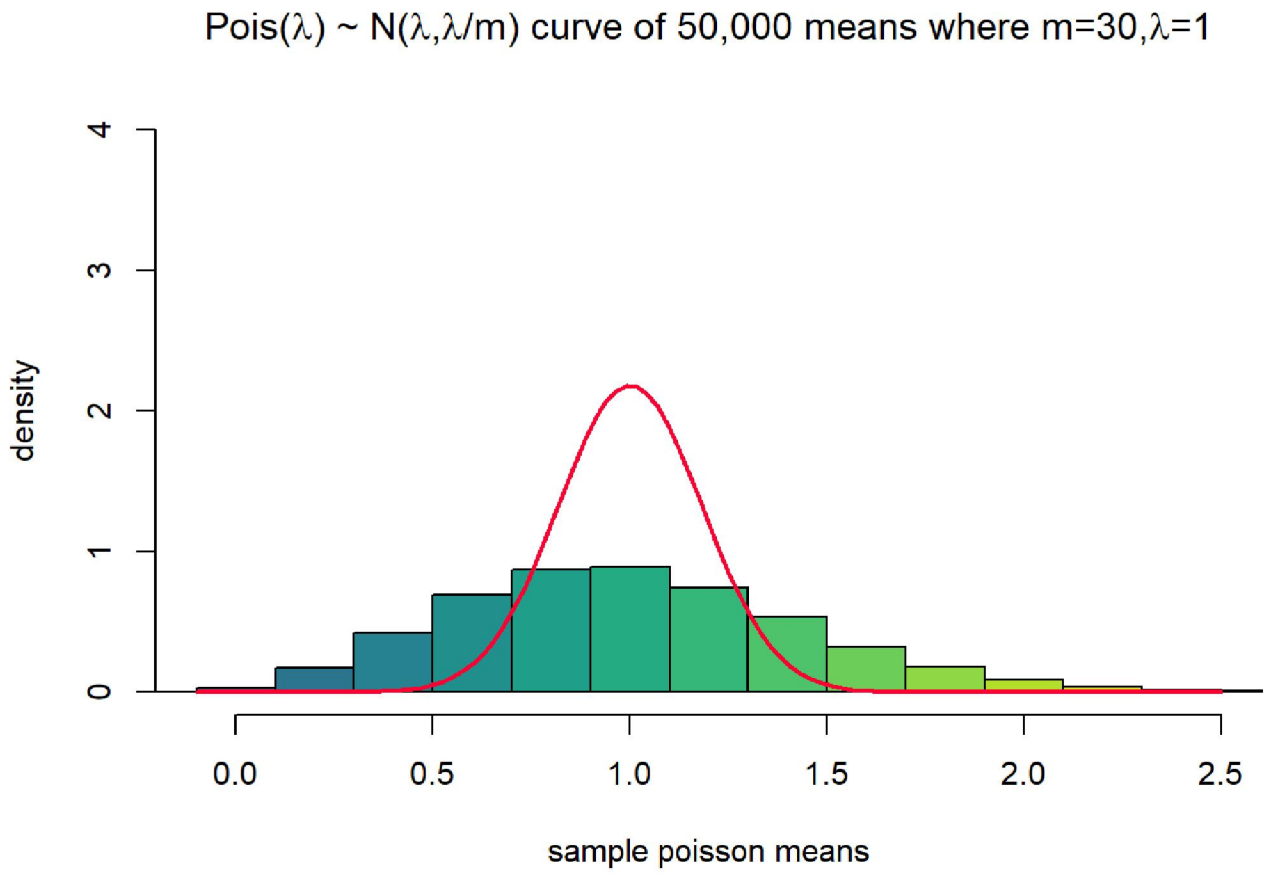
```
rMeans10 <- matrix(rowMeans(ds10),nrow=50000,ncol=1)

binlen <- c(-5.1,-4.9,-4.7,-4.5,-4.3,-4.1,-3.9,-3.7,-3.5,-3.3,-3.1,-2.9,-2.7,-2.5,-2.3
,-2.1,-1.9,-1.7,-1.5,-1.3,-1.1,-0.9,-0.7,-0.5,-0.3,-0.1,0.1,0.3,0.5,0.7,0.9,1.1,1.3,1.
5,1.7,1.9,2.1,2.3,2.5,2.7,2.9,3.1,3.3,3.5,3.7,3.9,4.1,4.3,4.5,4.7,4.9,5.1)

hist(rMeans10,breaks=binlen ,main=expression(paste("Pois(",lambda,") ~ N(",lambda,",",
lambda,"/m) curve of 50,000 means where m=10,",lambda,"=1"))  ,xlab="sample poisson me
ans" ,xlim=c(-0.1,2.5) ,ylab="density" ,ylim=c(0,4) ,freq = FALSE, col=viridis(19))

x <- seq(0,2, by = 0.2)

y10 <- dnorm(x, mean=lambda.1, sd=lambda.1/(sqrt(m10)))

curve(dnorm(x, mean=lambda.1, sd=(lambda.1/sqrt(m10))), add=TRUE, col="#cc0000", lwd=2
)
```



```
TF_values.10 <- rMeans10 >= (lambda.1+((2*lambda.1)/sqrt(m10)))

TFprob10 <- sum(TF_values.10)/M; print(TFprob10)

## [1] 0.0668

NormProb10 <- dnorm((lambda.1+((2*lambda.1)/sqrt(m10))),mean=lambda.1,sd=(lambda.1/sqr
t(m10))); print(NormProb10)

## [1] 0.1707344
```

FOR m = 30,λλ = 1

```
m30 = 30; lambda.1 = 1; M = 30000

ds30 <- matrix(replicate(M,(rpois(m30,1))),nrow=30000,ncol=5)

rMeans30 <- matrix(rowMeans(ds30),nrow=30000,ncol=1)

binlen <- c(-5.1,-4.9,-4.7,-4.5,-4.3,-4.1,-3.9,-3.7,-3.5,-3.3,-3.1,-2.9,-2.7,-2.5,-2.3
,-2.1,-1.9,-1.7,-1.5,-1.3,-1.1,-0.9,-0.7,-0.5,-0.3,-0.1,0.1,0.3,0.5,0.7,0.9,1.1,1.3,1.
5,1.7,1.9,2.1,2.3,2.5,2.7,2.9,3.1,3.3,3.5,3.7,3.9,4.1,4.3,4.5,4.7,4.9,5.1)

hist(rMeans30,breaks=binlen ,main=expression(paste("Pois(",lambda,") ~ N(",lambda,",",
lambda,"/m) curve of 50,000 means where m=30,",lambda,"=1"))  ,xlab="sample poisson me
ans" ,xlim=c(-0.1,2.5) ,ylab="density" ,ylim=c(0,4) ,freq = FALSE, col=viridis(19))

x <- seq(0,2, by = 0.2)

y30 <- dnorm(x, mean=lambda.1, sd=lambda.1/(sqrt(m30)))

curve(dnorm(x, mean=lambda.1, sd=(lambda.1/sqrt(m30))), add=TRUE, col="#FF0033", lwd=2
)
```

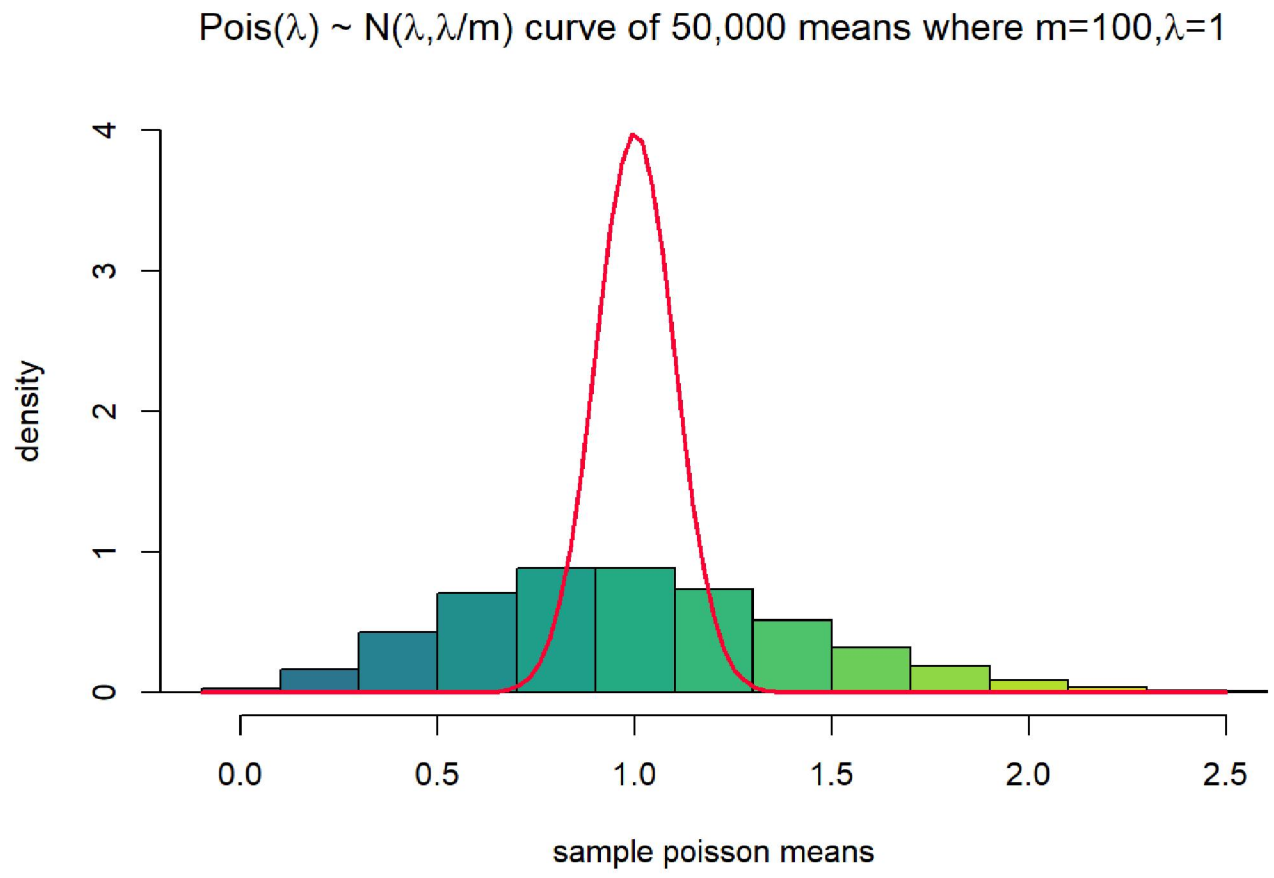


```
TF_values.30 <- rMeans30 >= (lambda.1+((2*lambda.1)/sqrt(m30)))

TFprob30 <- sum(TF_values.30)/M; print(TFprob30)

## [1] 0.238
```

```
NormProb30 <- dnorm((lambda.1+((2*lambda.1)/sqrt(m30))),mean=lambda.1,sd=(lambda.1/sqr
t(m30))); print(NormProb30)

## [1] 0.2957207
```

FOR m = 100,λλ = 1

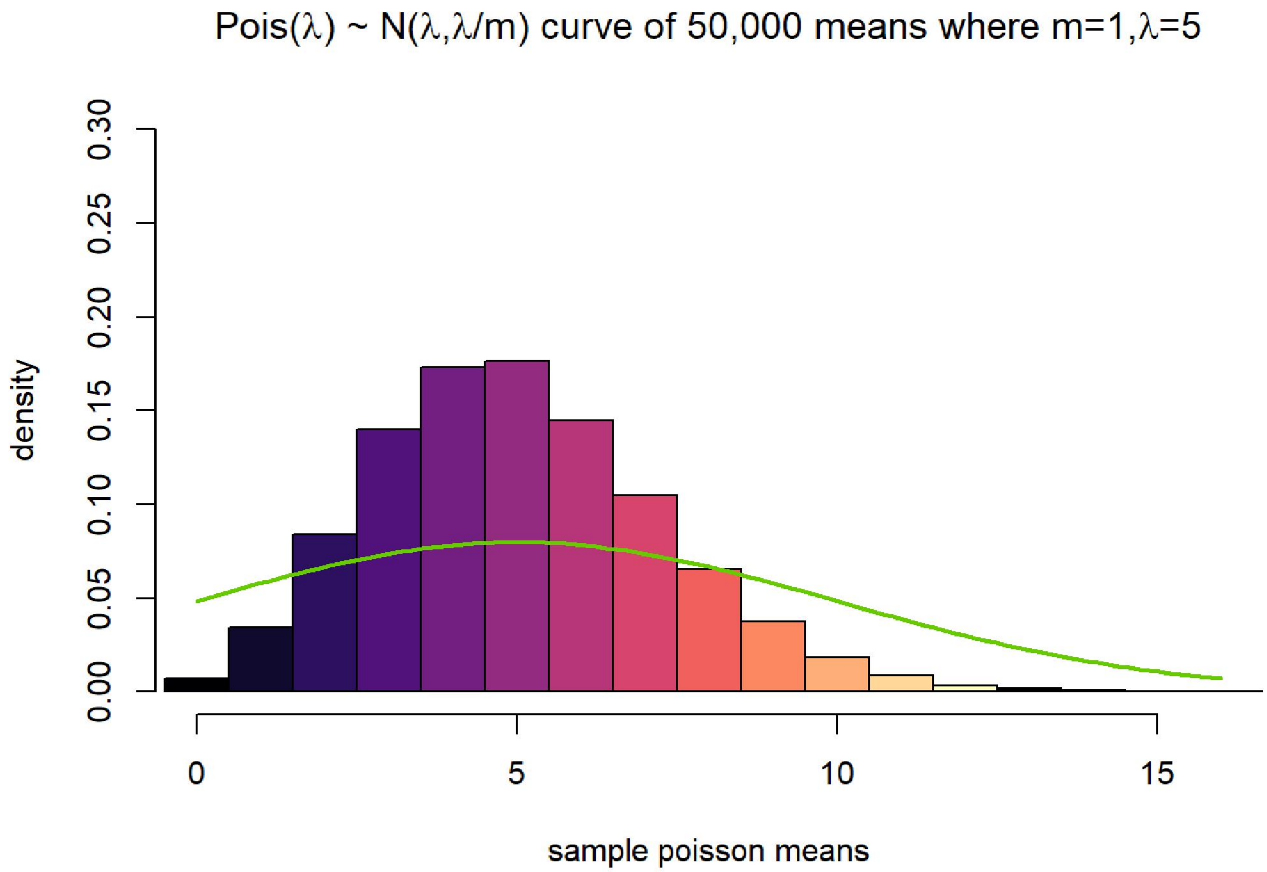
```
m100 = 100; lambda.1 = 1; M = 50000

ds100 <- matrix(replicate(M,(rpois(m100,1))),nrow=50000,ncol=5)

rMeans100 <- matrix(rowMeans(ds100),nrow=50000,ncol=1)

binlen <- c(-5.1,-4.9,-4.7,-4.5,-4.3,-4.1,-3.9,-3.7,-3.5,-3.3,-3.1,-2.9,-2.7,-2.5,-2.3
,-2.1,-1.9,-1.7,-1.5,-1.3,-1.1,-0.9,-0.7,-0.5,-0.3,-0.1,0.1,0.3,0.5,0.7,0.9,1.1,1.3,1.
5,1.7,1.9,2.1,2.3,2.5,2.7,2.9,3.1,3.3,3.5,3.7,3.9,4.1,4.3,4.5,4.7,4.9,5.1)

hist(rMeans100,breaks=binlen ,main=expression(paste("Pois(",lambda,") ~ N(",lambda,","
,lambda,"/m) curve of 50,000 means where m=100,",lambda,"=1"))  ,xlab="sample poisson
means" ,xlim=c(-0.1,2.5) ,ylab="density" ,ylim=c(0,4) ,freq = FALSE, col=viridis(19))

x <- seq(0,2, by = 0.2)

y100 <- dnorm(x, mean=lambda.1, sd=lambda.1/(sqrt(m100)))

curve(dnorm(x, mean=lambda.1, sd=(lambda.1/sqrt(m100))), add=TRUE, col="#FF0033", lwd=
2)
```



```
TF_values.100 <- rMeans100 >= (lambda.1+((2*lambda.1)/sqrt(m100)))

TFprob100 <- sum(TF_values.100)/M; print(TFprob100)

## [1] 0.38108

NormProb100 <- dnorm((lambda.1+((2*lambda.1)/sqrt(m100))),mean=lambda.1,sd=(lambda.1/sqrt(m100))); print(NormProb100)

## [1] 0.5399097
```

Repeat all of the above for λλ = 5 and λλ = 25. You should have a total of 12 scenarios/plots FOR m = 1,λλ = 5

```
m1 = 1; lambda.5 = 5; M = 50000

ds1 <- matrix(replicate(M,(rpois(m1,lambda.5))),nrow=50000,ncol=5)

rMeans1 <- matrix(rowMeans(ds1),nrow=50000,ncol=1)

binlen <- c(-0.5,0.5,1.5,2.5,3.5,4.5,5.5,6.5,7.5,8.5,9.5,10.5,11.5,12.5,13.5,14.5,15.5,16.5,17.5,18.5,19.5,20.5,21.5,22.5,23.5,24.5,25.5,26.5,27.5,28.5,29.5,30.5)

hist(rMeans1,breaks=binlen ,main=expression(paste("Pois(",lambda,") ~ N(",lambda,",",lambda,"/m) curve of 50,000 means where m=1,",lambda,"=5"))  ,xlab="sample poisson means" ,xlim=c(0,16) ,ylab="density" ,ylim=c(0,0.3) ,freq = FALSE, col=magma(13))

x <- seq(-4,5, by = 0.1)

y1 <- dnorm(x, mean = lambda.5, sd = lambda.5/sqrt(m1))

curve(dnorm(x, mean=lambda.5, sd=lambda.5/sqrt(m1)), add=TRUE, col="#66CC00", lwd=2)
```

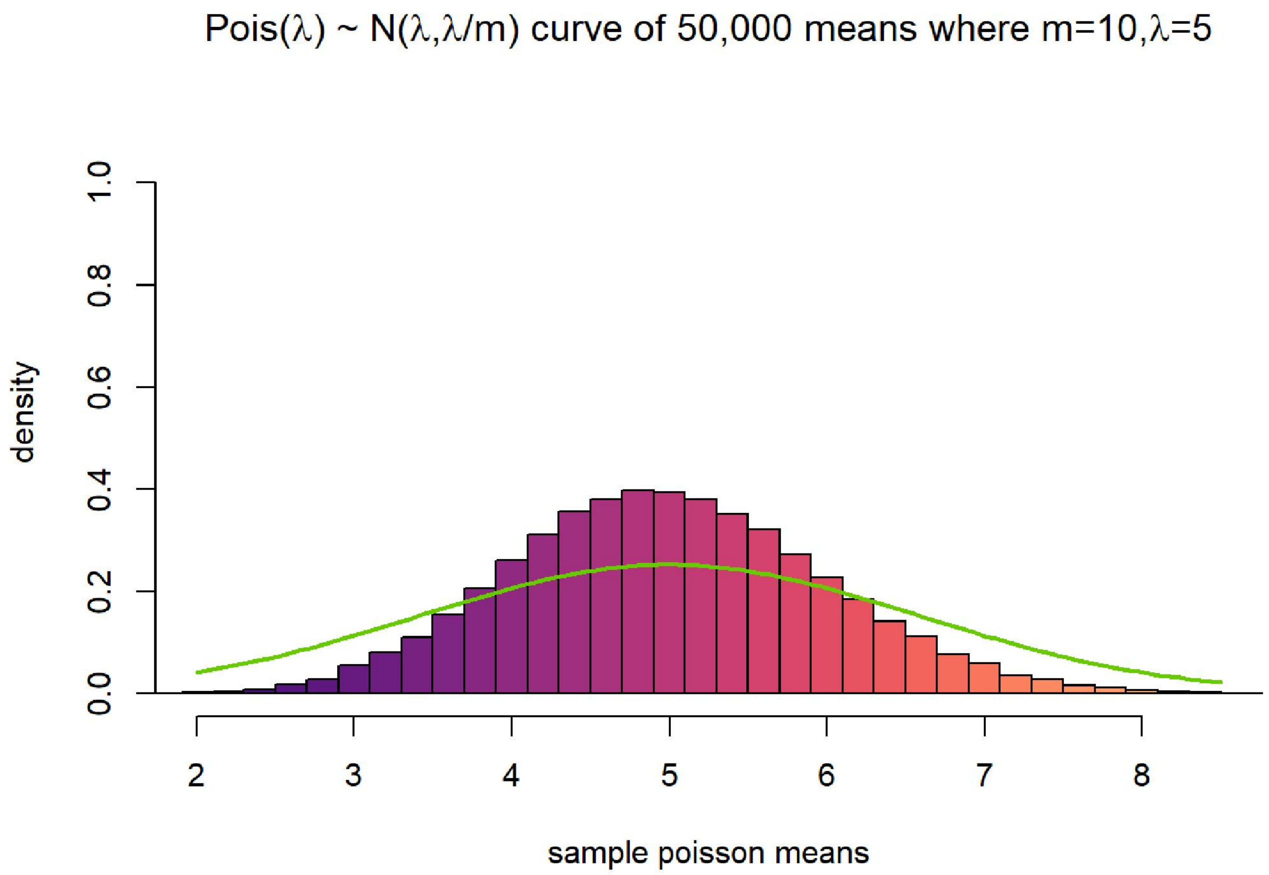


```
TF_values.1 <- rMeans1 >= (lambda.5+((2*lambda.5)/sqrt(m1)))

TFprob1 <- sum(TF_values.1)/M; print(TFprob1)

## [1] 0.00016

NormProb1 <- dnorm((lambda.5+((2*lambda.5)/sqrt(m1))),mean=lambda.5,sd=(lambda.5/sqrt(
m1))); print(NormProb1)

## [1] 0.01079819
```

FOR m = 10,λλ = 5

```
m10 = 10; lambda.5 = 5; M = 50000

ds10 <- matrix(replicate(M,(rpois(m10,lambda.5))),nrow=50000,ncol=5)

rMeans10 <- matrix(rowMeans(ds10),nrow=50000,ncol=1)

binlen <- c(-0.1,0.1,0.3,0.5,0.7,0.9,1.1,1.3,1.5,1.7,1.9,2.1,2.3,2.5,2.7,2.9,3.1,3.3,3
.5,3.7,3.9,4.1,4.3,4.5,4.7,4.9,5.1,5.3,5.5,5.7,5.9,6.1,6.3,6.5,6.7,6.9,7.1,7.3,7.5,7.7
,7.9,8.1,8.3,8.5,8.7,8.9,9.1,9.3,9.5,9.7,9.9,10.1,10.3,10.5,10.7,10.9,11.1,11.3,11.5,1
1.7,11.9)

hist(rMeans10,breaks=binlen ,main=expression(paste("Pois(",lambda,") ~ N(",lambda,",",
lambda,"/m) curve of 50,000 means where m=10,",lambda,"=5"))  ,xlab="sample poisson me
ans" ,xlim=c(2,8.5) ,ylab="density" ,ylim=c(0,1.1) ,freq = FALSE, col=magma(50))
```

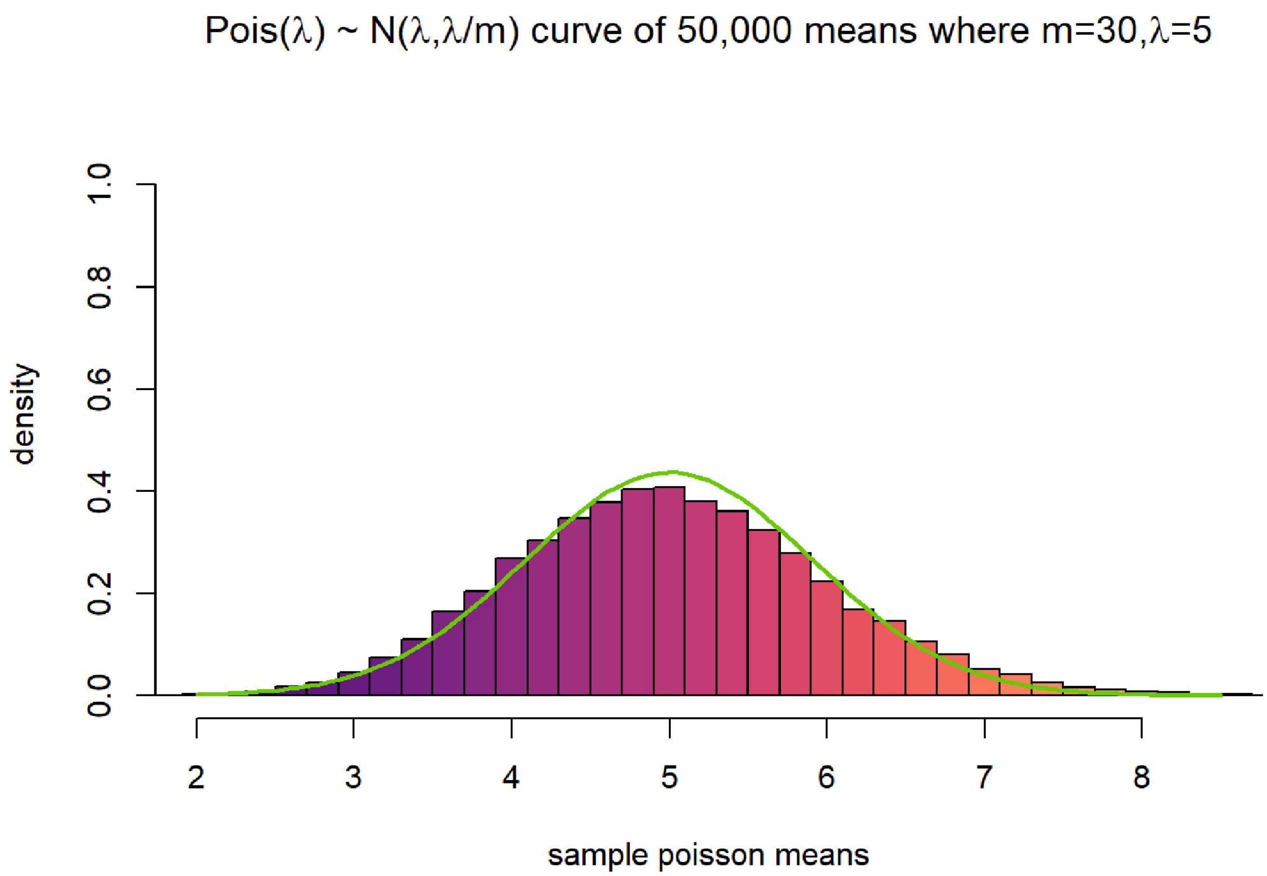
```
x <- seq(-4,5, by = 0.1)

y10 <- dnorm(x, mean = lambda.5, sd = lambda.5/sqrt(m10))

curve(dnorm(x, mean=lambda.5, sd=lambda.5/sqrt(m10)), add=TRUE, col="#66CC00", lwd=2)
```



```
TF_values.10 <- rMeans10 >= (lambda.5+((2*lambda.5)/sqrt(m10)))

TFprob10 <- sum(TF_values.10)/M; print(TFprob10)

## [1] 0.0018

NormProb10 <- dnorm((lambda.5+((2*lambda.5)/sqrt(m10))),mean=lambda.5,sd=(lambda.5/sqr
t(m10))); print(NormProb10)

## [1] 0.03414689
```

FOR m = 30,λλ = 5

```
m30 = 30; lambda.5 = 5; M = 30000

ds30 <- matrix(replicate(M,(rpois(m30,lambda.5))),nrow=30000,ncol=5)

rMeans30 <- matrix(rowMeans(ds30),nrow=30000,ncol=1)

binlen <- c(-0.1,0.1,0.3,0.5,0.7,0.9,1.1,1.3,1.5,1.7,1.9,2.1,2.3,2.5,2.7,2.9,3.1,3.3,3
.5,3.7,3.9,4.1,4.3,4.5,4.7,4.9,5.1,5.3,5.5,5.7,5.9,6.1,6.3,6.5,6.7,6.9,7.1,7.3,7.5,7.7
,7.9,8.1,8.3,8.5,8.7,8.9,9.1,9.3,9.5,9.7,9.9,10.1,10.3,10.5,10.7,10.9,11.1,11.3,11.5,11.7,11.9)

hist(rMeans30,breaks=binlen ,main=expression(paste("Pois(",lambda,") ~ N(",lambda,",",lambda,"/m) curve of 50,000 means where m=30,",lambda,"=5")) ,xlab="sample poisson means" ,xlim=c(2,8.5) ,ylab="density" ,ylim=c(0,1.1) ,freq = FALSE, col=magma(50))

x30 <- seq(-4,5, by = 0.1)

y30 <- dnorm(x30, mean = lambda.5, sd = lambda.5/sqrt(m30))

curve(dnorm(x, mean=lambda.5, sd=lambda.5/sqrt(m30)), add=TRUE, col="#66CC00", lwd=2)
```

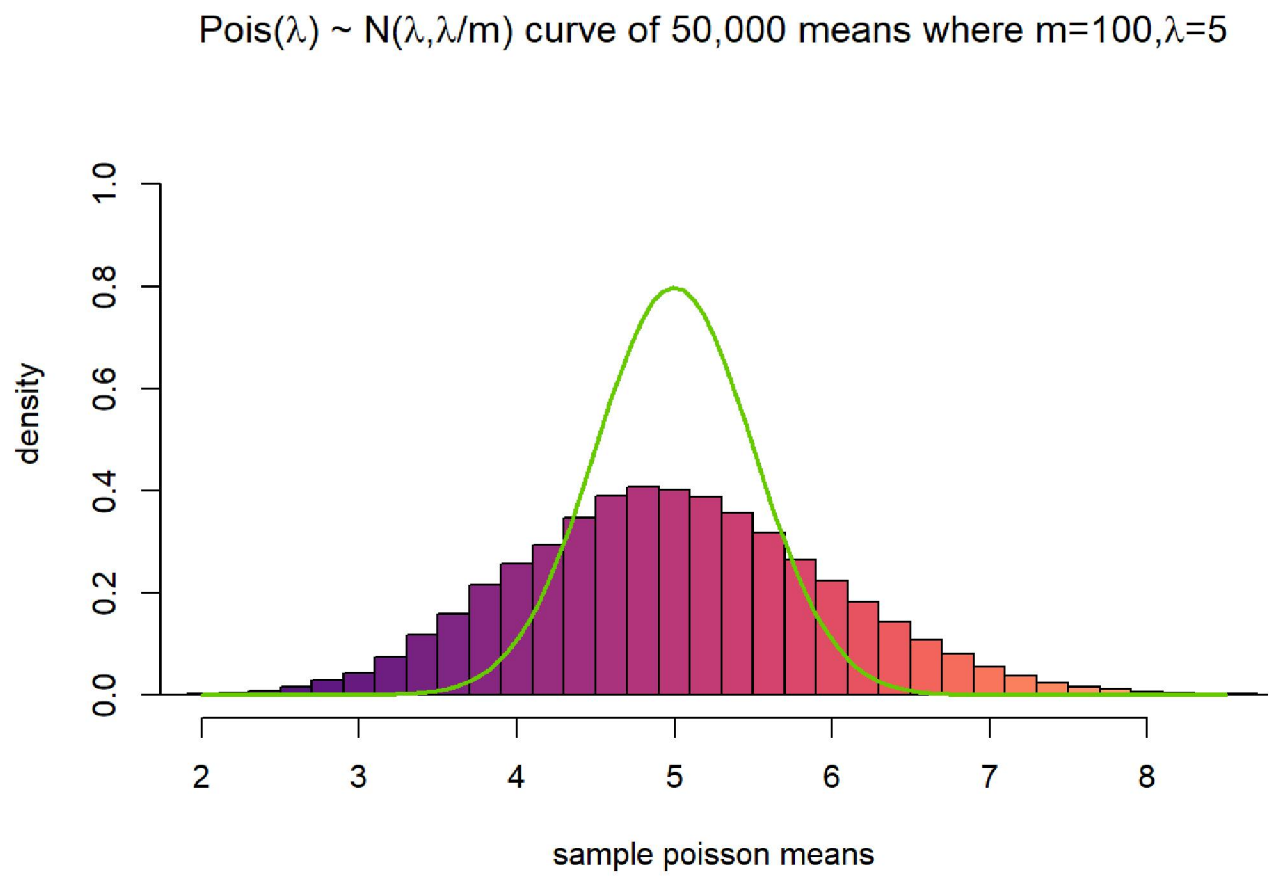


```
TF_values.30 <- rMeans30 >= (lambda.5+((2*lambda.5)/sqrt(m30)))

TFprob30 <- sum(TF_values.30)/M;print(TFprob30)

## [1] 0.03346667

NormProb30 <- dnorm((lambda.5+((2*lambda.5)/sqrt(m30))),mean=lambda.5,sd=(lambda.5/sqrt(m30)));print(NormProb30)

## [1] 0.05914414
```

FOR m = 100,λλ = 5

```
m100 = 100; lambda.5 = 5; M = 50000
```

```
ds100 <- matrix(replicate(M,(rpois(m100,lambda.5))),nrow=50000,ncol=5)

rMeans100 <- matrix(rowMeans(ds100),nrow=50000,ncol=1)

binlen <- c(-0.1,0.1,0.3,0.5,0.7,0.9,1.1,1.3,1.5,1.7,1.9,2.1,2.3,2.5,2.7,2.9,3.1,3.3,3.5,3.7,3.9,4.1,4.3,4.5,4.7,4.9,5.1,5.3,5.5,5.7,5.9,6.1,6.3,6.5,6.7,6.9,7.1,7.3,7.5,7.7,7.9,8.1,8.3,8.5,8.7,8.9,9.1,9.3,9.5,9.7,9.9,10.1,10.3,10.5,10.7,10.9,11.1,11.3,11.5,11.7,11.9)

hist(rMeans100,breaks=binlen ,main=expression(paste("Pois(",lambda,") ~ N(",lambda,",",lambda,"/m) curve of 50,000 means where m=100,",lambda,"=5"))  ,xlab="sample poisson means" ,xlim=c(2,8.5) ,ylab="density" ,ylim=c(0,1.1) ,freq = FALSE, col=magma(50))

x100 <- seq(-4,5, by = 0.1)

y100 <- dnorm(x100, mean = lambda.5, sd = lambda.5/sqrt(m100))

curve(dnorm(x, mean=lambda.5, sd=lambda.5/sqrt(m100)), add=TRUE, col="#66CC00", lwd=2)
```

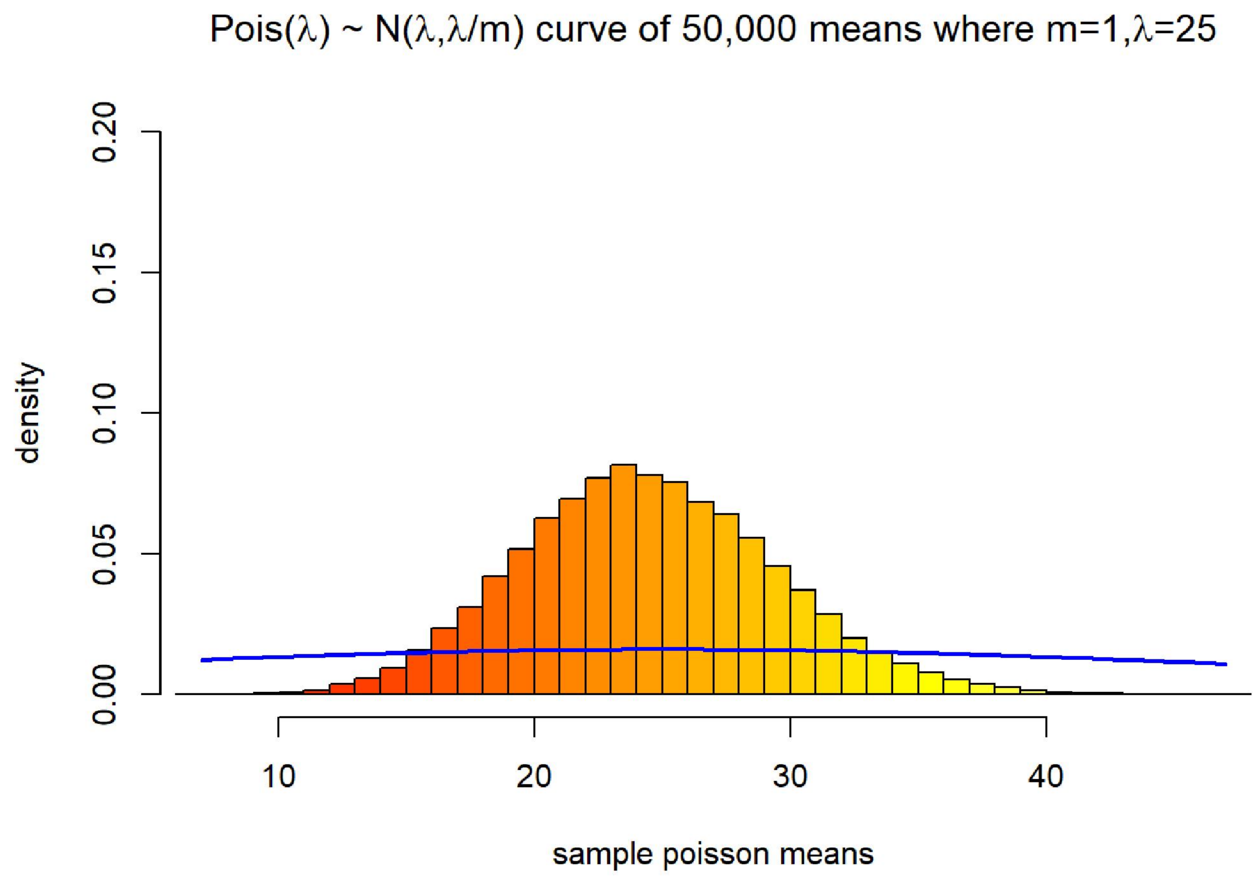


```
TF_values.100 <- rMeans100 >= (lambda.5+((2*lambda.5)/sqrt(m100)))

TFprob100 <- sum(TF_values.100)/M;print(TFprob100)

## [1] 0.18026

NormProb100 <- dnorm((lambda.5+((2*lambda.5)/sqrt(m100))),mean=lambda.5,sd=(lambda.5/sqrt(m100)));print(NormProb100)
```

```
## [1] 0.1079819
```

FOR m = 1,λλ = 25

```
m1 = 1; lambda.25 = 25; M = 50000

ds1 <- matrix(replicate(M,(rpois(m1,lambda.25))),nrow=50000,ncol=5)

rMeans1 <- matrix(rowMeans(ds1),nrow=50000,ncol=1)

hist(rMeans1, breaks=40,main=expression(paste("Pois(",lambda,") ~ N(",lambda,",",lambd
a,"/m) curve of 50,000 means where m=1,",lambda,"=25"))  ,xlab="sample poisson means"
,xlim=c(7,47) ,ylab="density" ,ylim=c(0,0.2) ,freq = FALSE, col=heat.colors(40))

x <- seq(-4,5, by = 0.1)

y1 <- dnorm(x, mean = lambda.25, sd = lambda.25/sqrt(m1))

curve(dnorm(x, mean=lambda.25, sd=lambda.25/sqrt(m1)), add=TRUE, col="blue", lwd=2)
```

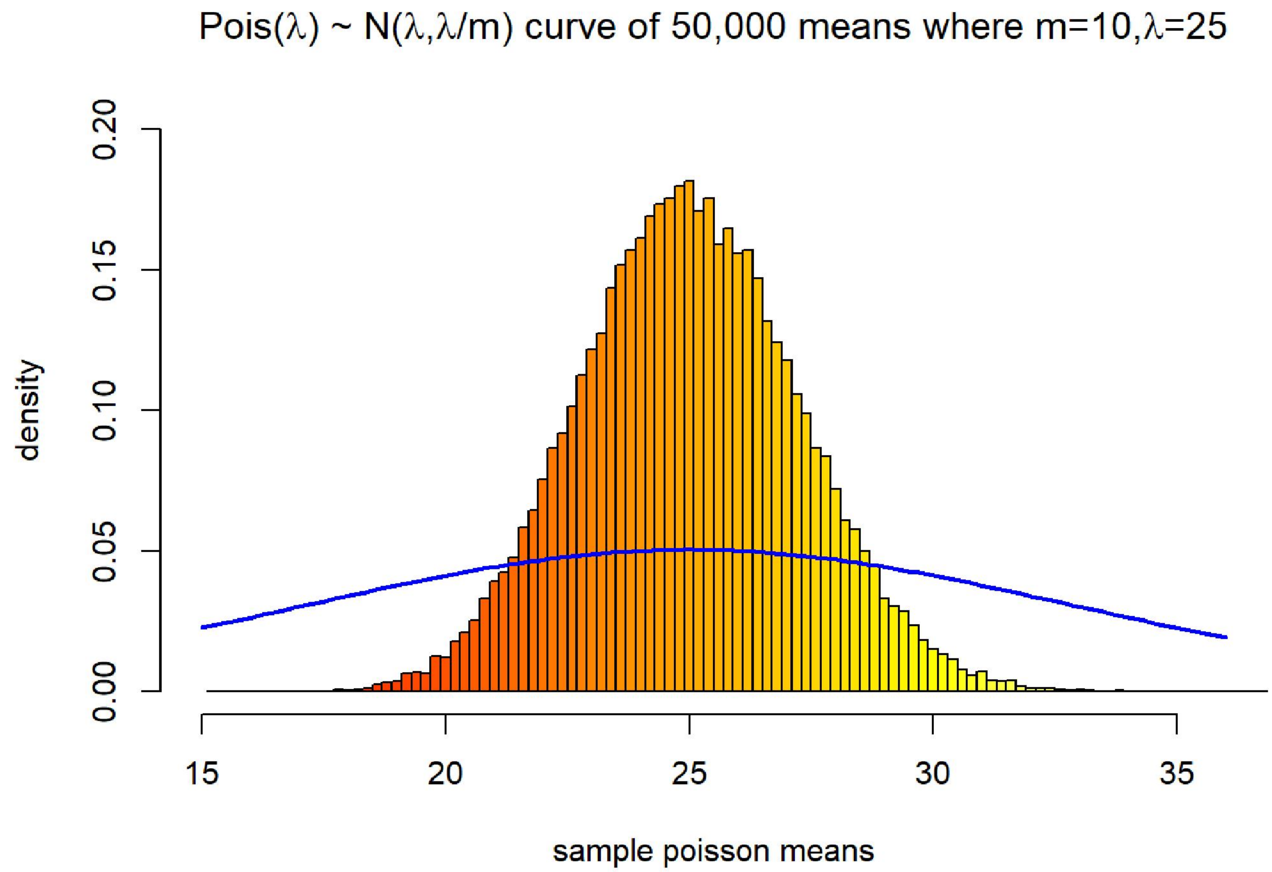


```
TF_values.1 <- rMeans1 >= (lambda.25+((2*lambda.25)/sqrt(m1)))

TFprob1 <- sum(TF_values.1)/M; print(TFprob1)

## [1] 0
```

```
NormProb1 <- dnorm((lambda.25+((2*lambda.25)/sqrt(m1))),mean=lambda.25,sd=(lambda.25/s
qrt(m1))); print(NormProb1)

## [1] 0.002159639
```

### FOR m = 10,λλ = 25

```
m10 = 10; lambda.25 = 25; M = 50000

ds10 <- matrix(replicate(M,(rpois(m10,lambda.25))),nrow=50000,ncol=5)

rMeans10 <- matrix(rowMeans(ds10),nrow=50000,ncol=1)

binlen <- c(15.1,15.3,15.5,15.7,15.9,16.1,16.3,16.5,16.7,16.9,17.1,17.3,17.5,17.7,17.9
,18.1,18.3,18.5,18.7,18.9,19.1,19.3,19.5,19.7,19.9,20.1,20.3,20.5,20.7,20.9,21.1,21.3,
21.5,21.7,21.9,22.1,22.3,22.5,22.7,22.9,23.1,23.3,23.5,23.7,23.9,24.1,24.3,24.5,24.7,2
4.9,25.1,25.3,25.5,25.7,25.9,26.1,26.3,26.5,26.7,26.9,27.1,27.3,27.5,27.7,27.9,28.1,28
.3,28.5,28.7,28.9,29.1,29.3,29.5,29.7,29.9,30.1,30.3,30.5,30.7,30.9,31.1,31.3,31.5,31.
7,31.9,32.1,32.3,32.5,32.7,32.9,33.1,33.3,33.5,33.7,33.9,34.1,34.3,34.5,34.7,34.9,35.1
,35.3,35.5,35.7,35.9,36.1,36.3,36.5,36.7,36.9,37.1,37.3)

hist(rMeans10,breaks=binlen ,main=expression(paste("Pois(",lambda,") ~ N(",lambda,",",
lambda,"/m) curve of 50,000 means where m=10,",lambda,"=25"))  ,xlab="sample poisson m
eans" ,xlim=c(15,36) ,ylab="density" ,ylim=c(0,0.2) ,freq = FALSE, col=heat.colors(100
))

x <- seq(-4,5, by = 0.1)

y10 <- dnorm(x, mean = lambda.25, sd = lambda.25/sqrt(m10))

curve(dnorm(x, mean=lambda.25, sd=lambda.25/sqrt(m10)), add=TRUE, col="blue", lwd=2)
```

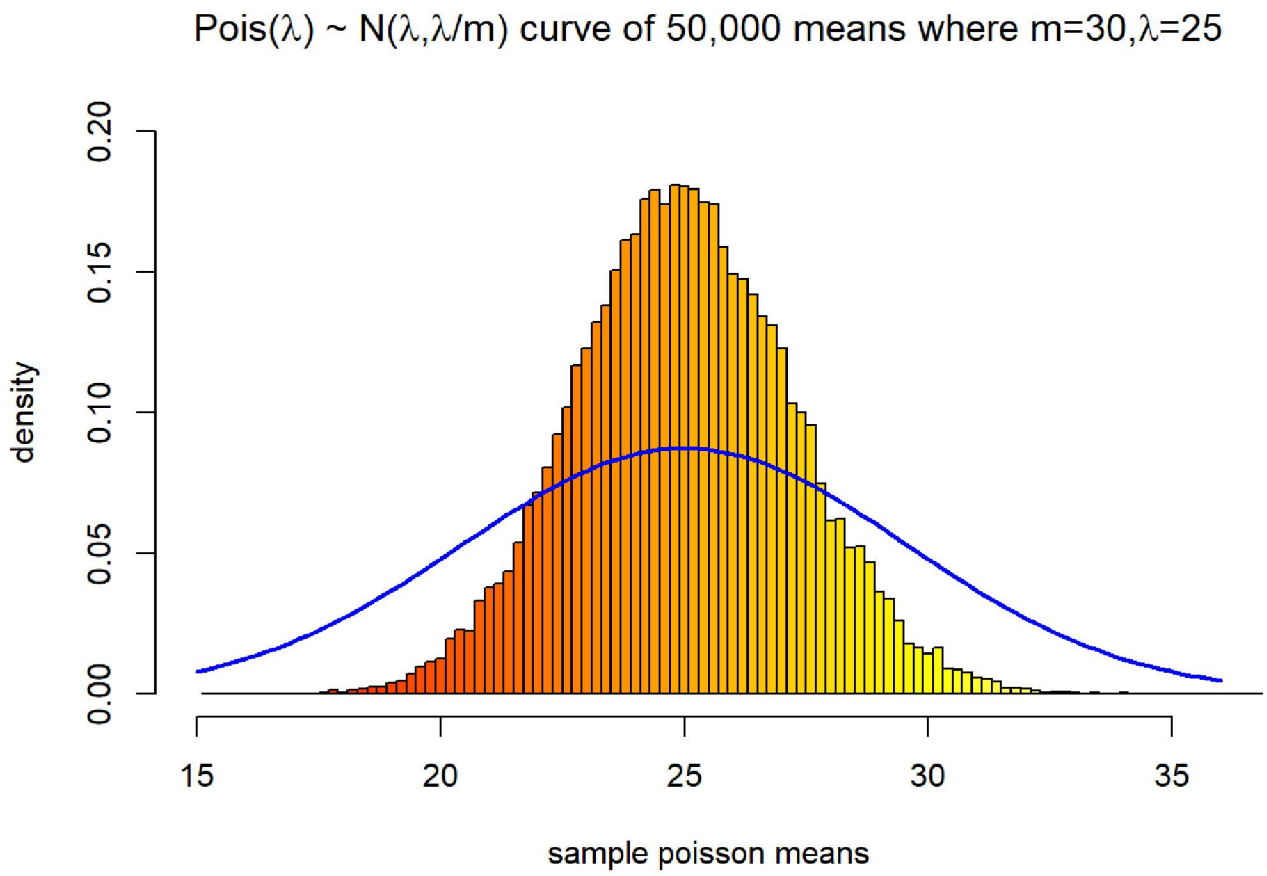


```
TF_values.10 <- rMeans10 >= (lambda.25+((2*lambda.25)/sqrt(m10)))

TFprob10 <- sum(TF_values.10)/M; print(TFprob10)

## [1] 0

NormProb10 <- dnorm((lambda.25+((2*lambda.25)/sqrt(m10))),mean=lambda.25,sd=(lambda.25/sqrt(m10))); print(NormProb10)

## [1] 0.006829377
```

FOR m = 30,λλ = 25

```
m30 = 30; lambda.25 = 25; M = 30000

ds30 <- matrix(replicate(M,(rpois(m30,lambda.25))),nrow=30000,ncol=5)

rMeans30 <- matrix(rowMeans(ds30),nrow=30000,ncol=1)

binlen <- c(15.1,15.3,15.5,15.7,15.9,16.1,16.3,16.5,16.7,16.9,17.1,17.3,17.5,17.7,17.9,18.1,18.3,18.5,18.7,18.9,19.1,19.3,19.5,19.7,19.9,20.1,20.3,20.5,20.7,20.9,21.1,21.3,21.5,21.7,21.9,22.1,22.3,22.5,22.7,22.9,23.1,23.3,23.5,23.7,23.9,24.1,24.3,24.5,24.7,24.9,25.1,25.3,25.5,25.7,25.9,26.1,26.3,26.5,26.7,26.9,27.1,27.3,27.5,27.7,27.9,28.1,28.3,28.5,28.7,28.9,29.1,29.3,29.5,29.7,29.9,30.1,30.3,30.5,30.7,30.9,31.1,31.3,31.5,31.7,31.9,32.1,32.3,32.5,32.7,32.9,33.1,33.3,33.5,33.7,33.9,34.1,34.3,34.5,34.7,34.9,35.1,35.3,35.5,35.7,35.9,36.1,36.3,36.5,36.7,36.9,37.1,37.3)
```

```
hist(rMeans30,breaks=binlen ,main=expression(paste("Pois(",lambda,") ~ N(",lambda,",",lambda,"/m) curve of 50,000 means where m=30,",lambda,"=25")) ,xlab="sample poisson means" ,xlim=c(15,36)

     ,ylab="density" ,ylim=c(0,0.2) ,freq = FALSE, col=heat.colors(100))

x30 <- seq(-4,5, by = 0.1)

y30 <- dnorm(x30, mean = lambda.25, sd = lambda.25/sqrt(m30))

curve(dnorm(x, mean=lambda.25, sd=lambda.25/sqrt(m30)), add=TRUE, col="blue", lwd=2)
```

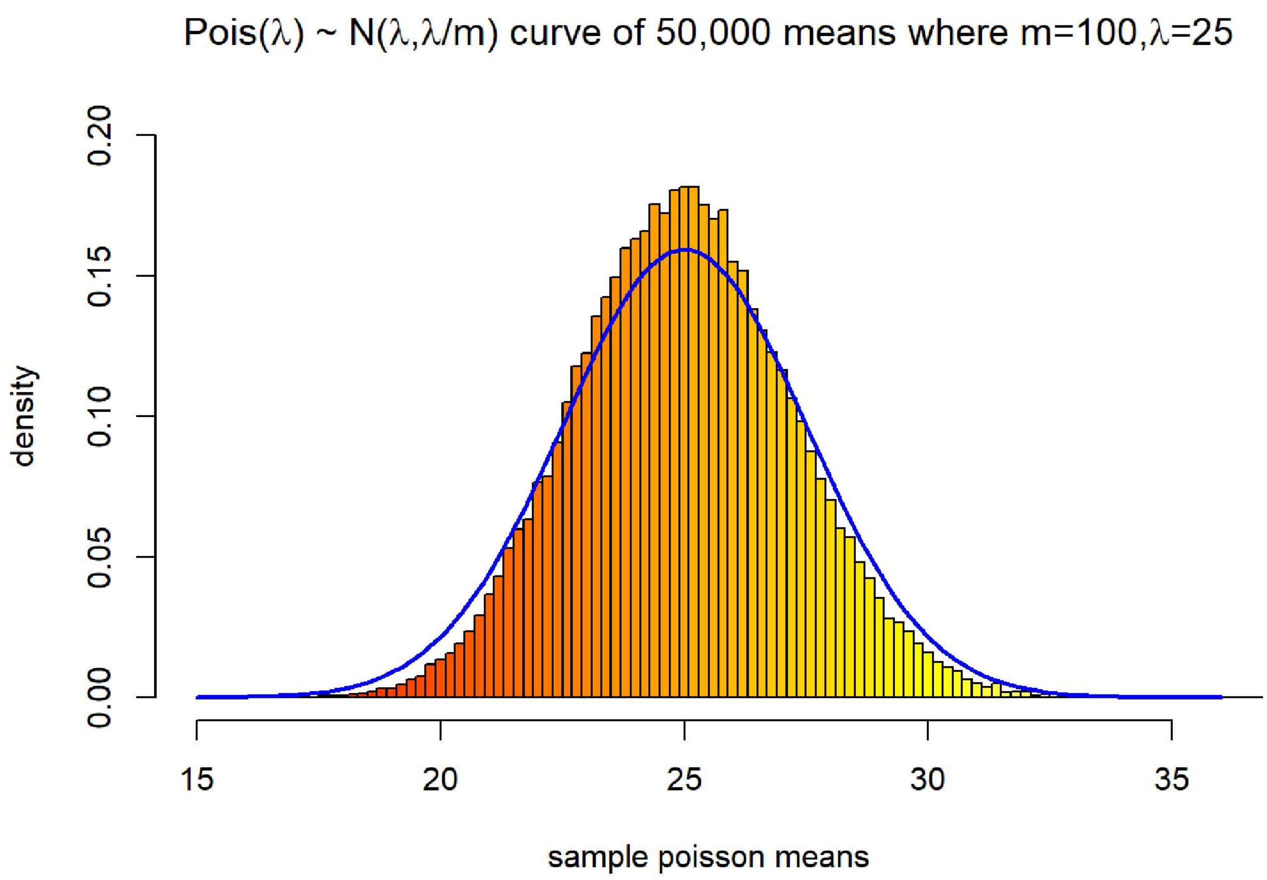


```
TF_values.30 <- rMeans30 >= (lambda.25+((2*lambda.25)/sqrt(m30)))

TFprob30 <- sum(TF_values.30)/M; print(TFprob30)

## [1] 6.666667e-05

NormProb30 <- dnorm((lambda.25+((2*lambda.25)/sqrt(m30))),mean=lambda.25,sd=(lambda.25/sqrt(m30)));print(NormProb30)

## [1] 0.01182883
```

FOR m = 100,λλ = 25

```
m100 = 100; lambda.25 = 25; M = 50000
```

```
ds100 <- matrix(replicate(M,(rpois(m100,lambda.25))),nrow=50000,ncol=5)

rMeans100 <- matrix(rowMeans(ds100),nrow=50000,ncol=1)

binlen <- c(15.1,15.3,15.5,15.7,15.9,16.1,16.3,16.5,16.7,16.9,17.1,17.3,17.5,17.7,17.9,18.1,18.3,18.5,18.7,18.9,19.1,19.3,19.5,19.7,19.9,20.1,20.3,20.5,20.7,20.9,21.1,21.3,21.5,21.7,21.9,22.1,22.3,22.5,22.7,22.9,23.1,23.3,23.5,23.7,23.9,24.1,24.3,24.5,24.7,24.9,25.1,25.3,25.5,25.7,25.9,26.1,26.3,26.5,26.7,26.9,27.1,27.3,27.5,27.7,27.9,28.1,28.3,28.5,28.7,28.9,29.1,29.3,29.5,29.7,29.9,30.1,30.3,30.5,30.7,30.9,31.1,31.3,31.5,31.7,31.9,32.1,32.3,32.5,32.7,32.9,33.1,33.3,33.5,33.7,33.9,34.1,34.3,34.5,34.7,34.9,35.1,35.3,35.5,35.7,35.9,36.1,36.3,36.5,36.7,36.9,37.1,37.3)

hist(rMeans100,breaks=binlen ,main=expression(paste("Pois(",lambda,") ~ N(",lambda,",",lambda,"/m) curve of 50,000 means where m=100,",lambda,"=25"))  ,xlab="sample poisson means" ,xlim=c(15,36) ,ylab="density" ,ylim=c(0,0.2) ,freq = FALSE, col=heat.colors(100))

x100 <- seq(-4,5, by = 0.1)

y100 <- dnorm(x100, mean = lambda.25, sd = lambda.25/sqrt(m100))

curve(dnorm(x, mean=lambda.25, sd=lambda.25/sqrt(m100)), add=TRUE, col="blue", lwd=2)
```



```
TF_values.100 <- rMeans100 >= (lambda.25+((2*lambda.25)/sqrt(m100)))

TFprob100 <- sum(TF_values.100)/M; print(TFprob100)

## [1] 0.01628
```

```
NormProb100 <- dnorm((lambda.25+((2*lambda.25)/sqrt(m100))),mean=lambda.25,sd=(lambda.
25/sqrt(m100)));print(NormProb100)

## [1] 0.02159639
```

**Discuss how these plots and probabilities can help someone understand convergence in distribution.**

It is interesting to see a discrete poisson distribution converging toward a continuous normal on a single graph. While poisson(λλ) is observed at its smallest evaluated values, there are significantly less individually distinct values a mean can take on in the same interval length of the largest observed poisson(λλ) value. For example, at λλ = 1 from (0,3), a mean can be observed at the integers 0,1,2,3. At λλ = 100, observed mean values on that same range include rational numbers whose means have intervals of 0.2; i.e. 0.0,0.2,0.4,0.6,…etc. We can interpret that as having less variability at λλ = 1 as opposed to λλ = 100. Even at lower λλ values, we can observe a relatively decent fit of the normality curve with small sample sizes. The normality curve fits λλ = 1 best betwen m=10 and m=30. It appears that as m=100, the λλ = 1 model show more variability than the curve. The inverse is usually true when λλ is higher and m is lower.

When poisson(λλ) reaches λλ = 5, we have a beautifully shaped histogram that looks far more closer to a normally-shaped curve than λλ = 1. At λλ = 5, the sample size of 1 is too small to fit this model well. The curve is too horizontal and illustrates more variability than the poisson(λλ). We will need more sample. It is not until m=30 that we get our closest curve yet to the histogram.

Finally, poisson(λλ)obtains our highest observed λλ at λλ = 25 and the m has to be big in order for the poisson(λλ) to converge toward the normal(λλ,λλ/m). The line is too horizontal at m=1 and m=10. At m=30 the curve begins to takes shape and by m=100, the curve is bending just beneath the histogram.

This is an interesting result because of the implications for finding various statistical parameters for a poisson distribution from a normal distribution when that poisson distributionconverges to a normal.

**Why do you think the large-sample approximation works better for larger λλ values?**

With smaller-sample approximation there are fewer observations and thus more oppurtunity for random values to appear distant from the constant to which the pois(λλ) is expected to converge to. In a small sample, These values will carry more weight as it pertains to variability. A large enough sample and a large enough centering mass of observations about the constant will have a sort of diluting or drowning out effect on the values further from the converging constant. In turn, these larger samplings will fit the pois(λλ) more closely.