stocahstic homework4

kallakuri radhakrishna

15/06/2020

example 1

a

Generate a sequence of 100 candies dropped from the machine.

```
candie_types <- c("red", "orange", "yellow", "green")
candie_prob <- c(0.25,0.25,0.25,0.25)
s <- sample(candie_types, 100, replace = TRUE, prob = candie_prob)
s
```

```
##   [1] "red"    "yellow" "yellow" "red"    "red"    "yellow" "yellow"
##   [8] "orange" "red"    "red"    "red"    "red"    "red"    "red"
##  [15] "orange" "orange" "red"    "orange" "green"  "red"    "yellow"
##  [22] "yellow" "yellow" "red"    "yellow" "green"  "yellow" "red"
##  [29] "green"  "orange" "orange" "red"    "green"  "red"    "yellow"
##  [36] "red"    "yellow" "orange" "green"  "green"  "yellow" "green"
##  [43] "red"    "yellow" "yellow" "yellow" "yellow" "orange" "green"
##  [50] "green"  "green"  "red"    "orange" "green"  "red"    "red"
##  [57] "yellow" "orange" "red"    "red"    "red"    "red"    "green"
##  [64] "yellow" "green"  "orange" "orange" "yellow" "red"    "red"
##  [71] "red"    "orange" "orange" "orange" "orange" "red"    "yellow"
##  [78] "green"  "yellow" "orange" "yellow" "red"    "green"  "yellow"
##  [85] "red"    "red"    "yellow" "red"    "green"  "orange" "red"
##  [92] "red"    "yellow" "red"    "red"    "green"  "yellow" "orange"
##  [99] "red"    "green"
```

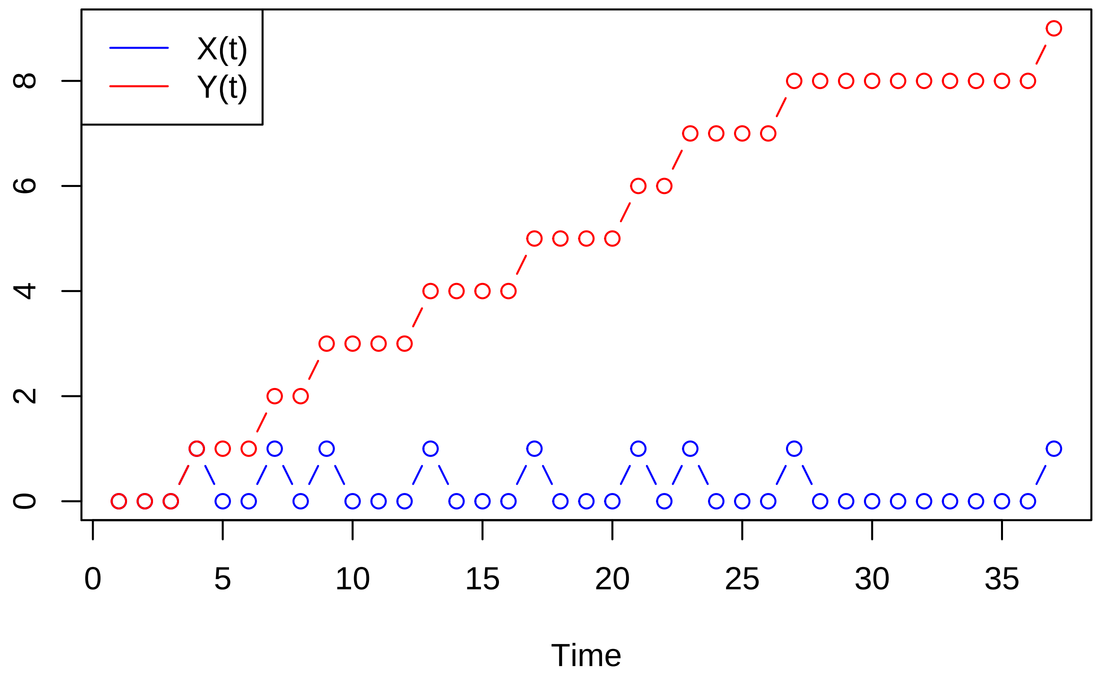
b

Plot the total number of red candies as an arrival process.

```
length(which(s=="red"))
```

```
## [1] 37
```

```
p <- .25
s1 <- sample(c(0, 1),length(which(s=="red")) , replace = T, prob = c(1-p, p))
ap <- cumsum(s1)
plot.ts(cbind(s1, ap), plot.type = "single", type = "b", col = c("blue", "red"), ylab = "")
legend("topleft", c("X(t)", "Y(t)"), col = c("blue", "red"), lt = 1 )
```



c

From the obtained process calculate the epochs and interarrival times.

```
epochs <- function( process ) {
  epochs <- c()
  for (i in 1:max(process)) epochs <- c(epochs, which(process == i)[1]) # the i-th arrival time
  epochs
}
epochs(ap)
```

```
## [1]  4  7  9 13 17 21 23 27 37
```

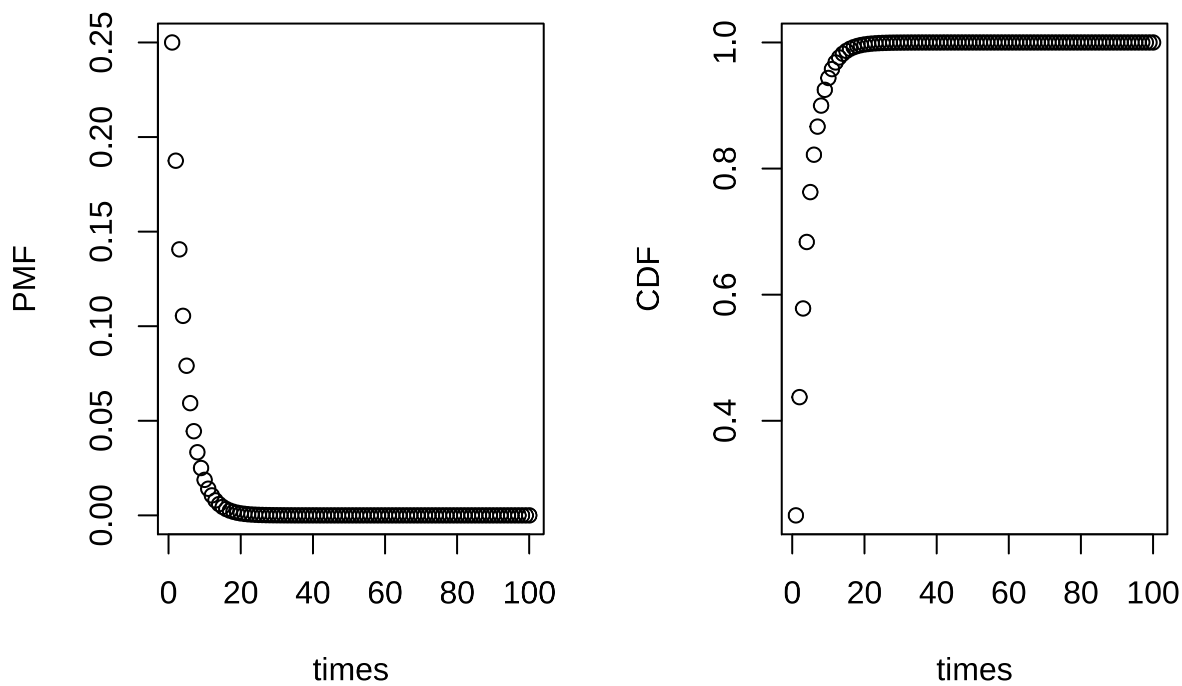
```
interarrival <- function( process ) {
  inttimes <- c()
  ep <- epochs(process) # calculate epochs
  diff(c(0, ep)) # interarrival times are differences between epochs
  # zero epoch is 0
}
interarrival(ap)
```

```
## [1]  4  3  2  4  4  4  2  4 10
```

d

Let T be the number of trials needed to get a red candy. Find the distribution of T.

```r
times <- 1:100
par(mfrow = c(1, 2))
plot(times, dgeom(times-1, prob = 0.25), ylab = "PMF")
plot(times, pgeom(times-1, prob = 0.25), ylab = "CDF")
```



e

What is the probability that in a sequence of 10 consecutive candies sold there is no red candy? in sequence of 10 consecutive candies there is no candie so we take number of failures as 10. The answer is 0.014.

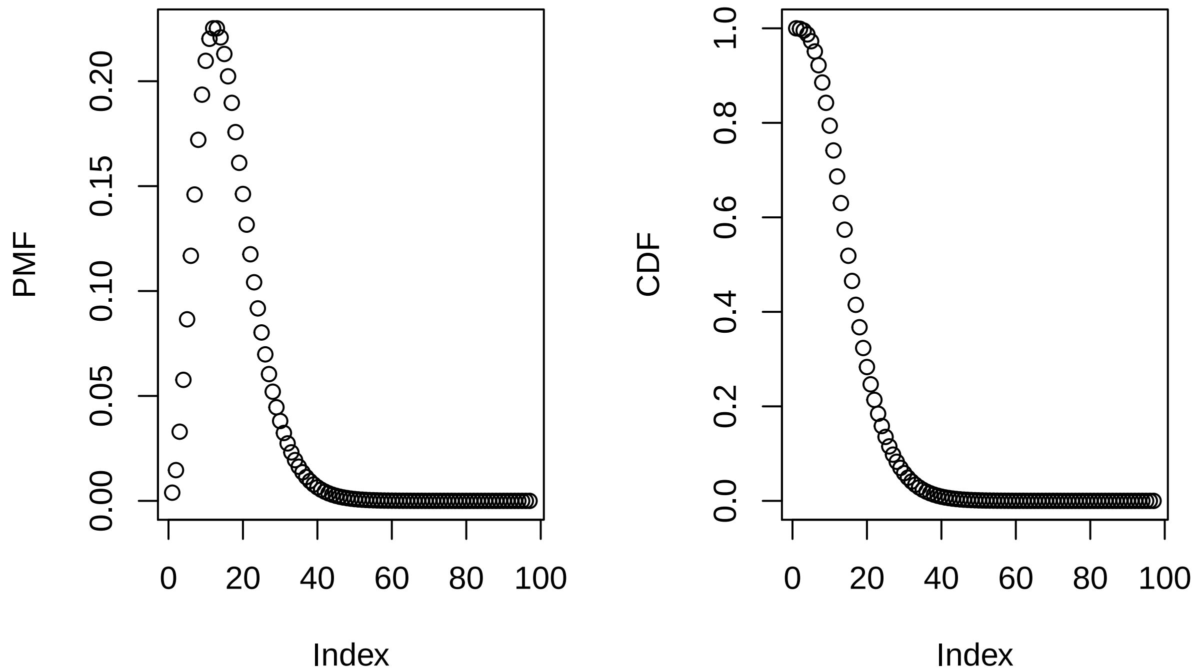
```r
dgeom(10,0.25)
```

```
## [1] 0.01407838
```

f

A child would like to get 4 red candies. Let S4 be the number of candies he will have to buy. What is the distribution of S4?

```r
n <- 4:100
k <- 4
par(mfrow = c(1, 2))
plot( dbinom(k, size = n, prob = .25), ylab = "PMF")
plot( pbinom(k, size = n, prob = .25), ylab = "CDF")
```



g

What is the probability that the child will have to buy at most 10 candies to get four red ones? answer is 0.145

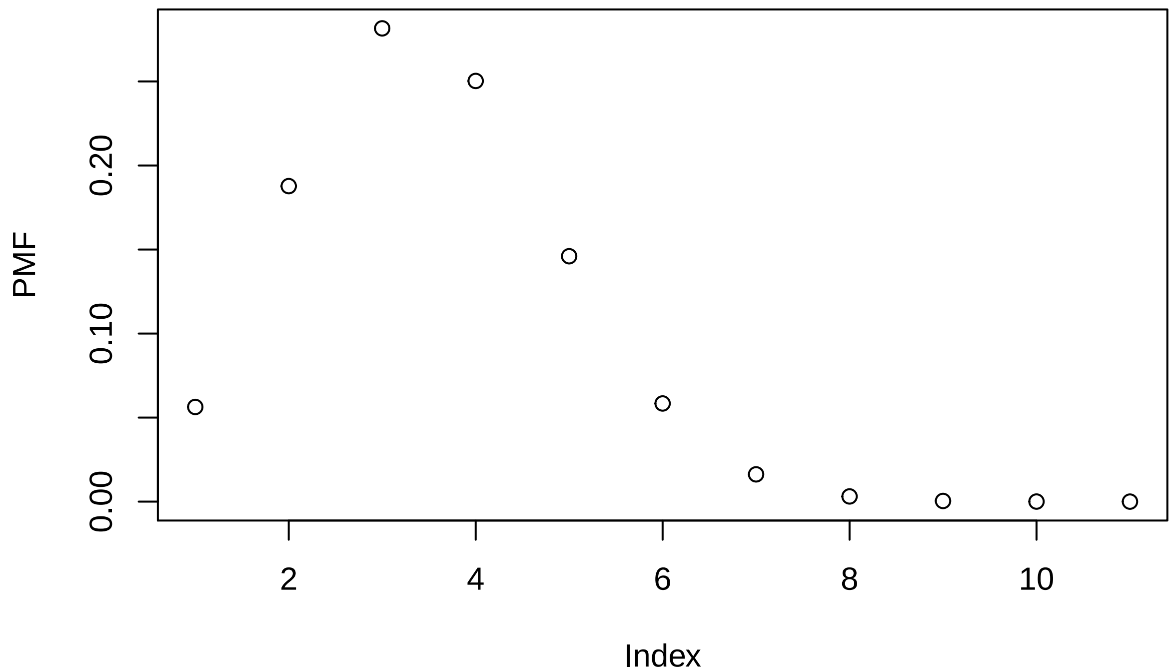
```r
dbinom(4,10,0.25)
```

```
## [1] 0.145998
```

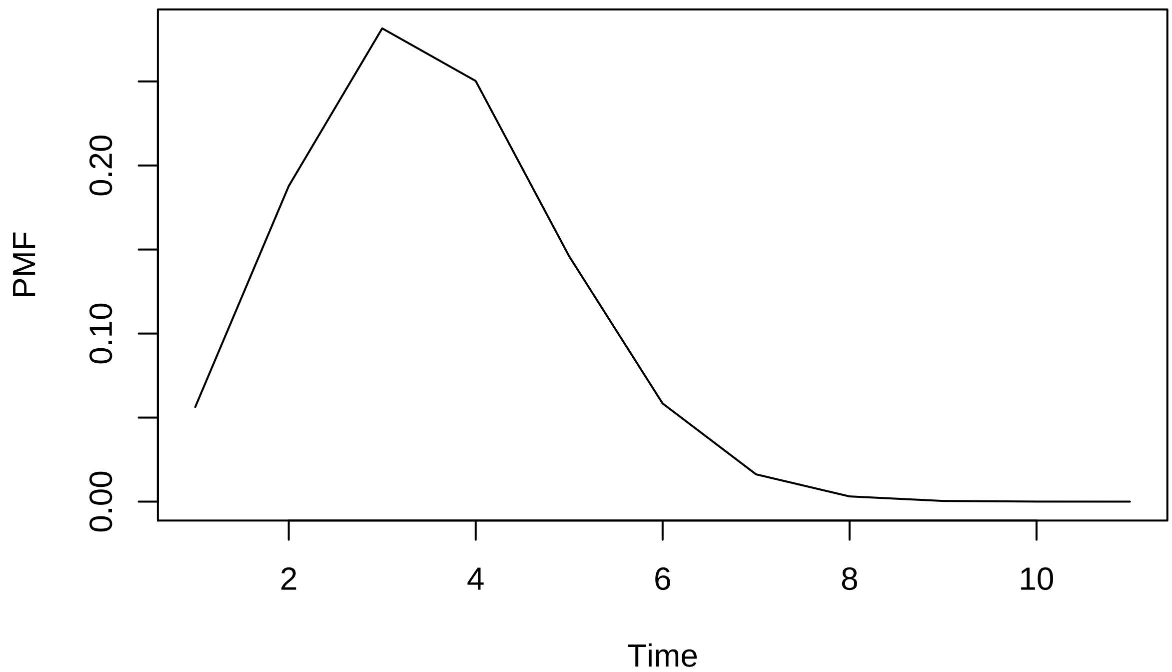
h

Let N(10) be the total number of red candies dropped out in 10 trials. Find the distribution of N(10). Plot its probability mass function.

```r
k1 <- 0:10
 plot( dbinom(k1, size = 10, prob = .25), ylab = "PMF")
```




```
plot.ts( dbinom(k1, size = 10, prob = .25), ylab = "PMF")
```



i

What is the probability that N(10) > 4? Compare this probability with the result of g). Explain. The probability of getting 4 red candies is 0.145 which is higher than that of getting more than 4 red candies which is 0.0781. As this is a binomial distribution after 4 the probability values take a fall as seen from the previous graph.

```
1 - pbinom(4, size = 10, prob = .25)
```

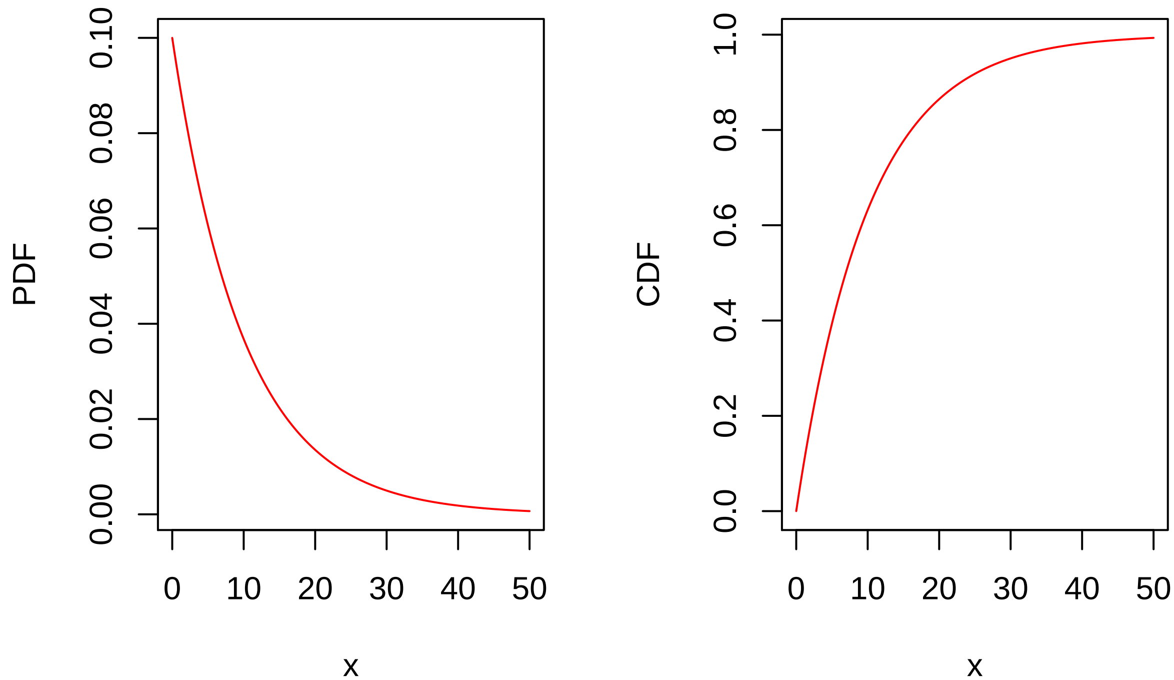
```
## [1] 0.07812691
```

example 2

a

Density function and range is between 0 to 0.1

```
lamb <- 1/10 # rate lambda
x <- 0:1000/20 # x-points
f <- dexp(x, rate = lamb) # y-points
par(mfrow = c(1, 2))
plot(x, f, type = "l", col = "red", xlab = "x", ylab = "PDF")
plot(x, pexp(x, rate=lamb), type = "l", col = "red", xlab = "x", ylab = "CDF")
```



b

Calculate the probability that it will pass less than five years between consecutive financial crises. answer is 0.393

```
pexp(5, rate=lamb)
```

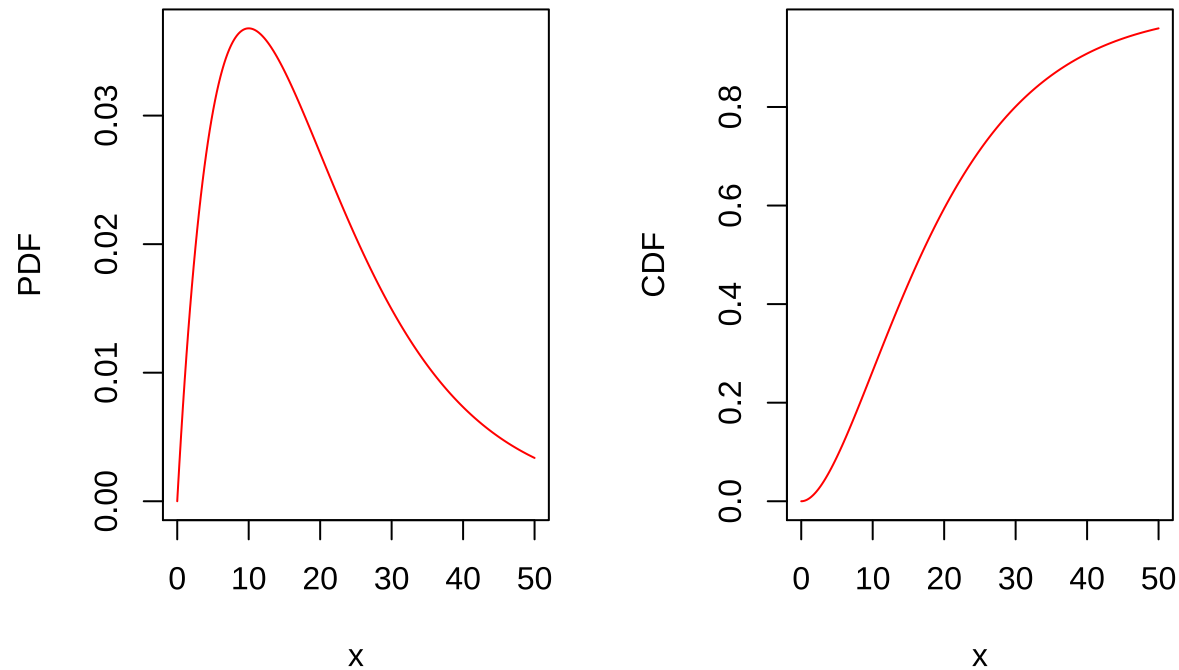
```
## [1] 0.3934693
```

c

Let S2 denote the time until the second financial crisis. Find its distribution and plot the corresponding density function.

```
# erlang dist
lamb <- 1/10; n <- 2 # number of arrivals
x <- 0:1000/20
# distribution function

f1 <- dgamma(x, shape = n, rate = lamb)
par(mfrow = c(1, 2))
plot(x, f1, type = "l", col = "red", xlab = "x", ylab = "PDF")
plot(x, pgamma(x, shape = n, rate = lamb), type = "l", col = "red", xlab = "x", ylab = "CDF")
```



d

Calculate and interpret the 0.8 quantile for S2 Quantiles are statistics that describe various subdivisions of a frequency distribution into equal proportions. The simplest division that can be envisioned is into two equal halves and the quantile that does this, the median value of the variate, is used also as a measure of central tendency for the distribution. 80% of the vaues of second financial crisis is less than 29.94.

```
qgamma(0.8, shape = n, rate = lamb)
```

```
## [1] 29.94308
```

e

What is the probability that someone who starts a business will experience more than three financial crises in the next 20 years period? answer is 0.67

```
lamb <- 1/10; n <- 3
 x <- 20

1- pgamma(x, shape = n, rate = lamb)
```

```
## [1] 0.6766764
```

example 3

a

Model the costumer service process as an M/M/1 queue and describe its parameters.

```
# incoming rate
lam <- 3
# service rate
mu <- 6
# traffic intensity
rho <- lam/mu


# expected number of customers in the system
L <- lam/(mu-lam)
# average time in the system
W <- 1/(mu-lam)
# average time in the queue
Wq <- lam/(mu*(mu-lam))
# average number of customers waiting
Lq <- lam^2/(mu*(mu-lam))
L
```

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

```
W
```

```
## [1] 0.3333333
```

```
Wq
```

```
## [1] 0.1666667
```

```
Lq
```

```
## [1] 0.5
```

b

Calculate the long term distribution for the number of requests in the process.

```
k <- 0:10
# limit distribution
pi <- (1-rho)*rho^k
pi
```

```
##  [1] 0.5000000000 0.2500000000 0.1250000000 0.0625000000 0.0312500000
##  [6] 0.0156250000 0.0078125000 0.0039062500 0.0019531250 0.0009765625
## [11] 0.0004882812
```

c

Calculate the expected number of requests waiting in the queue. Answer is 0.5

```
# average number of customers waiting
Lq <- lam^2/(mu*(mu-lam))
Lq
```

```
## [1] 0.5
```

d

Due to the a new user interface, the request rate increases to 4 per hour. How much does the expected queue length increase? answer is 0.833

```
lam1 <- 4
Lq1  <- lam1^2/(mu*(mu-lam1))
 Lq1
```

```
## [1] 1.333333
```

```
 increase <- Lq1 -Lq
increase
```

```
## [1] 0.8333333
```

e

To ensure smooth service, the support office employs a new employee. What is the expected queue length now? answer is 0.533

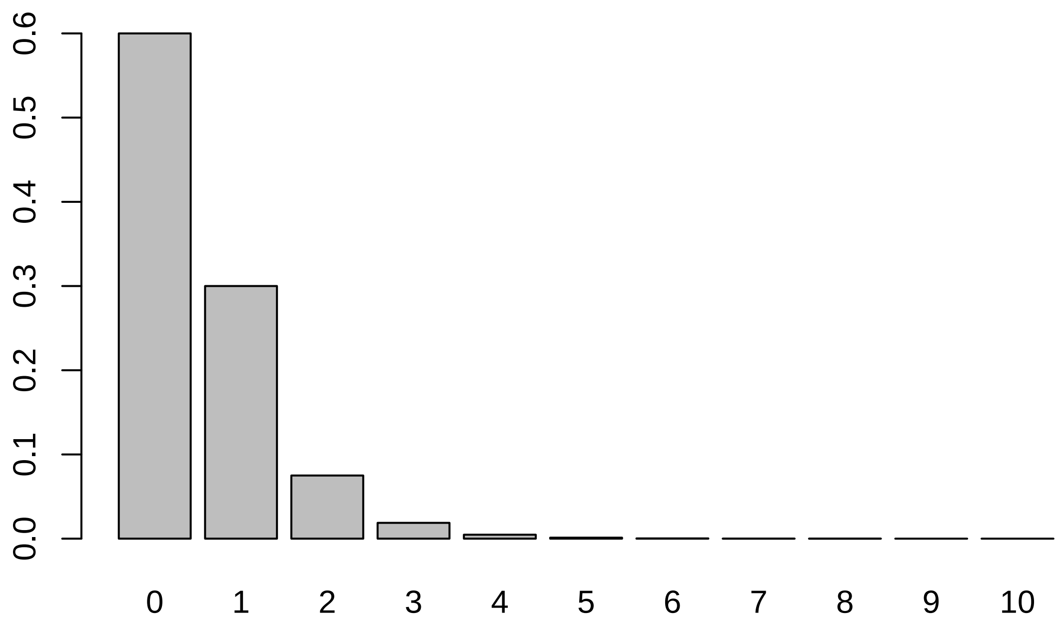
```
# number of servers
s <- 2
# incoming rate
lam <- 3
# service rate
mu <- 6
k <- 0:10
# traffic intensity
rho <- lam/mu
# indicator for k <= s
kls <- as.numeric(k <= s)
# indicator for k > s
kgs <- as.numeric(k > s)
# limit distribution
# pi = pp*pi0
pp <- kls*rho^k/factorial(k) + kgs*rho^k/(factorial(s)*s^(k-s))

pi0 <- 1/(sum(pp[1:s]) + rho^s/factorial(s)*s/(s-rho))

show(pi0)
```

```
## [1] 0.6
```

```
# the limit distribution
pi <- pp*pi0
barplot(pi, names = k)
```




```
xi <- rho/s
# average number of customers waiting
Lq <- rho^(s+1)/(factorial(s)*s)/(1-xi)^2*pi0
# average time in the queue
Wq <- Lq/lam
# average time in the system
W <- Wq + 1/mu
# expected number of customers in the system
L <- lam*W
L
```

```
## [1] 0.5333333
```

example 4

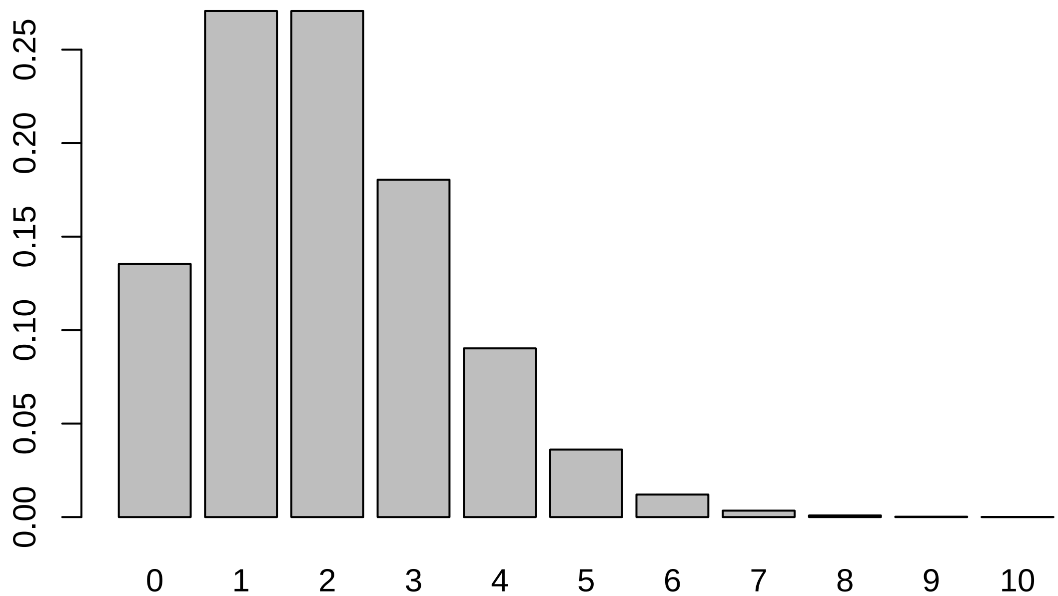
(a) Model the computer lab occupancy as a no-wait queue. Determine its parameters.

```
# number of servers
s <- 10
# incoming rate
lam <- 1
# service rate
mu <- 1/2
# capacity
K <- s
# traffic intensity
rho <- lam/mu
```

```r
# limit distribution
k <- 0:K
# limit distribution
# pi = pp*pi0
pp <- rho^k/factorial(k)
pi0 <- 1/sum(pp)
show(pi0)
```

```
## [1] 0.1353364
```

```r
# the limit distribution
pi <- pp*pi0
barplot(pi, names = k)
```




```r
# average entering rate
lame <- lam*(1-pi[s+1])
# expected number of customers in the system
L <- lame/mu
# average time in the system is equal to the service time
W <- 1/mu
show(c(lame,L,W))
```

```
## [1] 0.9999618 1.9999236 2.0000000
```

b

Calculate the probability that all computers are occupied.

answer is 3.819017e-05

```
p1 <- (rho)^s/factorial(s)
 pp <- rho^k/factorial(k)
pi0 <- 1/sum(pp)
p <- p1*pi0
p
```

```
## [1] 3.819017e-05
```

c

What is the probability that more then two computers are available? Same as occupied computers less tan equal to 2, thus answer is 0.4060092.

```
sum(pi[1:2])
```

```
## [1] 0.4060092
```

d

Calculate the expected number of occupied computers. answer is 1.999

```
# average entering rate
lame <- lam*(1-pi[s+1])
show(lame)
```

```
## [1] 0.9999618
```

```
# expected number of occupied computers customers
L <- lame/mu
# average time in the system is equal to the service time
W <- 1/mu
show(c(L, W))
```

```
## [1] 1.999924 2.000000
```