****

#Part1) Binomial distribution

```
#a)
p <- 40/100
n <- 5
Code section 1.a:
pmf <- dbinom(0:n,size = n,prob = p)
pmf
cdf <- pbinom(0:n,size = n,prob = p)
cdf
```

#for plotting pmf

```
heights <- dbinom(0:n,size = n,prob = p)
plot(0:n,heights,type = "h",main ="Probablity distribution for perfect score",
    xlab = "Number of attempts for a perfect score",ylab  = "PMF" )
points(0:n, heights,pch=16)
```

#for plotting cdf

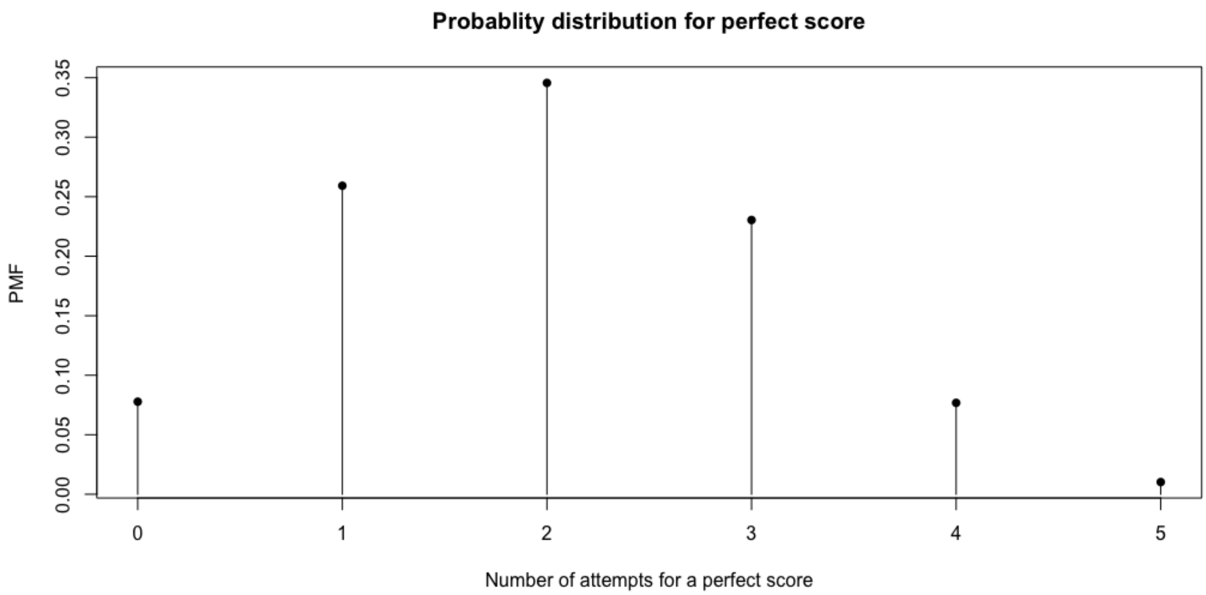
```
#inserting first values of 0 for corresponding F(x), x<0
cdf <- c(0,cdf)
cdf

cdfplot <- stepfun(0:n,cdf)
plot(cdfplot,verticals = FALSE,pch = 16, main = "CDF plot",
    xlab = "Number of attemts for a perfect scores",ylab = "CDF" )
```

Console secton 1.a:

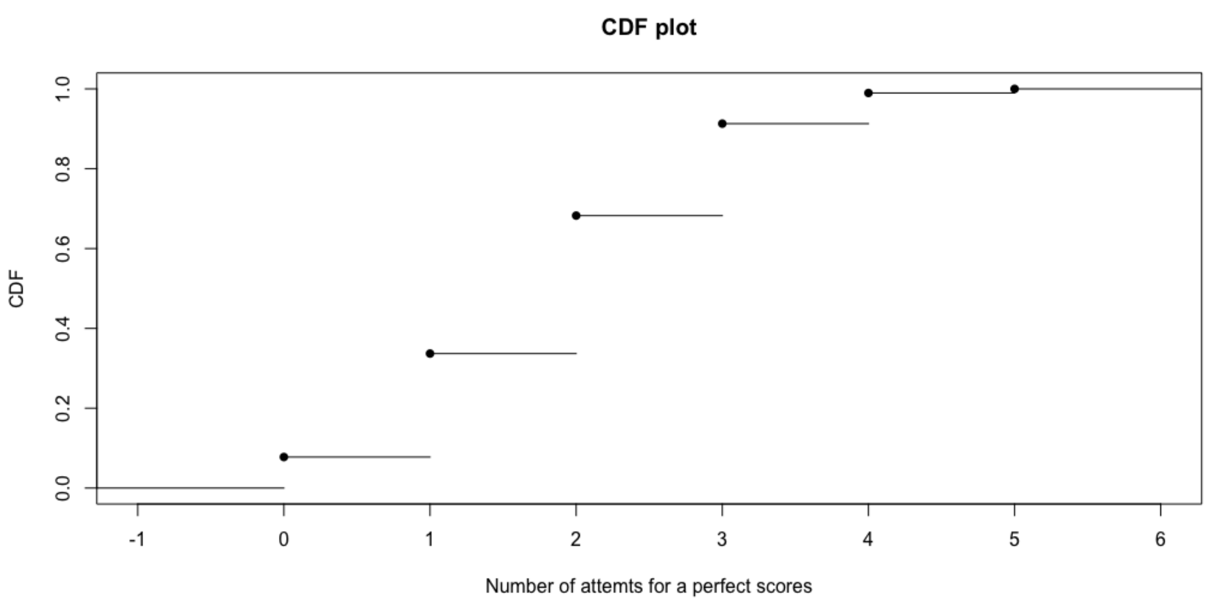
```
> p <- 40/100
> n <- 5
> pmf <- dbinom(0:n,size = n,prob = p)
> pmf
[1] 0.07776 0.25920 0.34560 0.23040 0.07680 0.01024
> cdf <- pbinom(0:n,size = n,prob = p)
> cdf
[1] 0.07776 0.33696 0.68256 0.91296 0.98976 1.00000
> #for plotting pmf
> heights <- dbinom(0:n,size = n,prob = p)
> plot(0:n,heights,type = "h",main ="Probablity distribution for perfect score",
+      xlab = "Number of attempts for a perfect score",ylab  = "PMF" )
> points(0:n, heights,pch=16)
> #for plotting pmf
> #inserting first values of 0 for corresponding F(x), x<0
> cdf <- c(0,cdf)
```

```
> cdf
[1] 0.00000 0.07776 0.33696 0.68256 0.91296 0.98976 1.00000
> cdfplot <- stepfun(0:n,cdf)
> plot(cdfplot,verticals = FALSE,pch = 16, main = "CDF plot",
+      xlab = "Number of attemts for a perfect scores",ylab = "CDF" )
```

Plotting section 1.a:
Plotting for pmf:



Plotting for cdf:



1.b

Code section 1.b

```
#b
#finding perfect scores in exactly 2 out of 5 attempts
#USING R FUNCTION
dbinom(2,size = n,prob = p)
#using bionomial coeficients
choose(5,2)*p^2*(1-p)^3
```

Console section 1.b

```
> #b)
> #finding perfect scores in exactly 2 out of 5 attempts.
> #USING R FUNCTION
> dbinom(2,size = n,prob = p)
[1] 0.3456
> #using bionomial coeficients
> choose(5,2)*p^2*(1-p)^3
[1] 0.3456
```

Code section 1.c

```
#c)
#finding probability for perfect scores in at least 2 attempts out of 5
# finding values fx(2)+fx(3)+fx(4) +fx(5) which denotes at least two perfect scores out of 5
attempts.

atleast_two <- sum(dbinom(2:n,size = n,prob = p))
atleast_two
#alternatively, 1-P(X<2)
pbinom(1,size = n,prob = p,lower.tail = FALSE)
```

console section 1.c

```
> #finding probability for perfect scores in at least 2 attempts out of 5
> # finding values fx(2)+fx(3)+fx(4) +fx(5) which denotes at least two perfect scores out of 5
attempts.
> atleast_two <- sum(dbinom(2:n,size = n,prob = p))
> atleast_two
[1] 0.66304
> #alternatively, 1-P(X<2)
> pbinom(1,size = n,prob = p,lower.tail = FALSE)
[1] 0.66304
```

Code section 1.d

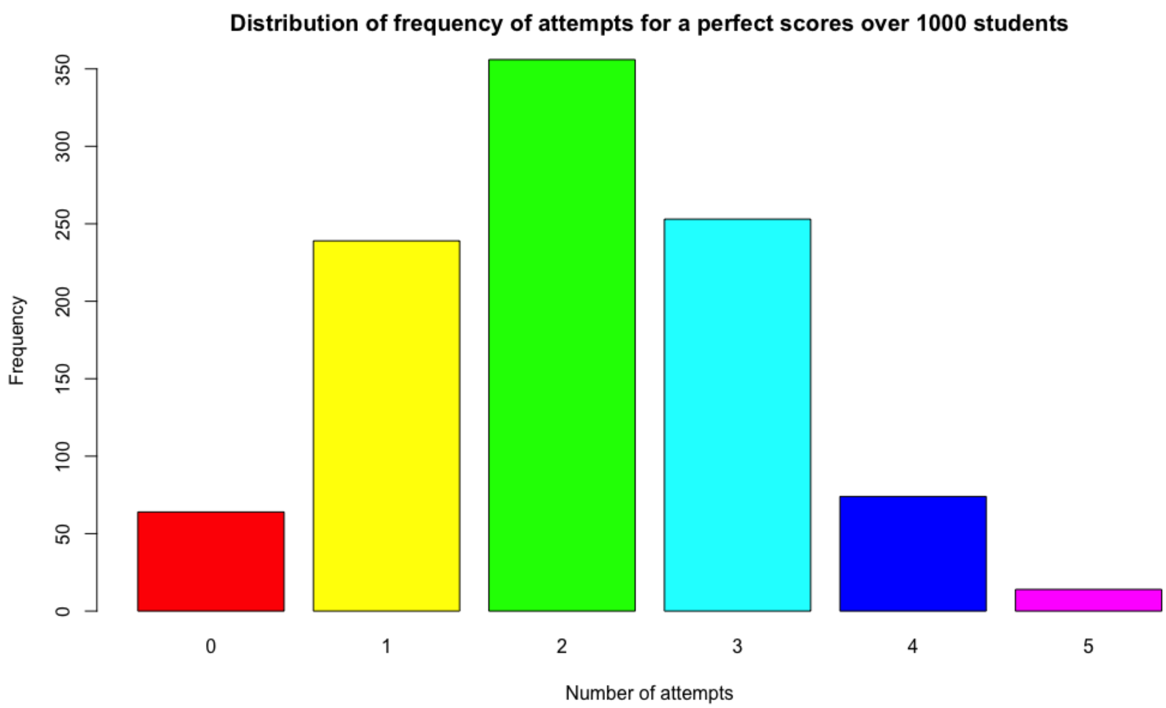
```
#d)
#using same distribution for 1000 students
rdistibution <- rbinom(1000,size = n,prob = p)
```

```
#plotting barplot
barplot(table(rdistibution),xlab = "Number of attempts",ylab = "Frequency",
       col = rainbow(6),main = "Distribution of frequency of attempts for a perfect scores over
1000 students"
       ,ylim = c(0,350))
```

Console section 1.d

```
#d)
> #using same distribution for 1000 students
> rdistibution <- rbinom(1000,size = n,prob = p)
> #plotting barplot
> barplot(table(rdistibution),xlab = "Number of attempts",ylab = "Frequency",
+        col = rainbow(6),main = "Distribution of frequency of attempts for a perfect scores over
1000 students"
+        ,ylim = c(0,350))
```

Plot section 1.d



#Part2) Negative Binomial distribution

****

Code section 2.a:

```
#probablity of perfect scores
p2 <- 60/100
```

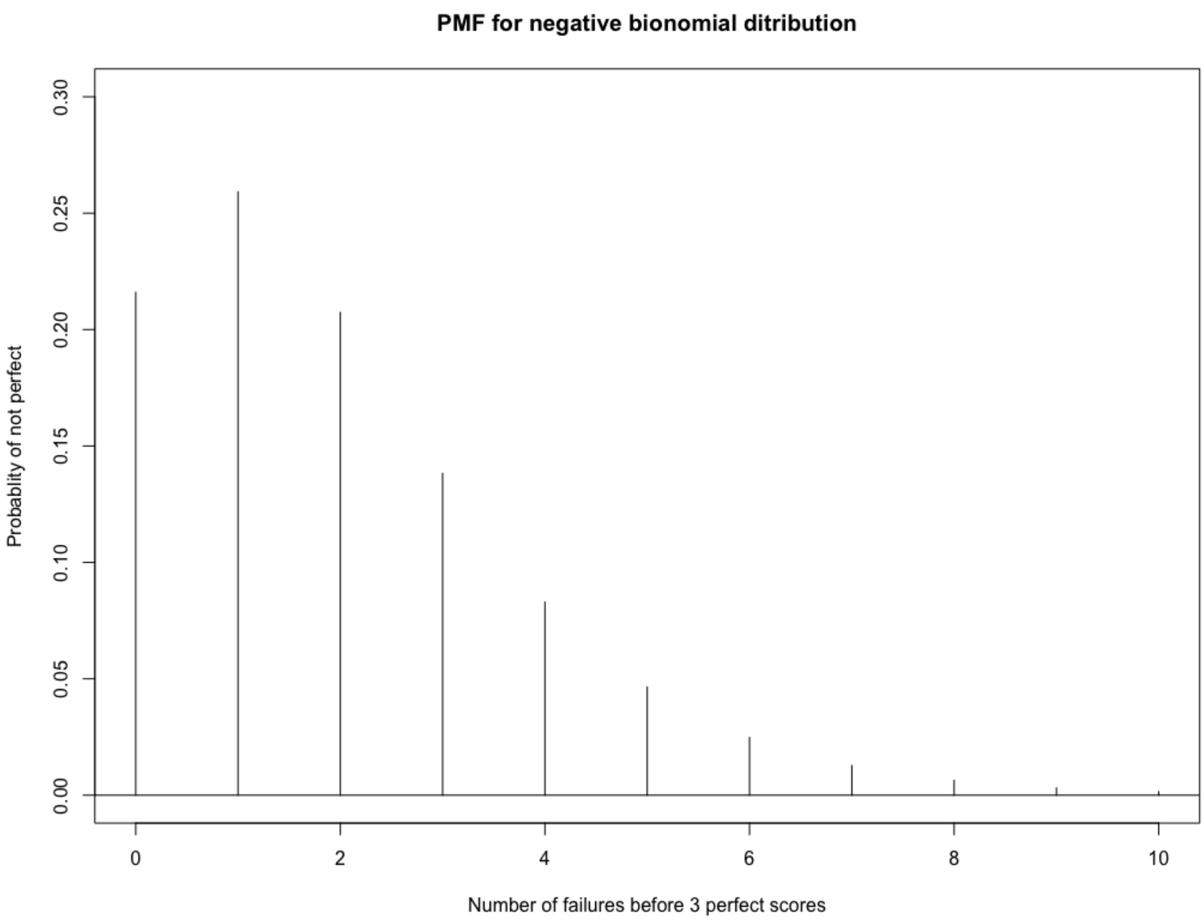
# number of peferct scores
r2 <- 3
#a)
#probability of geeting n not perfect scores before 3 perfect scores can be calculated by
0 not perfect before 3 perfect
#1 not perfect before 3 perfect
# 2 not perfect before 3 perfect
# 3 not perfect before 3 perfect
# so on ... upto 10 not perfect before 3 perfect scores.
pmf2 <- dnbinom(0:10,size = r2,prob = p2)

 # Plotting pmf
plot(0:10,pmf2,type = "h", xlab = "Number of failures before 3 perfect scores",
    ylab = "Probablity of not perfect",
    main ="PMF for negative bionomial ditribution",ylim = c(0,0.3))
abline(h=0)
plotting for cdf
cdf2 <- pnbinom(0:10,size = r2,prob = p2)
cdf2
#inserting 0 in cdf2
cdf2 <- c(0,cdf2)
cdf2
cdfplot <- stepfun(0:10,cdf2)
plot(cdfplot,verticals = FALSE,pch=16,main = "CDF plot",
    xlab = "Number of failures before 3 perfect scores",ylab = "CDF")

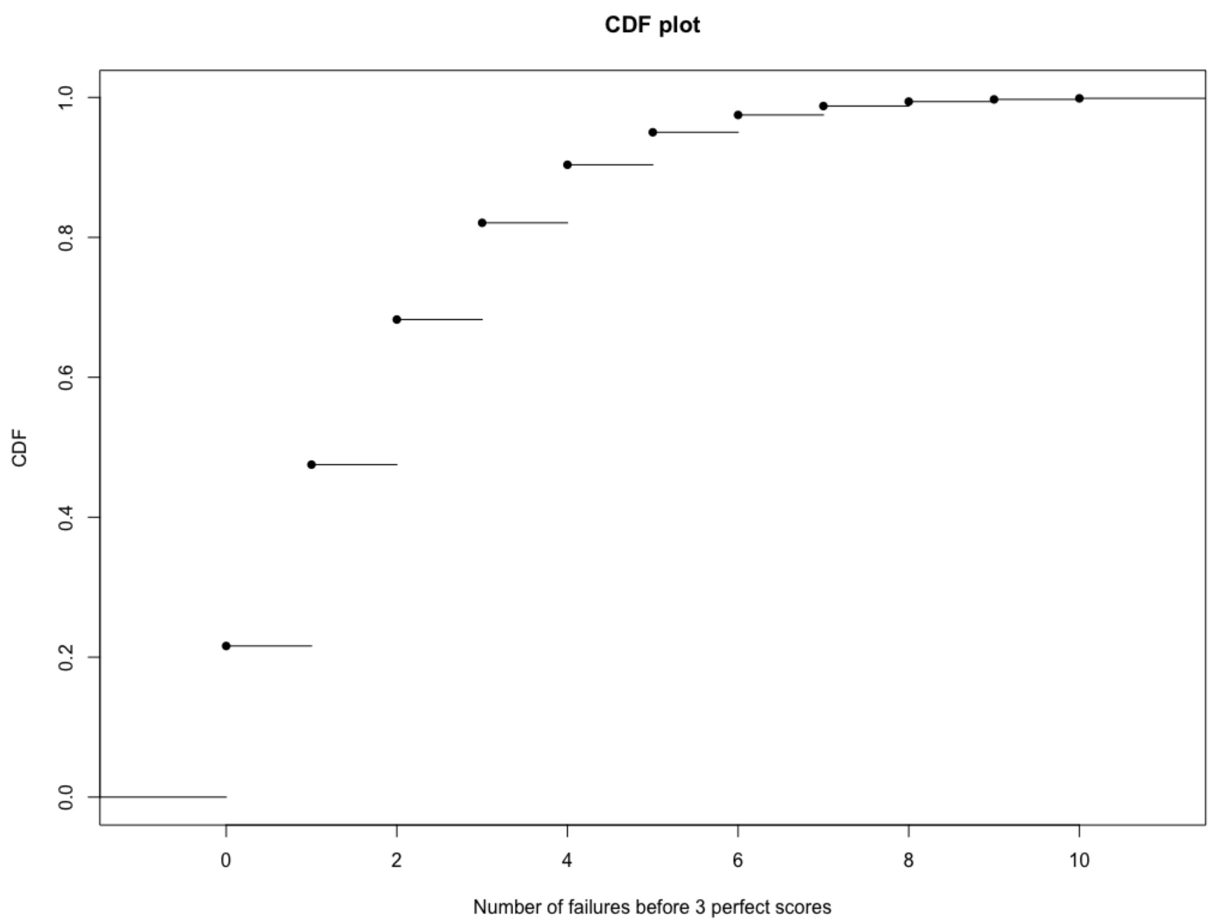
Console section 2.a:
#probablity of perferct scores
> p2 <- 60/100
> # number of peferct scores
> r2 <- 3
> #a)
> #probability of geeting n not perfect scores before 3 perfect scores can be calculated by
> # 0 not perfect before 3 perfect
> #1 not perfect before 3 perfect
> # 2 not perfect before 3 perfect
> # 3 not perfect before 3 perfect
> # so on ... upto 10 not perfect before 3 perfect scores.
> pmf2 <- dnbinom(0:10,size = r2,prob = p2)
>
> # Plotting pmf
> plot(0:10,pmf2,type = "h", xlab = "Number of failures before 3 perfect scores",
+   ylab = "Probablity of not perfect",
+   main ="PMF for negative bionomial ditribution",ylim = c(0,0.3))

```
> abline(h=0)
> cdf2 <- pnbinom(0:10,size = r2,prob = p2)
> cdf2
 [1] 0.2160000 0.4752000 0.6825600 0.8208000 0.9037440 0.9501926 0.9749652 0.9877054
0.9940755 0.9971898 0.9986847
> #inserting 0 in cdf2
```
#plotting for cdf
```
> cdf2 <- c(0,cdf2)
> cdf2
 [1] 0.0000000 0.2160000 0.4752000 0.6825600 0.8208000 0.9037440 0.9501926 0.9749652
0.9877054 0.9940755 0.9971898 0.9986847
> cdfplot <- stepfun(0:10,cdf2)
> plot(cdfplot,verticals = FALSE,pch=16,main = "CDF plot",
+      xlab = "Number of failures before 3 perfect scores",ylab = "CDF")
```

Plotting section 2.a:
Plotting for pmf:



Plotting for cdf:



Code section 2.b:

```
#b)
#finding probablity that student will have the 3 perfect scores
#with exactly 4 failures.
#i.e.P(X=4) ?

# using R function
failures4 <- dnbinom(4,size = r2,prob = p2)
failures4
# this can be also calculated as follow
#7th attempt is always perfect according to question
#first 6 attempts should have  2 perfect and and 4 failure attempts.
# probablity of geeting 4 exact failures before 3 perfect attempts is
alt_failures4 <- choose(6,2)*p2^2*(1-p2)^4*p2
alt_failures4
```

Console section 2.b:

```
> #b)
> #finding probablity that student will have the 3 perfect scores
```

```
> #with exactly 4 failures.
> #i.e.P(X=4) ?
>
> # using R function
> failures4 <- dnbinom(4,size = r2,prob = p2)
> failures4
[1] 0.082944
> # this can be also calculated as follow
> #7th attempt is always perfect according to question
> #first 6 attempts should have  2 perfect and and 4 failure attempts.
> # probablity of geeting 4 exact failures before 3 perfect attempts is
> alt_failures4 <- choose(6,2)*p2^2*(1-p2)^4*p2
> alt_failures4
[1] 0.082944
```

Code section 2.c:

```
#c)
# 3 perfet scores with at most 4 failures
#This includes following probablity.
#P(X=0)+P(X=1)+P(X=2)+P(X=3)+P(X=4)

prob_atmost4 <- pnbinom(4,size = r2,prob = p2)
prob_atmost4
```

Console section 2.c:

```
> #c)
> # 3 perfet scores with at most 4 failures
> #This includes following probablity.
> #P(X=0)+P(X=1)+P(X=2)+P(X=3)+P(X=4)
>
> prob_atmost4 <- pnbinom(4,size = r2,prob = p2)
> prob_atmost4
[1] 0.903744
```

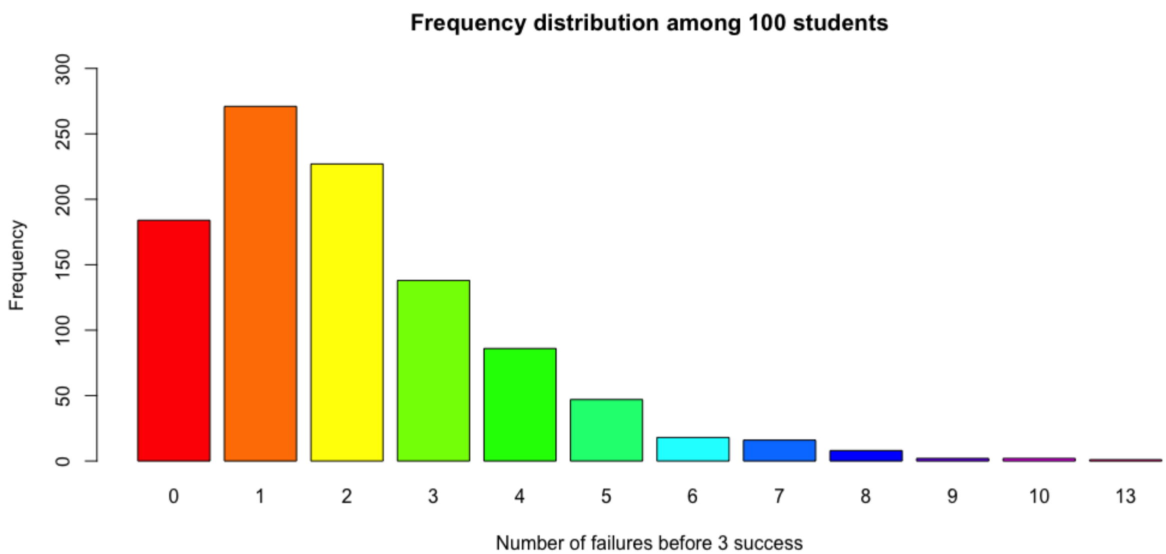
Code section 2.d:

```
#d)
#using R function
neg_bionomia_ditribution100 <- rnbinom(1000,size = r2,prob = p2)
barplot(table(neg_bionomia_ditribution100),col = rainbow(12),
       xlab = "Number of failures before 3 success",
       ylab = "Frequency",main = "Frequency distribution among 100 students",
       ylim = c(0,300))
```

Console section 2.d:

```
> #d)
> #using R function
> neg_bionomia_ditribution100 <- rnbinom(1000,size = r2,prob = p2)
> barplot(table(neg_bionomia_ditribution100),col = rainbow(12),
+       xlab = "Number of failures before 3 success",
+       ylab = "Frequency",main = "Frequency distribution among 100 students",
+       ylim = c(0,300))
```

Plot section 2.d:



#Part3) Hypergeometric distribution

****

Code section 3.a:

```
#a)
# Probability distribution of this question looks like
#Pm(X=0)and Pp(X=20) -.multiple 0 and programming 20 questions
#Pm(X=1) and Pp(X=19)->multiple 1 and programming 19
#so on.....
#Pm(X=20) and Pp(X=0)-> multiple 20 and programming 0


#using R function
Multi_quest <- 60
program_quest <- 40
k_chose <- 20
```

```
pmf_hyper <- dhyper(0:k_chose,m=Multi_quest,n=program_quest,k=k_chose)
cdf_hyper <- phyper(0:k_chose,m=Multi_quest,n=program_quest,k=k_chose)
```

 # Plotting pmf

```
heights_hyper <- dhyper(0:k_chose,m=Multi_quest,n=program_quest,k=k_chose)
plot(0:k_chose,pmf_hyper,type = "h",
    xlab = "Number of multiple choice question choosen out of 20",
    ylab = "probablity of multiple question choosen",
    main ="PMF when 20 qeustions are choosen ")
points(0:k_chose, heights_hyper,pch=16)
```

#plotting cdf

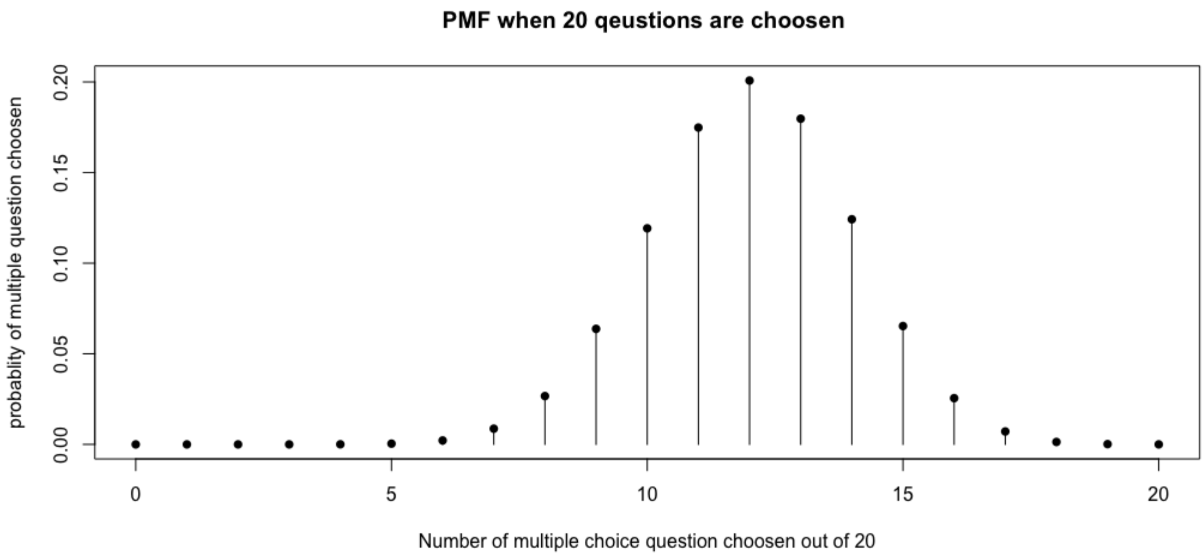
```
#inserting 0 in cdf2_hyper
cdf_hyper <- c(0,cdf_hyper)
cdfplot_hyper <- stepfun(0:k_chose,cdf_hyper)
plot(cdfplot_hyper,verticals = FALSE,pch=16,main = "CDF_hyper plot",
    xlab = "Number of multiple choice questions choosen out of 20",
    ylab = "CDF",col = rainbow(20))
```

Console section 3.a:

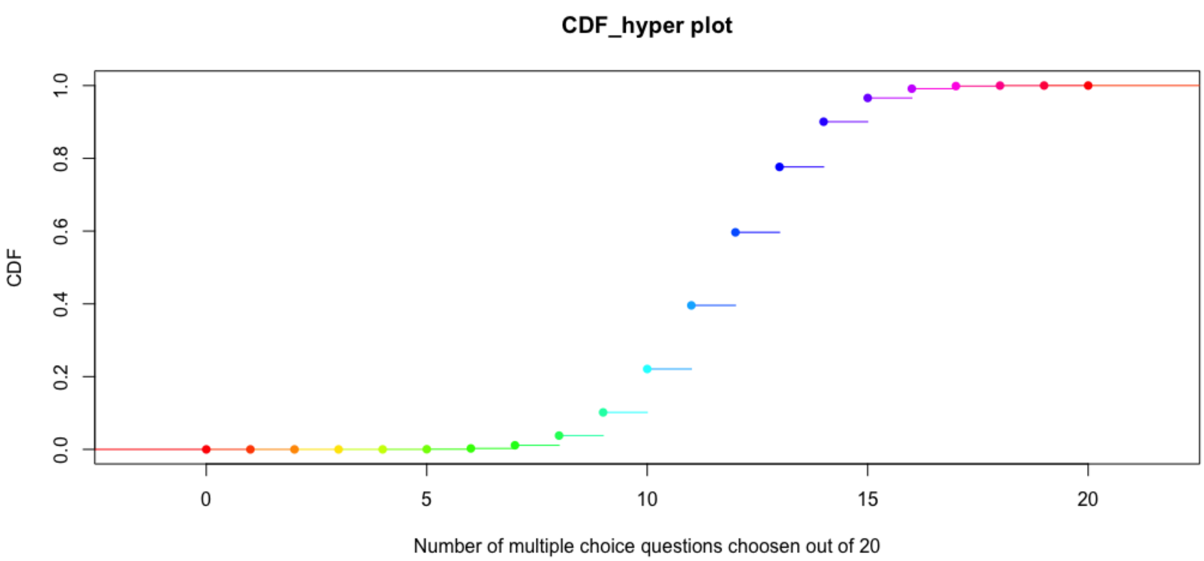
```
#a)
> # Probability distribution of this question looks like
> #Pm(X=0)and Pp(X=20) -.multiple 0 and programming 20 questions
> #Pm(X=1) and Pp(X=19)->multiple 1 and programming 19
> #so on.....
> #Pm(X=20) and Pp(X=0)-> multiple 20 and programming 0
>
>
> #using R function
> Multi_quest <- 60
> program_quest <- 40
> k_chose <- 20
> pmf_hyper <- dhyper(0:k_chose,m=Multi_quest,n=program_quest,k=k_chose)
> cdf_hyper <- phyper(0:k_chose,m=Multi_quest,n=program_quest,k=k_chose)
>
> # Plotting pmf
> heights_hyper <- dhyper(0:k_chose,m=Multi_quest,n=program_quest,k=k_chose)
> plot(0:k_chose,pmf_hyper,type = "h",
+     xlab = "Number of multiple choice question choosen out of 20",
+     ylab = "probablity of multiple question choosen",
+     main ="PMF when 20 qeustions are choosen ")
> points(0:k_chose, heights_hyper,pch=16)
> #plotting cdf
> #inserting 0 in cdf2_hyper
> cdf_hyper <- c(0,cdf_hyper)
```

```
> cdfplot_hyper <- stepfun(0:k_chose,cdf_hyper)
> plot(cdfplot_hyper,verticals = FALSE,pch=16,main = "CDF_hyper plot",
+      xlab = "Number of multiple choice questions choosen out of 20",
+      ylab = "CDF",col = rainbow(20))
```

Plotting section 3.a:
Plotting for pmf:



Plotting for cdf:



Code section 3.b:

```
#b
#Pm(X=10)?
```

```
# This means P(10 multiple choice questions,10 programming questions)
#using combinations method
prob_10multiple_quest <- choose(60,10)*choose(40,10)/choose(100,20)
prob_10multiple_quest
#using R explicit method
#from questions, m=60,n=40,k=20
dhyper(10,m=60,n=40,k=20)
```

console section 3.b:

```
#b
> #Pm(X=10)?
> # This means P(10 multiple choice questions,10 programming questions)
> #using combinations method
> prob_10multiple_quest <- choose(60,10)*choose(40,10)/choose(100,20)
> prob_10multiple_quest
[1] 0.1192361
> #using R explicit method
> #from questions, m=60,n=40,k=20
> dhyper(10,m=60,n=40,k=20)
[1] 0.1192361
```

Code section 3.c:

```
#c)P(X>=10)?
#Using R function
phyper(9,m=Multi_quest,n=program_quest,k=k_chose,lower.tail = FALSE)
#alliteratively,
sum(dhyper(10:20,m=Multi_quest,n=program_quest,k=k_chose))
```

console section 3.c:

```
> #c)P(X>=10)?
> #Using R function
> phyper(9,m=Multi_quest,n=program_quest,k=k_chose,lower.tail = FALSE)
[1] 0.8982561
> #alliteratively,
> sum(dhyper(10:20,m=Multi_quest,n=program_quest,k=k_chose))
[1] 0.8982561
```

Code section 3.d:

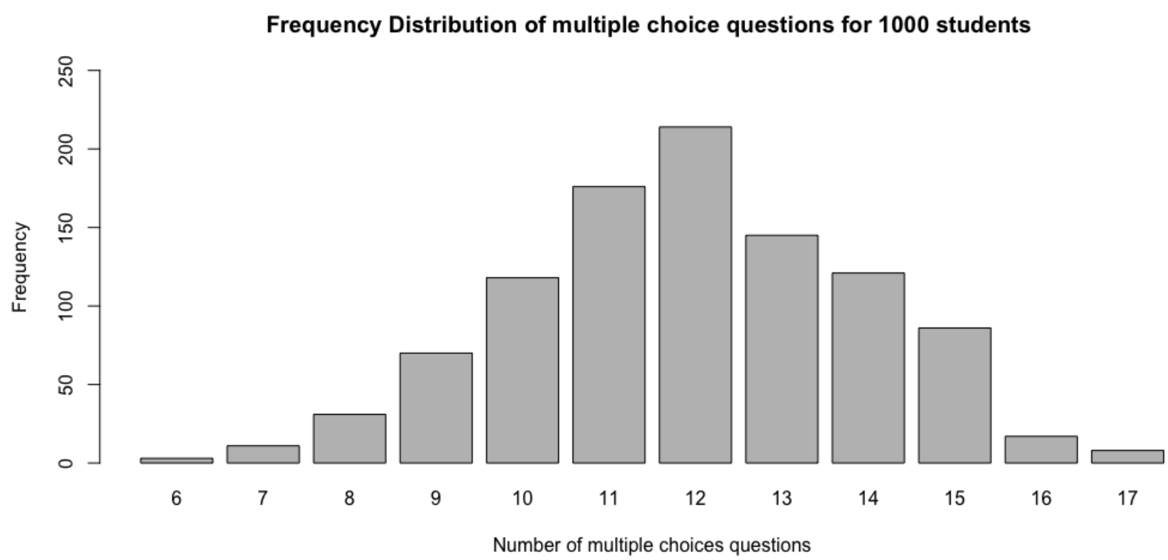
```
#d)
#using r function to find random distribution
multiple_choice_distrib <- rhyper(1000,m=Multi_quest,n=program_quest,k=k_chose)
barplot(table(multiple_choice_distrib),ylim = c(0,250),
	xlab = "Number of multiple choices questions",
```

```
    ylab = "Frequency",
    main = "Frequency Distribution of multiple choice questions  for 1000 students")
```

Console section 3.d:

```
#d)
> #using r function to find random distribution
> multiple_choice_distrib <- rhyper(1000,m=Multi_quest,n=program_quest,k=k_chose)
> barplot(table(multiple_choice_distrib),ylim = c(0,250),
+       xlab = "Number of multiple choices questions",
+       ylab = "Frequency",
+       main = "Frequency Distribution of multiple choice questions  for 1000 students")
```

Plot section 3.d:



#Part4) Poisson distribution

****

Code section 4.a:

```
#a)
lamda <- 10
#using R function
#probablity of getting exactly 8 question per day is
#P(X=8)
dpois(8,lambda = lamda)
```

Console section 4.a:

```
#a)
> lamda <- 10
> #using R function
> #probablity of getting exactly 8 question per day is
> #P(X=8)
> dpois(8,lambda = lamda)
[1] 0.112599
```

Code section 4.b:

```
#b)getting at most 8 questions is
#P(X<=8)
ppois(8,lambda = lamda)
#alternatively,
sum(dpois(0:8,lambda = lamda))
```

Console section 4.b:

```
#b)getting at most 8 questions is
> #P(X<=8)
> ppois(8,lambda = lamda)
[1] 0.3328197
> #alternatively,
> sum (dpois(0:8,lambda = lamda))
[1] 0.3328197
```

Code section 4.c:

```
#c)
sum(dpois(6:12,lambda = lamda))
#alternative1
ppois(12,lambda = lamda)-ppois(5,lambda = lamda)
#alternative2
diff(ppois(c(5,12),lambda = lamda))
```

Console section 4.c:

```
> #c)
> sum(dpois(6:12,lambda = lamda))
[1] 0.7244705
> #alternative1
> ppois(12,lambda = lamda)-ppois(5,lambda = lamda)
[1] 0.7244705
> #alternative2
> diff(ppois(c(5,12),lambda = lamda))
[1] 0.7244705
```

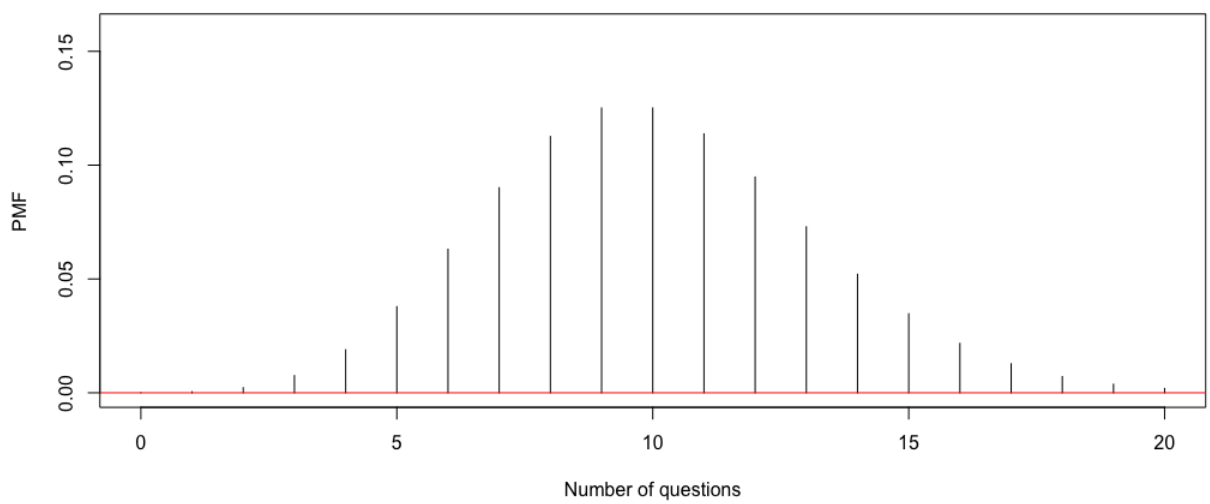
Code section 4.d:

```
#d)
pmf_pois <- dpois(0:20,lambda = lamda)
plot(0:20,pmf_pois,type = "h",xlab = "Number of questions",
    ylab = "PMF",ylim = c(0,0.16))
abline(h=0,col="red")
```

Console secton 4.d:

```
#d)
> pmf_pois <- dpois(0:20,lambda = lamda)
> plot(0:20,pmf_pois,type = "h",xlab = "Number of questions",
+      ylab = "PMF",ylim = c(0,0.16))
> abline(h=0,col="red")
```

Plot section 4.d:



Code section 4.e:

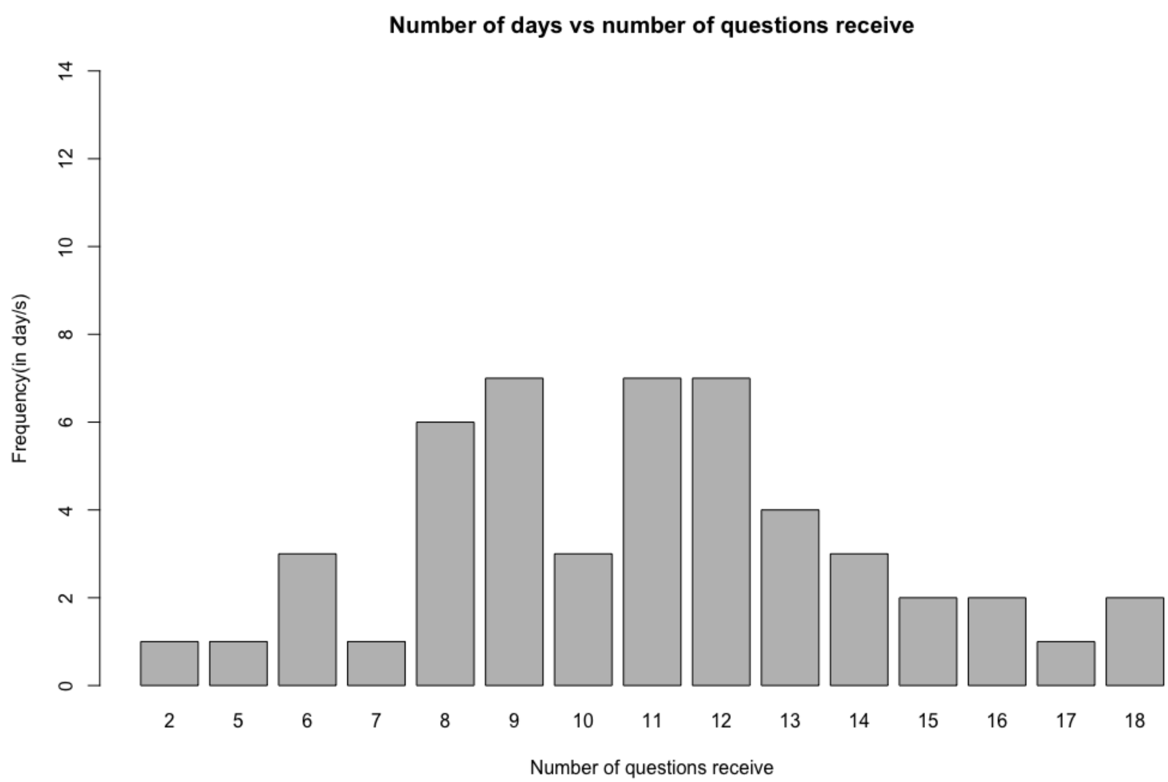
```
#e)
#distribution of number of questions a professor gets during 50 days periods

frequency_distrib <- rpois(50,lambda = lamda)
barplot(table(frequency_distrib),ylim = c(0,14),
      main = "Number of days vs number of questions receive",
      xlab = "Number of questions receive",ylab = "Frequency(in day/s)")

#plotting box plot of the number of questions
boxplot(frequency_distrib,horizontal = TRUE,pch=16)
```

Console section 4.e:

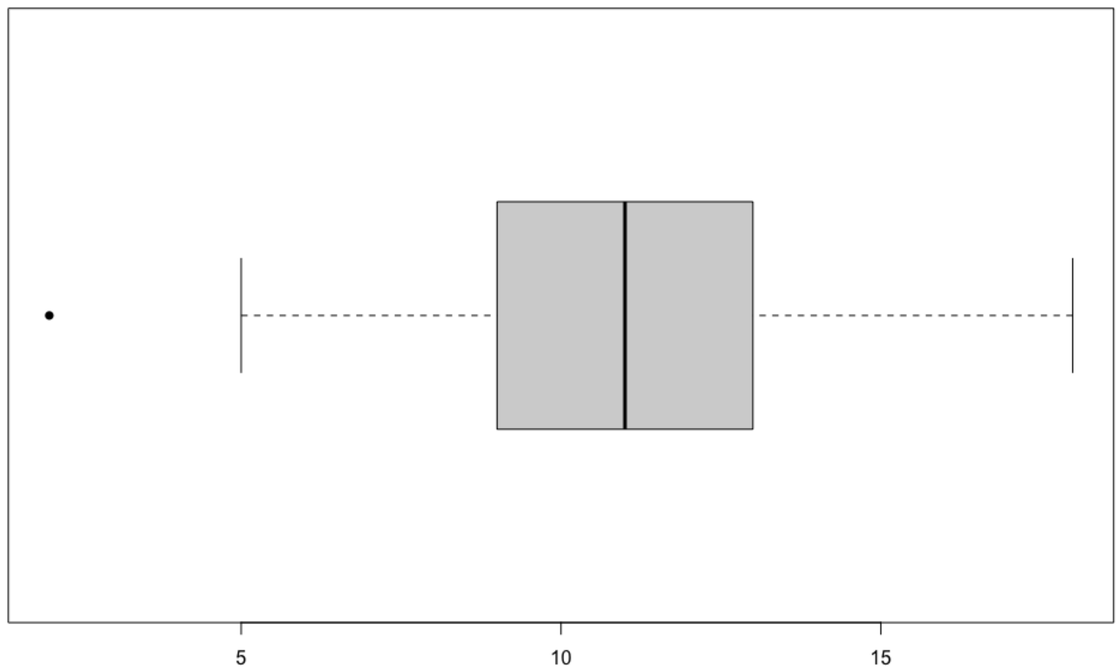
```
#e)
> #distribution of number of question a professor gets during 50 days periods
>
> frequency_distrib <- rpois(50,lambda = lamda)
> barplot(table(frequency_distrib),ylim = c(0,14),
+        main = "Number of days vs number of questions receive",
+        xlab = "Number of questions receive",ylab = "Frequency(in day/s)")

> #plotting box plot of the number of questions
> boxplot(frequency_distrib,horizontal = TRUE,pch=16)
```

Plotting section 4.e:

1.Barplot



2. Boxplot:



Inferences from the plots:
1.since these values are randomly generated, each time I run, the shape of the plot changes slightly.
2. there is one outlier towards lower end of the data set generated by rpois() function
3.upper whisker is longer than lower whisker, this means, there is more spread of data towards upper end than lower end.

#Part5) Normal distribution

****
Code section 5.a:

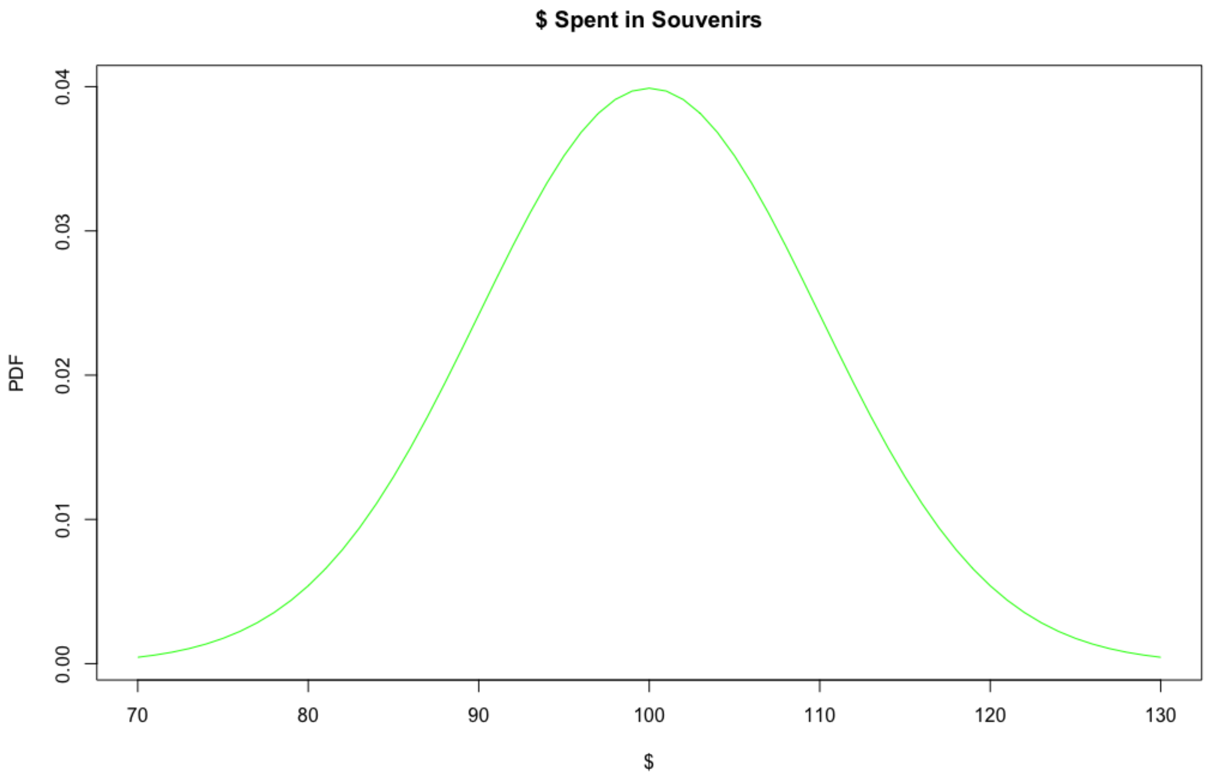
```
#a)
mu <- 100
sigma_sd <- 10
lower_end <- mu-3*sigma_sd
higher_end <- mu+3*sigma_sd
#probablity distribution
pdf <- dnorm(lower_end:higher_end,mean = mu,sd= sigma_sd)
plot(lower_end:higher_end,pdf,type = "l",col="green",main = "$ Spent in Souvenirs",
```

```
xlab = "$",ylab = "PDF")
```

Console section 5.a:

```
#a)
> mu <- 100
> sigma_sd <- 10
> lower_end <- mu-3*sigma_sd
> higher_end <- mu+3*sigma_sd
> #probablity distribution
> pdf <- dnorm(lower_end:higher_end,mean = mu,sd= sigma_sd)
> plot(lower_end:higher_end,pdf,type = "l",col="green",main = "$ Spent in Souvenirs",
+      xlab = "$",ylab = "PDF")
```

Plot section 5.a:



Code section 5.b:

```
#b)
#above 120 means beyound 2sd toeards upper ends
above120 <- pnorm(120,mean = mu,sd=sigma_sd,lower.tail = FALSE)
above120
#2.27% of chance sthat visitor will spend above $120
```

Console section 5.b:

b)

```
> #above 120 means beyound 2sd toeards upper ends
> above120 <- pnorm(120,mean = mu,sd=sigma_sd,lower.tail = FALSE)
> above120
[1] 0.02275013
> #2.27% of chance that visitor will spend above $120
```

Code section 5.c:

```
#c)
pnorm(90,mean = mu,sd=sigma_sd)-pnorm(80,mean = mu,sd=sigma_sd)
#alternatively
sum(dnorm(80:90,mean = mu,sd=sigma_sd))
#That is 13.59% chance.
```

Console section 5.c:

```
#c)
> pnorm(90,mean = mu,sd=sigma_sd)-pnorm(80,mean = mu,sd=sigma_sd)
[1] 0.1359051
> #alternatively
> sum(dnorm(80:90,mean = mu,sd=sigma_sd))
[1] 0.1508149
> #That is 13.59% chance.
```

Code section 5.d:

```
#d)
#with in 1sd
pnorm(mu+1*sigma_sd,mean = mu,sd=sigma_sd)-pnorm(mu-1*sigma_sd,mean =
mu,sd=sigma_sd)
#that is 68.26% chances

#with in 2sd
pnorm(mu+2*sigma_sd,mean = mu,sd=sigma_sd)-pnorm(mu-2*sigma_sd,mean =
mu,sd=sigma_sd)
#that is 95.44% chances

#with in 3sd
pnorm(mu+3*sigma_sd,mean = mu,sd=sigma_sd)-pnorm(mu-3*sigma_sd,mean =
mu,sd=sigma_sd)
#That is 99.73% chance.
```

Console section 5.d:

```
#d)
> #with in 1sd
```

```
> pnorm(mu+1*sigma_sd,mean = mu,sd=sigma_sd)-pnorm(mu-1*sigma_sd,mean =
mu,sd=sigma_sd)
[1] 0.6826895
> #that is 68.26% chances.
>
> #with in 2sd
> pnorm(mu+2*sigma_sd,mean = mu,sd=sigma_sd)-pnorm(mu-2*sigma_sd,mean =
mu,sd=sigma_sd)
[1] 0.9544997
> #that is 95.44% chances.
>
> #with in 3sd
> pnorm(mu+3*sigma_sd,mean = mu,sd=sigma_sd)-pnorm(mu-3*sigma_sd,mean =
mu,sd=sigma_sd)
[1] 0.9973002
> #That is 99.73% chance.
>
```

Code section 5.e:

```
#e)
#Middle 80% values.since distribution is symmetrical,its is distributed 40 % on
#each side of the mean
# this means lower 10% upper 90% covers middle 80%
#so two values can be calculated by using qnorm method.
c(qnorm(0.1,mean = mu,sd=sigma_sd),qnorm(0.9,mean = mu,sd=sigma_sd))
```

Console section 5.e:

```
#e)
> #Middle 80% values.since distribution is symmetrical,its is distributed 40 % on
> #each side of the mean
> # this means lower 10% upper 90% covers middle 80%
> #so two values can be calculated by using qnorm method.
> c(qnorm(0.1,mean = mu,sd=sigma_sd),qnorm(0.9,mean = mu,sd=sigma_sd))
[1] 87.18448 112.81552
```

So Two values are 87.18448 and 112.81552.

Code section 5.f:

```
#f)
```

```
#top 2 of the spenders means finding values for 98th percentale
top_2val <- qnorm(0.98,mean = mu,sd=sigma_sd)
top_2val
paste("Spending ",top_2val,"or more will be in top 2% group and gets free T-shirt")
```

Console section 5.f:

```
#f)
> #top 2 of the spenders means finding values for 98th percentale
> top_2val <- qnorm(0.98,mean = mu,sd=sigma_sd)
> top_2val
[1] 120.5375
> paste ("Spending ",top_2val,"or more will be in top 2% group and gets free T-shirt")
[1] "Spending 120.54 or more will be in top 2% group and gets free T-shirt."
```

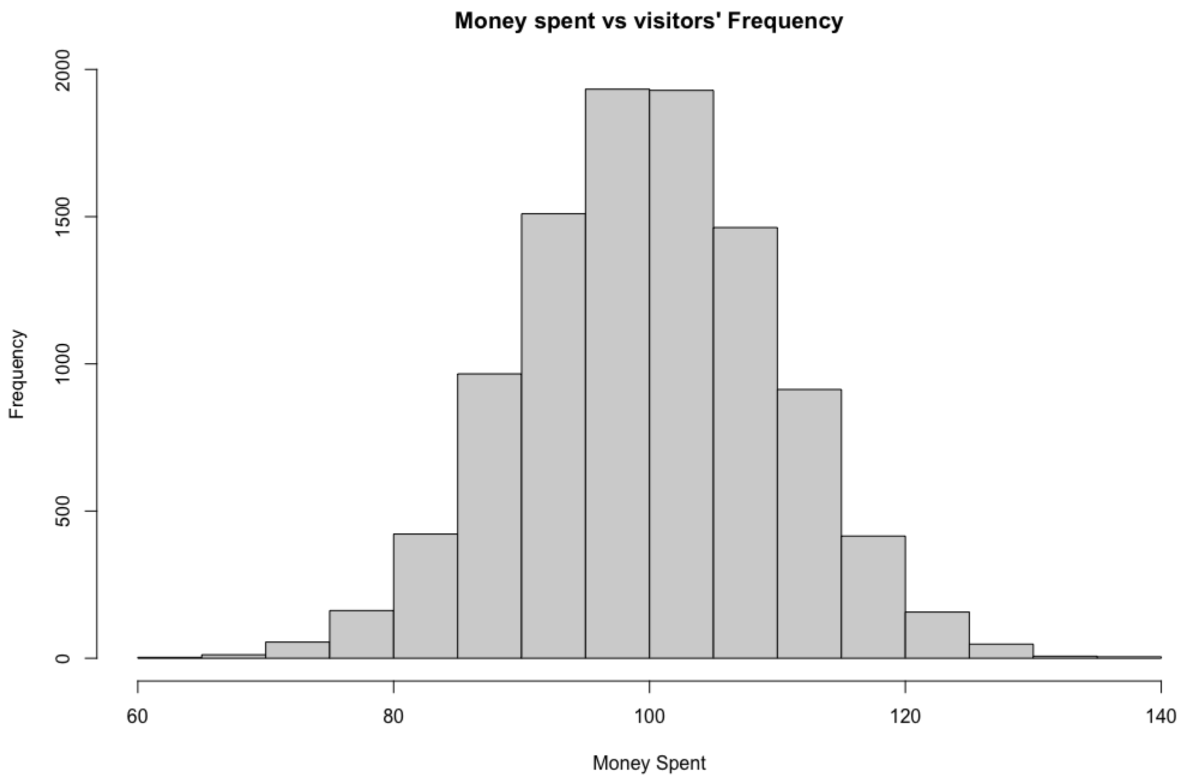
Code section 5.g:

```
#g
# with above mean and sd, plot of distribution of 10000 visitors can be done by
#first using rnorm and then using appropriate ploting method.
random_10000 <- rnorm(10000,mean = mu,sd=sigma_sd)
random_10000_hist <- hist(random_10000,xlab = "Money Spent",ylab = "Frequency",
    main = "Money spent vs visitors' Frequency")
```

Console section 5.g:

```
#g
> # with above mean and sd, plot of distribution of 10000 visitors can be done by
> #first using rnorm and then using appropriate ploting method.
> random_10000 <- rnorm(10000,mean = mu,sd=sigma_sd)
> random_10000_hist <- hist(random_10000,xlab = "Money Spent",ylab = "Frequency",
+      main = "Money spent vs visitors' Frequency")
```

Plot section 5.g:


Money spent vs visitors' Frequency
Frequency
0
500
1000
1500
2000
60
80
100
120
140
Money Spent

The End

****