1. let’s assume that two candidates are running in an election for Governor of California. This fictitious election pits Mr. Gubinator vs. Mr. Ventura. We would like to know who is winning the race, and therefore we conduct a poll of likely voters in California. If the poll gives the voters a choice between the two candidates, then the results can be reasonably modeled with the Binomial Distribution. The polls indicate that 62% of the population intend to vote for Mr. Gubinator (Success).

1. Calculate and PLOT the binomial probabilities (PMF and CDF), and CF (confidence interval) for the distribution of the samples

Calculate

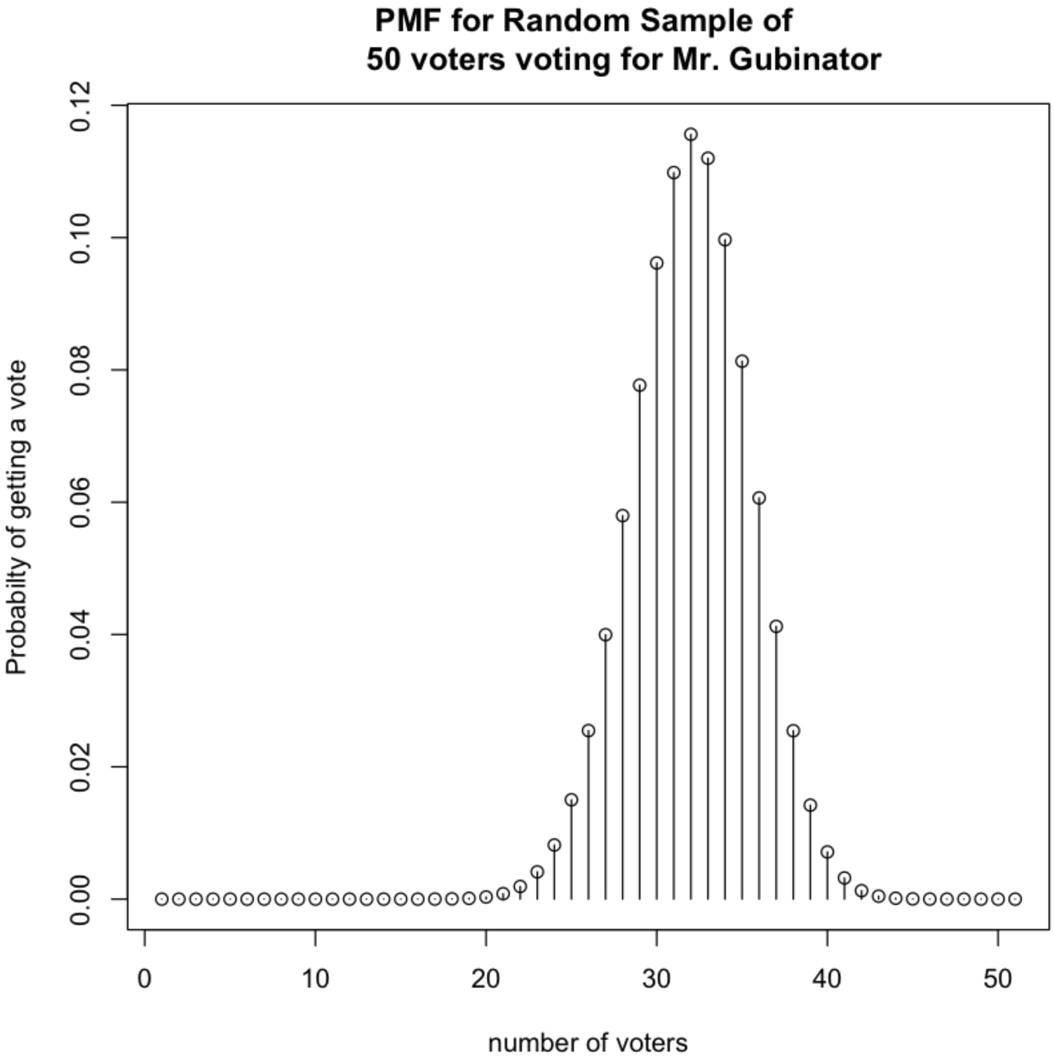
1. in a random sample of 50 voters we find 10 likely voters to vote for Mr. Gubinator.
2. in a random sample of 50 we find at least 3 likely voters to vote for Mr. Gubinator.
3. What is the average and standard deviation of the number of voters that we will find in our random sample? Hint: use the estimators of the respective distribution.

In [1]:

```
install.packages("binom", repos="http://R-Forge.R-project.org")
```

In [2]:

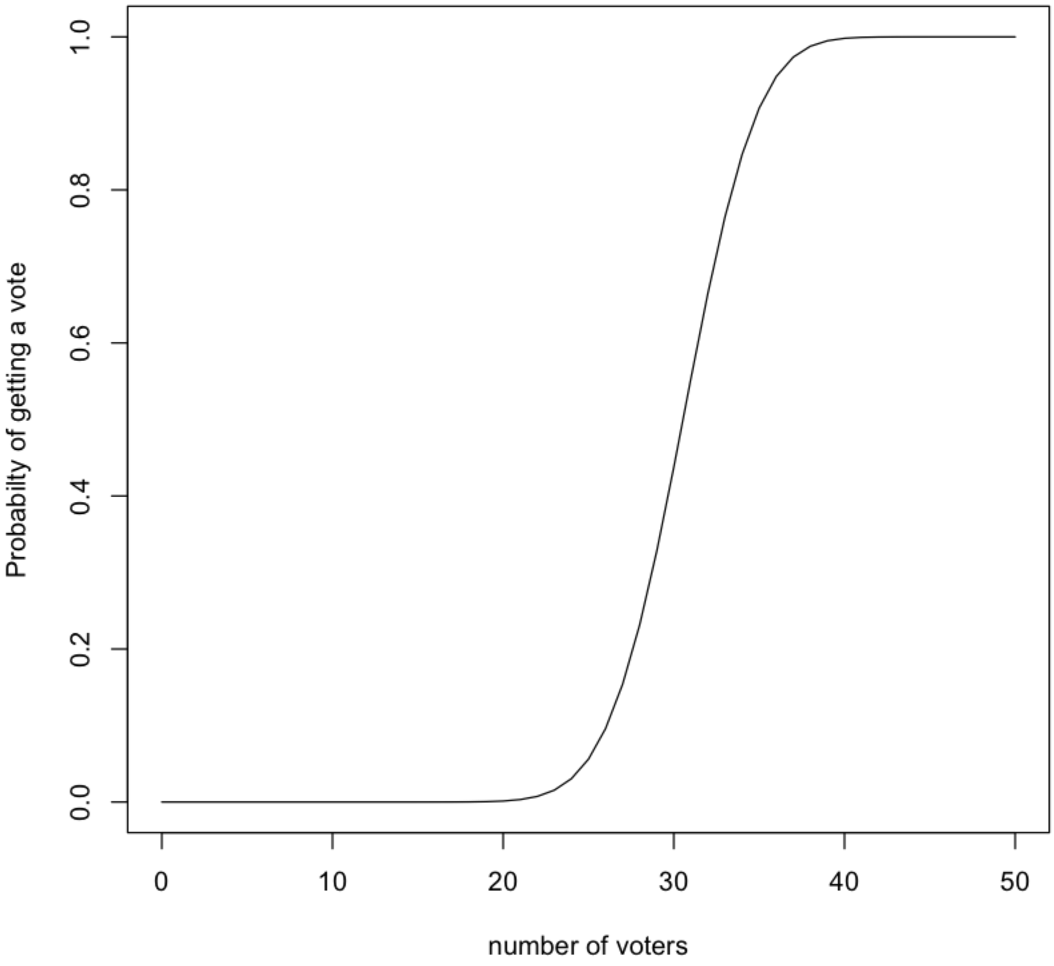
```
##Code
## Question 1 PMF
success = 0.62
polls_pmf <- dbinom(0:50, size = 50, prob = success)
plot(polls_pmf, type = "h", main="PMF for Random Sample of
        50 voters voting for Mr. Gubinator",xlab = "number of vo
ters"
        ,ylab = "Probabilty of getting a vote")
points(polls_pmf,pch=1)

## Question 1 CDF
plot(0:50,cumsum(polls_pmf), type = 'l', main = "Cumulative Dist
ributions Function
 for Random Sample of 50 voters/10 voters likely to vote for Mr.
Gubinator",
    xlab = "number of voters", ylab = "Probabilty of getting a v
ote")

## Question 1 Confidence Interval
library(binom)
cInt = binom.confint(31, 50, conf.level = 0.95, "exact")
cInt

polls_pmf <- dbinom(0:50, size = 50, prob = success)
plot(polls_pmf, type = "h", main="PMF for Random Sample of
        50 voters voting for Mr. Gubinator",xlab = "number of vo
ters"
        ,ylab = "Probabilty of getting a vote")
points(polls_pmf,pch=1)
abline(h = 0.05, xlim =c(0,1), col ='red')
legend("topleft","Confidence Interval 95%")
```

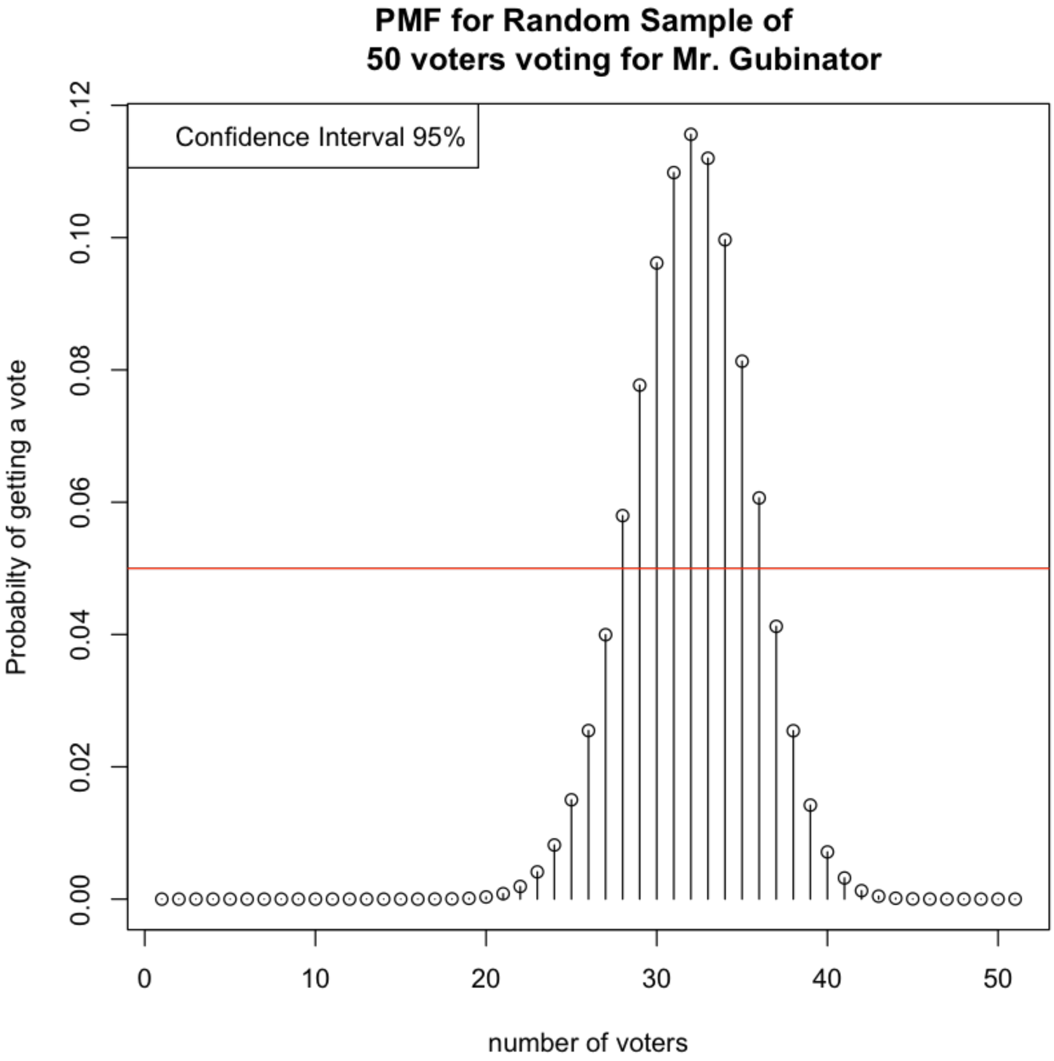


A data.frame: 1 × 6

	method	x	n	mean	lower	upper
	<fct>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	exact	31	50	0.62	0.4717492	0.7534989

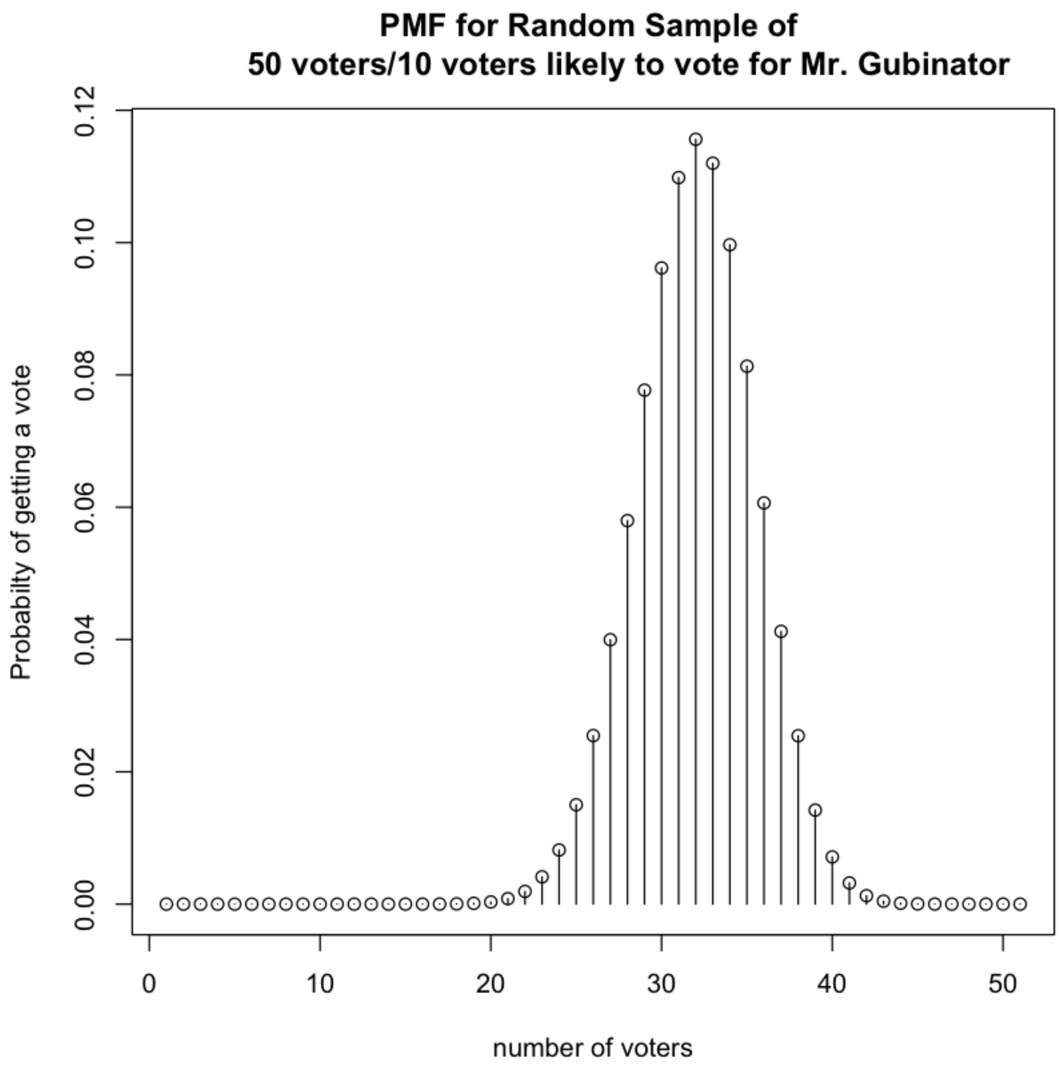
Cumulative Distributions Function
for Random Sample of 50 voters/10 voters likely to vote for Mr. Gubinat




PMF for Random Sample of
50 voters voting for Mr. Gubinator
Confidence Interval 95%
Probabilty of getting a vote
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0
10
20
30
40
50
number of voters

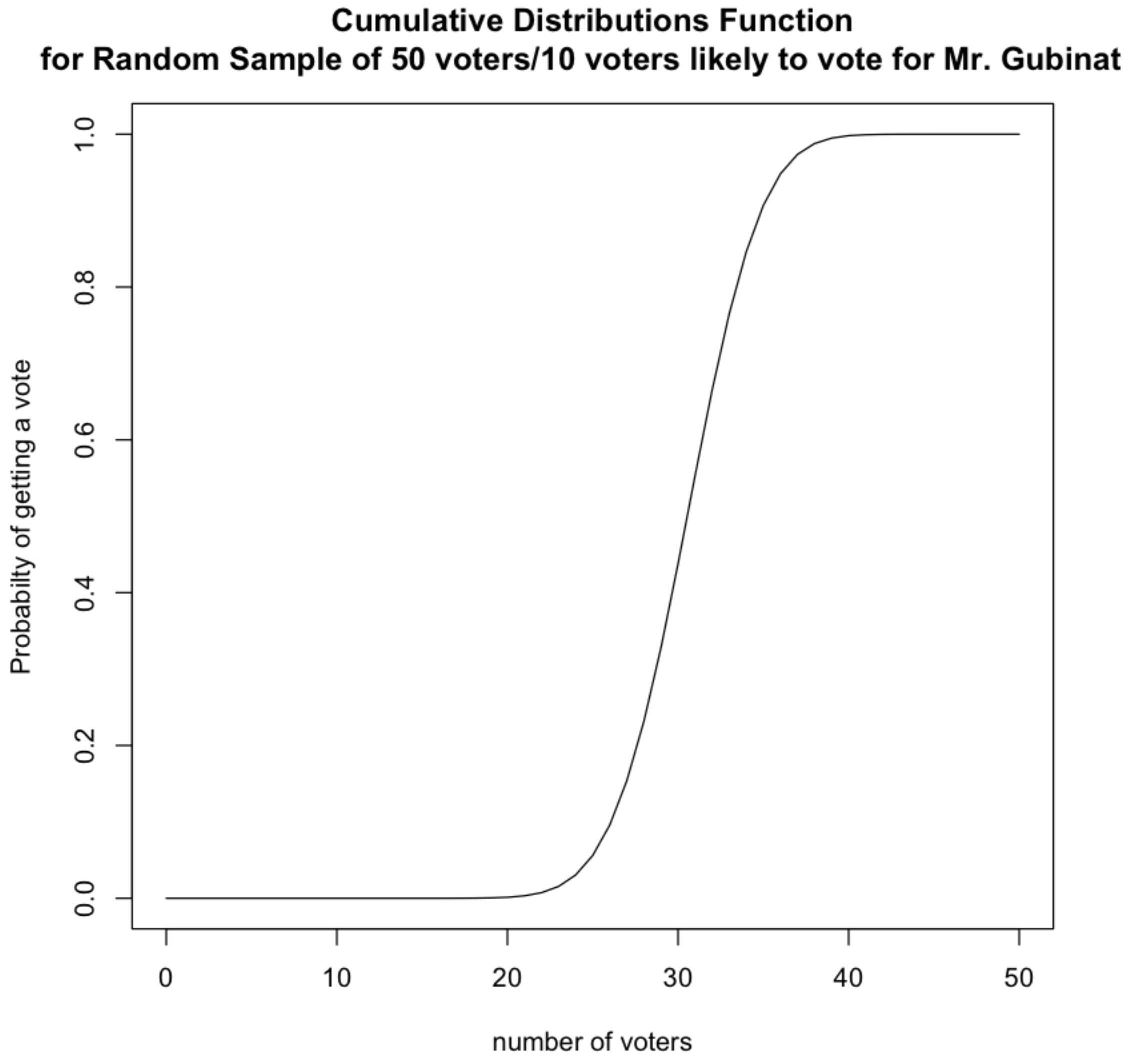
In [3]:

```
##Code
## Question 2 PMF
success = 0.62
polls_pmf <- dbinom(0:50, size = 50, prob = success)
plot(polls_pmf, type = "h", main="PMF for Random Sample of
        50 voters/10 voters likely to vote for Mr. Gubinator",xl
ab = "number of voters"
        ,ylab = "Probabilty of getting a vote")
points(polls_pmf,pch=1)

## Question 2 CDF
plot(0:50,cumsum(polls_pmf), type = 'l', main = "Cumulative Dist
ributions Function
 for Random Sample of 50 voters/10 voters likely to vote for Mr.
Gubinator",
    xlab = "number of voters", ylab = "Probabilty of getting a v
ote")

## Question 2 Confidence Interval
library(binom)
cI = binom.confint(10, 50, conf.level = 0.95, "exact")
cI

polls_pmf <- dbinom(0:50, size = 50, prob = success)
plot(polls_pmf, type = "l", main="PMF for Random Sample of
        50 voters/10 voters likely to vote for Mr. Gubinator",xl
ab = "number of voters"
        ,ylab = "Probabilty of getting a vote")
abline(h = 0.05, xlim =c(0,1), col ='red')
legend("topleft","Confidence Interval 95%")

prob1 = dbinom(10,50,success)
cat("Binomial Probability in a random sample of 50 voters to fin
d 10 likely voters to vote for Mr. Gubinator: "
    ,prob1, " \n")
```

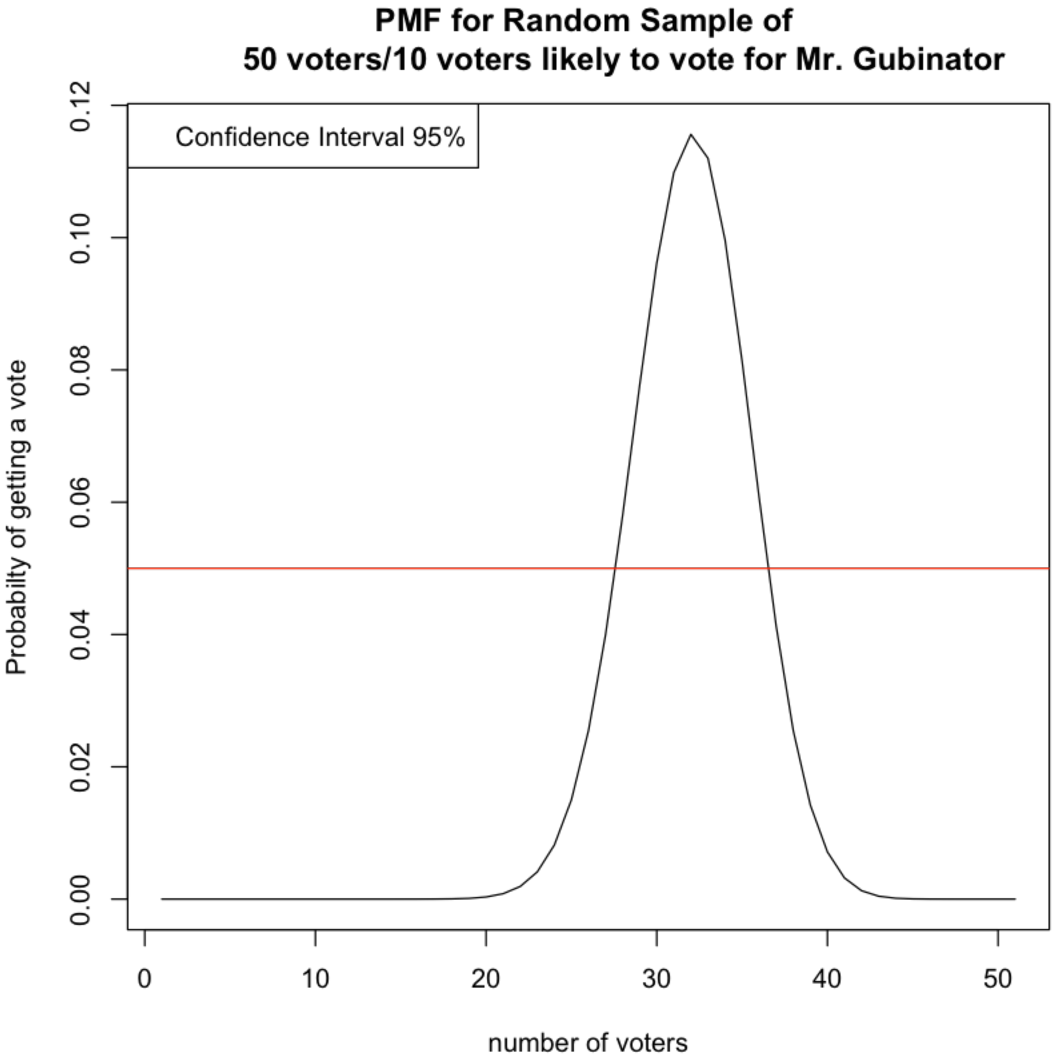

PMF for Random Sample of
50 voters/10 voters likely to vote for Mr. Gubinator

A data.frame: 1 × 6

	method	x	n	mean	lower	upper
	<fct>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	exact	10	50	0.2	0.1003022	0.3371831

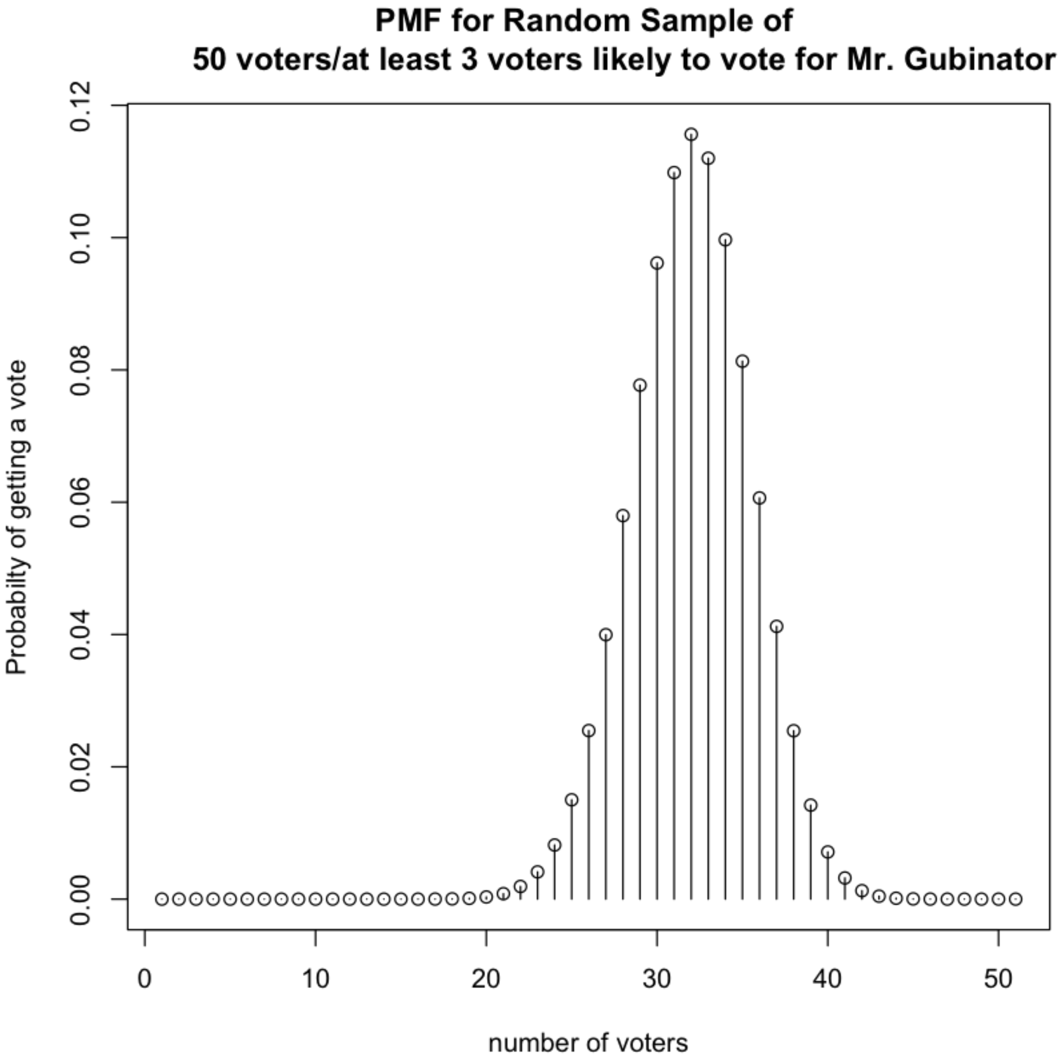


```
Binomial Probability in a random sample of 50 voters
to find 10 likely voters to vote for Mr. Gubinator:
1.339453e-09
```


PMF for Random Sample of
50 voters/10 voters likely to vote for Mr. Gubinator
Confidence Interval 95%
Probabilty of getting a vote
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0
10
20
30
40
50
number of voters

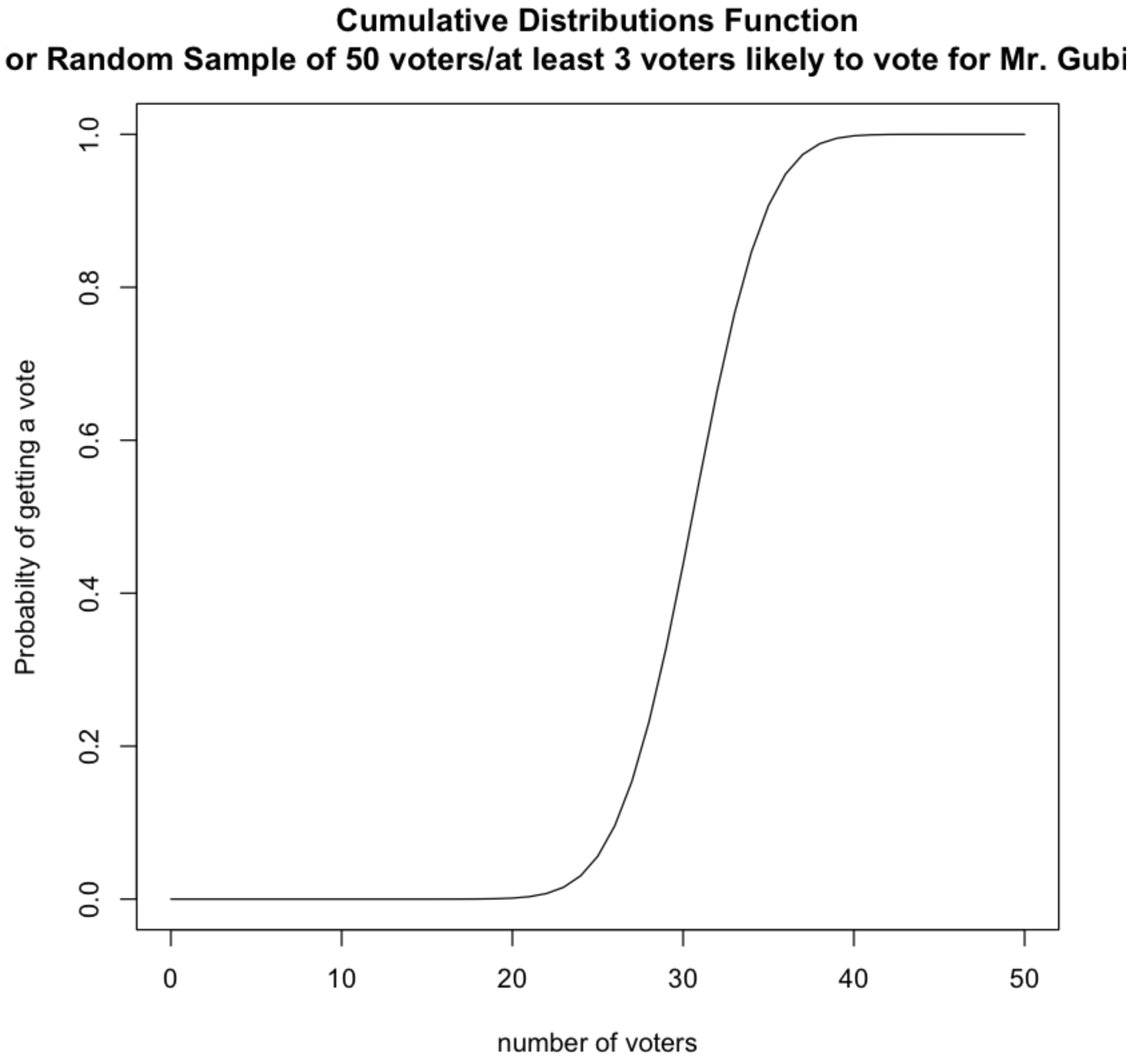
In [4]:

```
## Question 3 PMF
i <- 3
prob2 <- 0
while (i < 50) {
    prob2 = dbinom(i,50,success) + prob2
    i = i+1
}

success = 0.62
polls_pmf <- dbinom(0:50, size = 50, prob = success)
plot(polls_pmf, type = "h", main="PMF for Random Sample of
        50 voters/at least 3 voters likely to vote for Mr. Gubin
ator",xlab = "number of voters"
        ,ylab = "Probabilty of getting a vote")
points(polls_pmf,pch=1)

## Question 3 CDF
plot(0:50,cumsum(polls_pmf), type = 'l', main = "Cumulative Dist
ributions Function
 for Random Sample of 50 voters/at least 3 voters likely to vote
for Mr. Gubinator",
    xlab = "number of voters", ylab = "Probabilty of getting a v
ote")

## Question 3 Confidence Interval
library(binom)
cI_2 = binom.confint(3, 50, conf.level = 0.95, "exact")
cI_2

polls_pmf <- dbinom(0:50, size = 50, prob = success)
plot(polls_pmf, type = "l", main="PMF for Random Sample of
        50 voters/at least 3 voters likely to vote for Mr. Gubin
ator",xlab = "number of voters"
        ,ylab = "Probabilty of getting a vote")
abline(h = 0.05, xlim =c(0,1), col ='red')
legend("topleft","Confidence Interval 95%")

cat("Binomial Probability in a random sample of 50 voters to fin
d at least 3 voters to vote for Mr. Gubinator: "
    ,prob2, " \n")
```

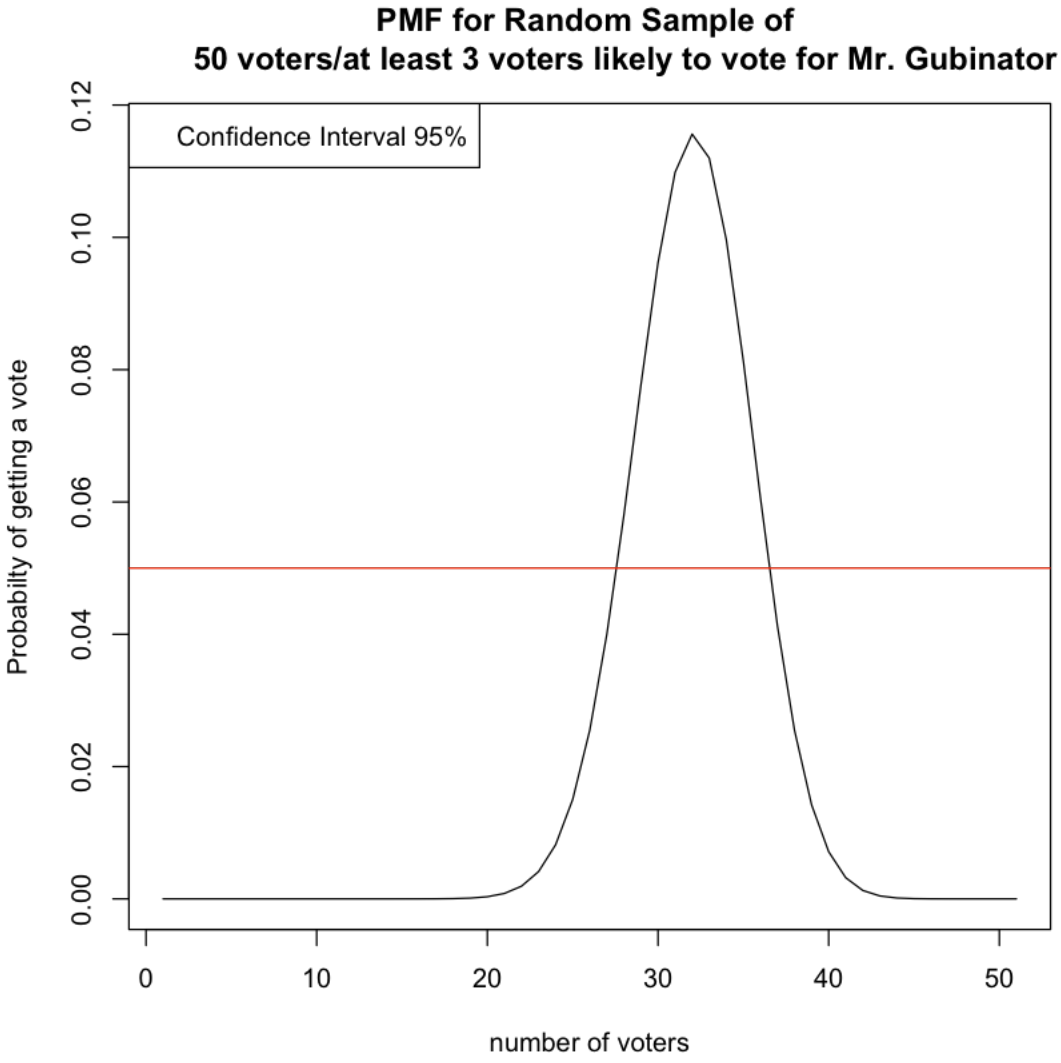


A data.frame: 1 × 6

	method	x	n	mean	lower	upper
	<fct>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	exact	3	50	0.06	0.01254859	0.1654819



```
Binomial Probability in a random sample of 50 voters
to find at least 3 voters to vote for Mr. Gubinator:
1
```



In [5]:

```
## Question 4 What is the average and standard deviation of the number of voters
#that we will find in our random sample? Hint: use the estimators of the respective distribution.
success = 0.62
mean = 50*success
cat("Average number of voters: ", mean, "\n")
std = (sqrt(50*success*(1-success)))
cat("Standard deviation of the number of voters: ", std)
```

```
Average number of voters:  31
Standard deviation of the number of voters:  3.4322
```

1. X is a continuous random variable with mean μ = 40, Find and plot the density distribution (pdf) with three different standard deviations (2,6,12) (Please draw the curves on the same figure with different colors and a legend - hint: see function lines()) for:

a. P(x < 40)

b. P(x > 21)

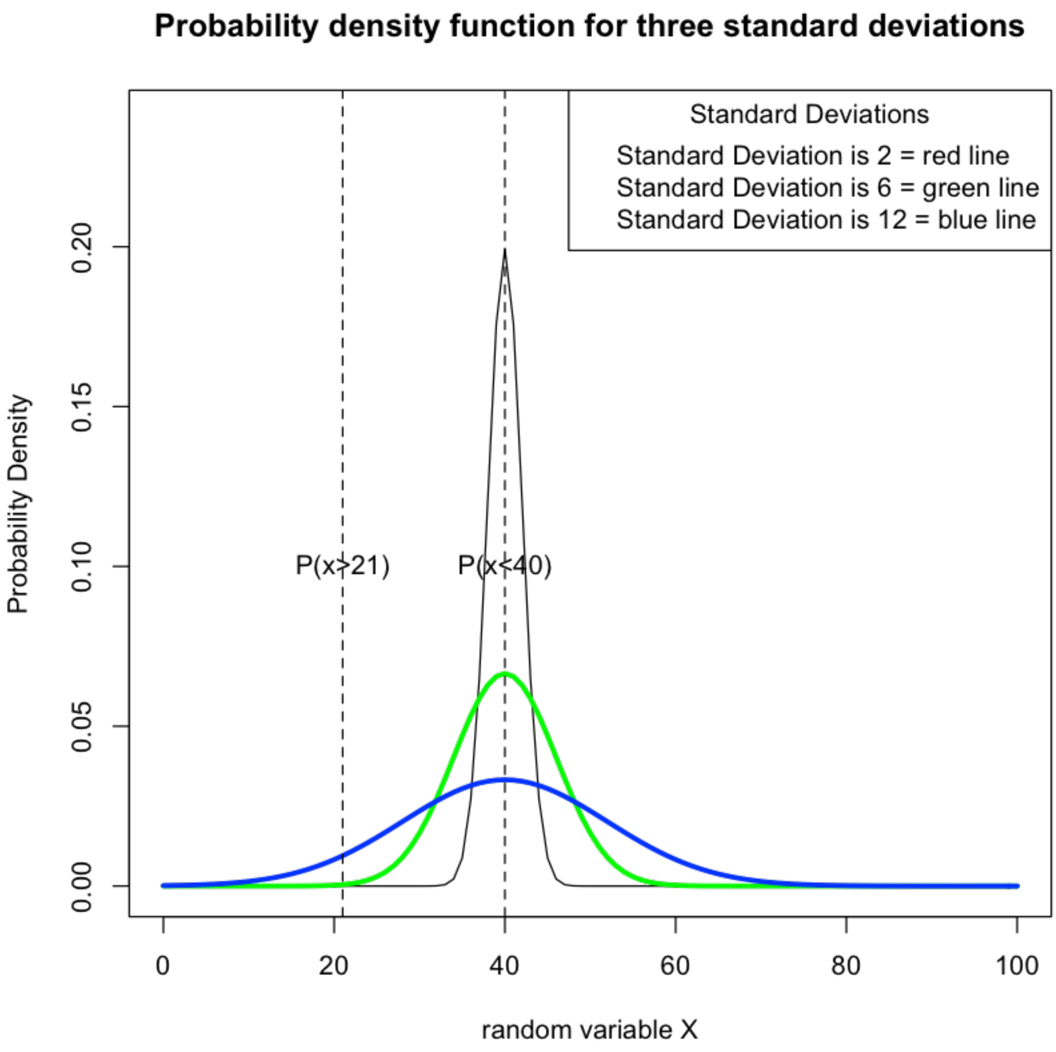
c. What conclusions can you draw from these plots regarding the effect of the SD on the distribution?

In [6]:

```
##Code
x <- 0:100

stdone<-dnorm(x,40,2)
stdtwo<-dnorm(x,40,6)
stdthr<-dnorm(x,40,12)

plot(x,stdone, type="l", ylim=c(0,1.2*max(stdone,stdtwo,stdthr))
, ylab="Probability Density",
     xlab = "random variable X", main = "Probability density fun
ction for three standard deviations")
lines(x,stdtwo, type="l", lwd=3, col="Green")
lines(x,stdthr, type="l", lwd=3, col="Blue")
abline(v = 40, col ='black', lty = 2)
abline(v = 21, col = 'black', lty = 2)
text(21,0.1, labels = "P(x>21)")
text(40, 0.1, labels = "P(x<40)")
legend("topright", title = "Standard Deviations", c("Standard De
viation is 2 = red line",
                                                    "Standard De
viation is 6 = green line",
                                                     "Standard D
eviation is 12 = blue line"))

## part a
f_ea = function(x) {dnorm(x,40,2)} ##create the function
cat("P(x<40) when Standard Deviation is 2: ")
integrate(f_ea,0,40) ##Integrate using the boundaries
```

```r
f_ea = function(x) {dnorm(x,40,6)} ##create the function
cat("P(x<40) when Standard Deviation is 6: ")
integrate(f_ea,0,40) ##Integrate using the boundaries

f_ea = function(x) {dnorm(x,40,12)} ##create the function
cat("P(x<40) when Standard Deviation is 12: ")
integrate(f_ea,0,40) ##Integrate using the boundaries

## part b
f_ea = function(x) {dnorm(x,40,2)} ##create the function
cat("P(x > 21) when Standard Deviation is 2: ")
integrate(f_ea,21,100) ##Integrate using the boundaries

f_ea = function(x) {dnorm(x,40,6)} ##create the function
cat("P(x > 21) when Standard Deviation is 6: ")
integrate(f_ea,21,100) ##Integrate using the boundaries

f_ea = function(x) {dnorm(x,40,12)} ##create the function
cat("P(x > 21) when Standard Deviation is 12: ")
integrate(f_ea,21,100) ##Integrate using the boundaries
```

```
P(x<40) when Standard Deviation is 2:

0.5 with absolute error < 3.7e-05

P(x<40) when Standard Deviation is 6:

0.5 with absolute error < 4.7e-07

P(x<40) when Standard Deviation is 12:

0.4995709 with absolute error < 1.4e-14

P(x > 21) when Standard Deviation is 2:

1 with absolute error < 1.3e-09

P(x > 21) when Standard Deviation is 6:

0.999229 with absolute error < 7.4e-06

P(x > 21) when Standard Deviation is 12:

0.943327 with absolute error < 3.1e-06
```



The larger standard deviation increased the chance of getting more values across a larger range of values. The standard deviation had no effect on the mean of the distribution. Larger standard deviation brings more uncertainity to the experiment.

3. There are 6 cars in a car shop out which 3 are defective. If 2 cars are picked randomly,

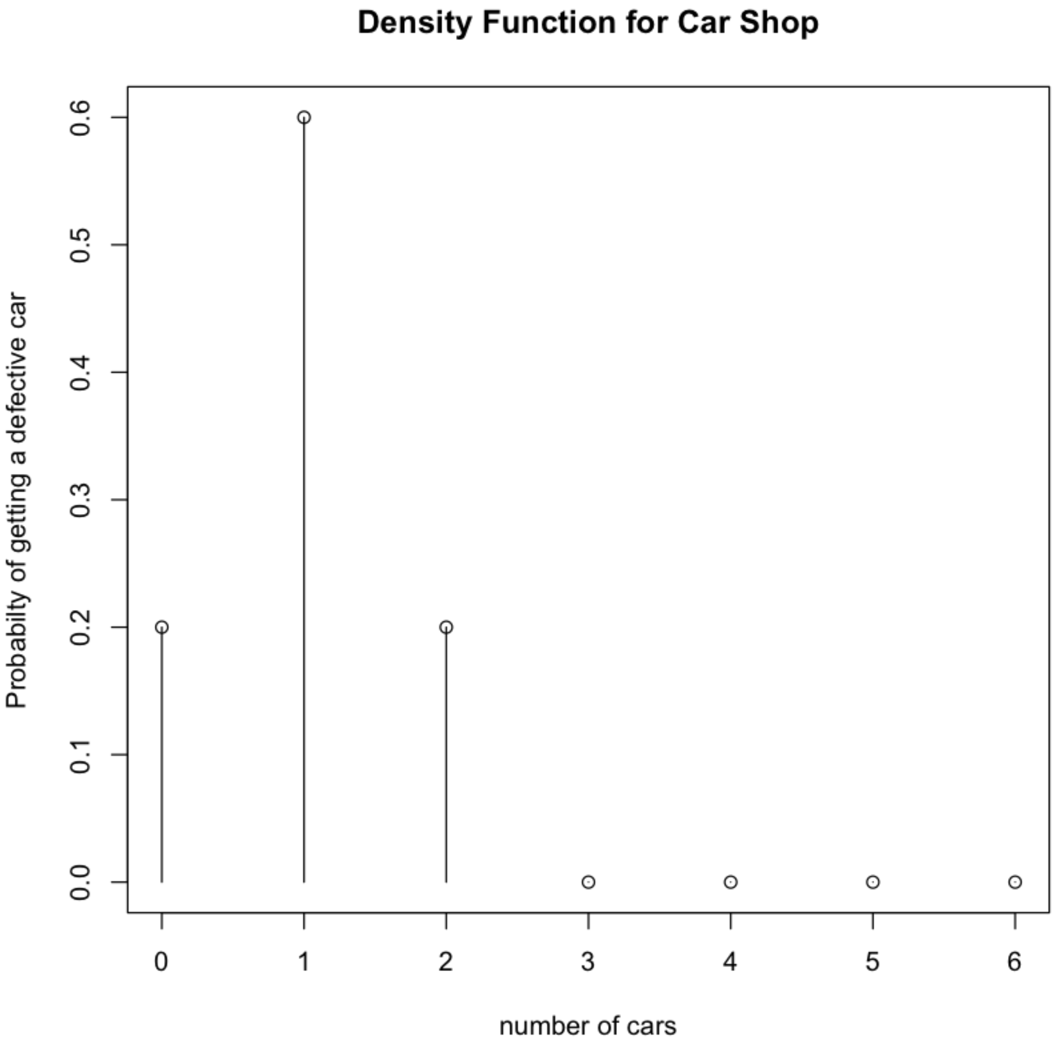
Find the probability that at least one is defective.

1. Use the density function
2. Use the R code to compare the results

In [7]:

```
##Code
x = seq(0,6,1)
polls_pmf <- dhyper(x, 3, 3, 2)
plot(x, polls_pmf, type = "h", main="Density Function for Car Shop",xlab = "number of cars"
        ,ylab = "Probabilty of getting a defective car", xlim = c(0,6))
points(x, polls_pmf,pch=1)

prob_defect = 3/6
two_cars = 1 - ((3/6)*(3/5))

cat("Density Function: ~0.7 \n")
cat("Rcode: " , two_cars)
```

```
Density Function: ~0.7
Rcode:  0.7
```



4. In the past, for every attempt to make a call there was a 70% probability of getting the call.

a. Calculate the probability of having 12 successes in 20 attempts.

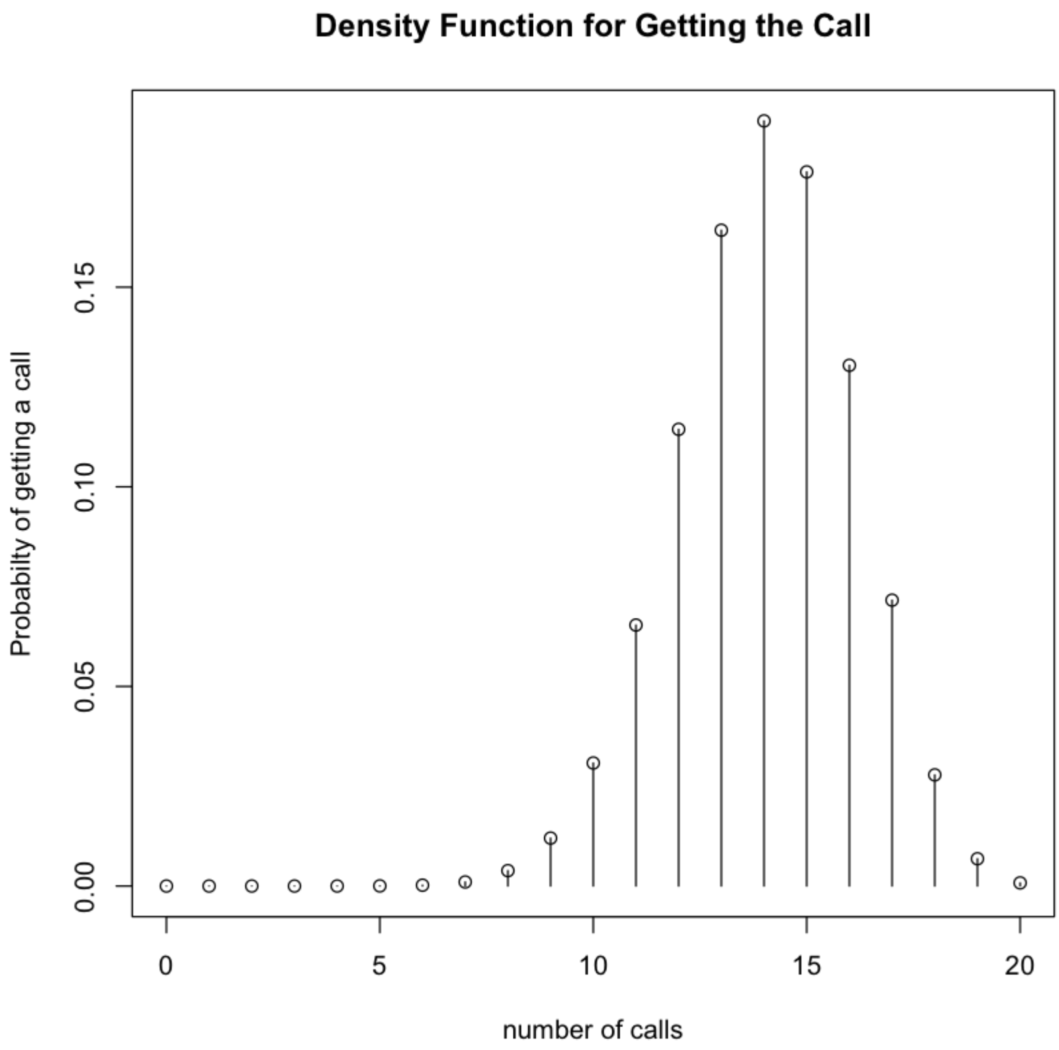
1. Use the density function
2. Use the R code to compare the results

b. Plot the distribution and describe the shape

In [8]:

```
## Code for part 1
success = 0.70
x = seq(0,20,1)
polls_pmf <- dbinom(x, size = 20, prob = success)
plot(x,polls_pmf, type = "h", main="Density Function for Getting the Call",xlab = "number of calls"
        ,ylab = "Probabilty of getting a call")
points(x,polls_pmf,pch=1)

prob_call = 0.70
twel_suc = dbinom(12,20,0.7)

ch = choose(20,12)
twel_suc2 = ch*(0.7^12)*(0.3^8)

cat("Density Function: ~0.11 \n")
cat("Rcode: " , twel_suc, " \n")
cat("Rcode: " , twel_suc2)
```

```
Density Function: ~0.11
Rcode:  0.1143967
Rcode:  0.1143967
```



5. In a company 3/4 of the females are single,

Calculate the probability that within the first 5 randomly selected females we find the first single woman?
In average in how many people we need to select before find a single female?

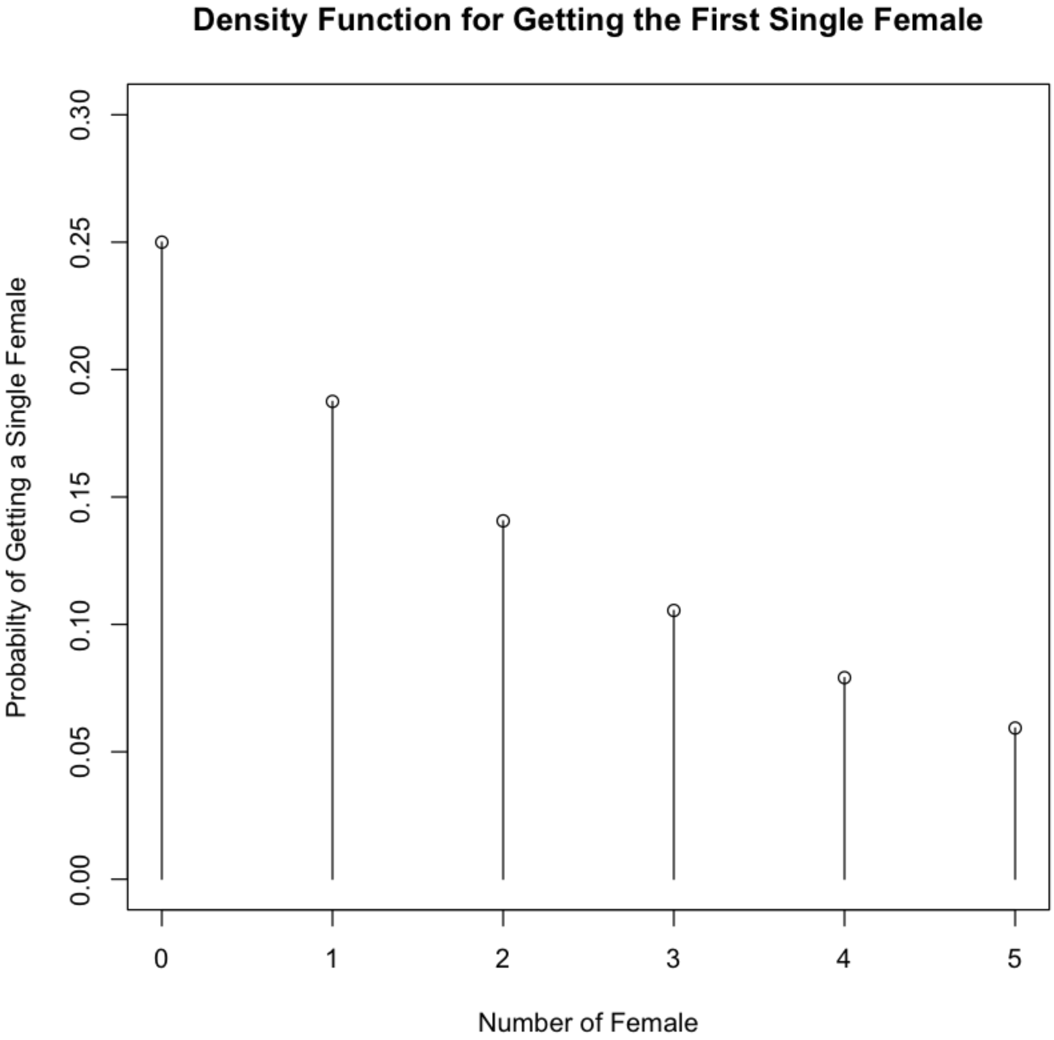
1. Use the density function
2. Use the R code to compare the results

In [9]:

```
failure_prob = 1-0.75
x = seq(0,5,1)
polls_pmf <- dgeom(x, failure_prob)
polls_pmf
plot(x,polls_pmf, type = "h", main="Density Function for Getting the First Single Female",xlab = "Number of Female"
        ,ylab = "Probabilty of Getting a Single Female", ylim = c(0,0.30))
points(x,polls_pmf,pch=1)

wom_success = sum(dgeom(x, failure_prob))

average_den = (0*0.25) + (1*0.1875) + (2*0.140625) + (3*0.10546875) + (4*0.0791015625) + (5*0.059326171875)

average = 1/0.75

cat("Density Function: ~0.82 \n")
cat("Rcode: " , wom_success, " \n")
cat("Density Function Average number of people needed to find a single female: ", average_den, " \n")
cat("Average number of people needed to find a single female: ", average)
```

0.25 ·   0.1875 ·   0.140625 ·   0.10546875 ·   0.0791015625 ·
0.059326171875

```
Density Function: ~0.82
Rcode:  0.8220215
Density Function Average number of people needed to
find a single female:  1.398193
Average number of people needed to find a single fem
ale:  1.333333
```



6.

Using the same format that I used for the other distributions, create a short explanation of the beta distribution: present the components of the distribution and give with one example of applications of this distribution. Remember: do not copy text from the resource you are using, try to understand and explain using your own words how and why is the beta distribution used. (This is useful as there many, many more distributions that you might encounter while working with statistics, and being able to generate a concise summary is a good skill)

The Beta distribution is a continuous probability distribution that has two free parameters, alpha and beta. Alpha and beta are the exponents of random variable. The values of alpha and beta determine the shape of the Beta distribution. This distribution is used to model the behavior of random variables that are limited to intervals of finite length. The Beta distribution is used in Bayesian analysis as a representation of the prior distribution for binomial, negative binomial and geometric distributions.

For example, a beta distribution would be used for a binomial distribution when the probability of success in each Bernoulli trial is random or unknown.

Optional: Choose one of the questions below.

1. Generate a tree graph that represents flipping a coin 4 times, let A be the event "the first outcome is tails" , B the event " the second outcome is head" and C the event "the third outcome is tails" calculate

p(AUBUC) = P (A U B U C) = P(A) + P(B) + P(C) - P(A ∩ B) - P(A ∩ C) - P(B ∩ C) + P(A ∩ B ∩ C)

In [10]:

```
prob_a = (0.5)
prob_b = (0.5)
prob_c = (0.5)

probAinterB = (0.5)*(0.5)
probAinterC = (0.5)*(0.5)
probBinterC = (0.5)*(0.5)
probAinterBinterC = (0.5)*(0.5)*(0.5)

probAuBuC = prob_a + prob_b + prob_c - probAinterB - probAinterC
- probBinterC + probAinterBinterC
cat("Probability of A union B union C: ", probAuBuC)
```

```
Probability of A union B union C:  0.875
```

In [11]:

```
installed.packages("ggplot2")
installed.packages("dplyr")
installed.packages("reshape2")
installed.packages("knitr")
installed.packages("igraph")
```

A matrix: 0 × 17 of type chr

Package	LibPath	Version	Priority	Depends	Imports	LinkingTo	Sugge

A matrix: 0 × 17 of type chr

Package	LibPath	Version	Priority	Depends	Imports	LinkingTo	Sugge

A matrix: 0 × 17 of type chr

Package	LibPath	Version	Priority	Depends	Imports	LinkingTo	Sugge

A matrix: 0 × 17 of type chr

Package	LibPath	Version	Priority	Depends	Imports	LinkingTo	Sugge

A matrix: 0 × 17 of type chr

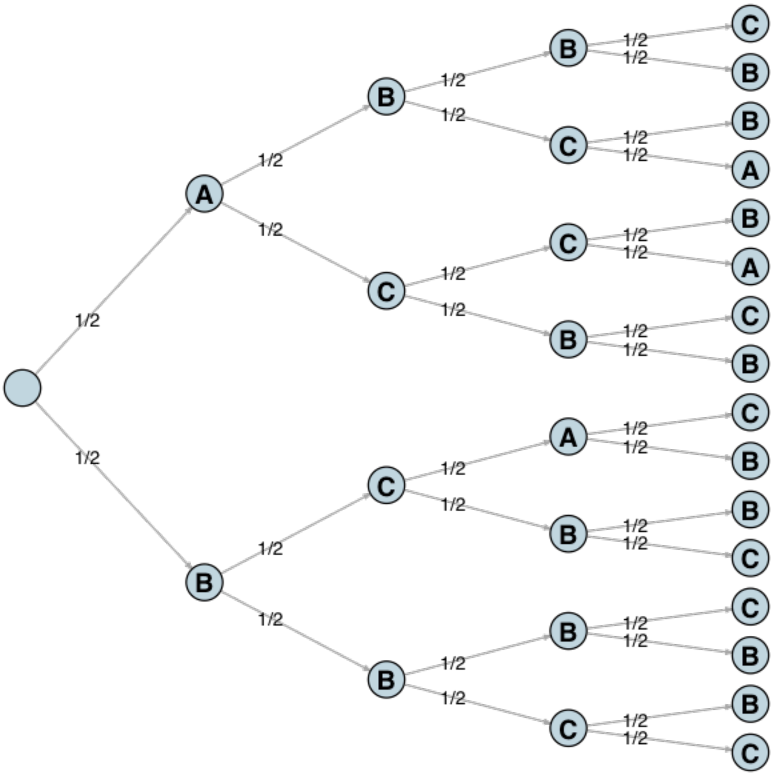
Package	LibPath	Version	Priority	Depends	Imports	LinkingTo	Sugge

In [12]:

```r
library(igraph)
g <- graph.tree(n = 2^5 - 1, children = 2)
n_1 = c("A", "B", "B")
n_2 = c("C")
n_3 = c("B")
n_4 = c("C", "C", "B")
n_5 = c("A", "B", "A")
n_6 = c("C", "B", "C")
node_labels <- c("",replicate(1,n_1), replicate(2, n_2), replicate(2, n_3), replicate(1, n_4), replicate(1, n_1),
                replicate(2, n_2), replicate(2, n_3), replicate(1,n_5), replicate(1,n_6), replicate(2, n_3),
                replicate(2, n_2),replicate(2, n_3),replicate(1, n_2))
#replicate(2,n_2)
edge_labels <- c("1/2")
edge_label2 = replicate(30,edge_labels)

#Assign Color
V(g)$color <- "#C4D8E2"
```

```
#V(g)$color[3] <- "white"
#V(g)$color[4] <- "green"

#assign position
coords <- layout_(g, as_tree())
coord2 = matrix(c(-coords[,2],-coords[,1]),ncol = 2)

plot(g,
     layout = coord2,            # draw graph as tree
     vertex.size = 10,                  # node size
     vertex.color = V(g)$color,          # node color
     vertex.label = node_labels,        # node labels
     vertex.label.cex = 1,             # node label size
     vertex.label.family = "Helvetica", # node label family
     vertex.label.font = 2,             # node label type (bold)
     vertex.label.color = '#000000',    # node label size
     edge.label = edge_label2,          # edge labels
     edge.label.cex = .7,               # edge label size
     edge.label.family = "Helvetica",   # edge label family
     edge.label.font = 1,               # edge label font type (
bold)
     edge.label.color = '#000000',      # edge label color
     edge.arrow.size = 0.2,              # arrow size
     edge.arrow.width = 1              # arrow width
)
```

```
Attaching package: ‘igraph’


The following objects are masked from ‘package:stats
’:

    decompose, spectrum


The following object is masked from ‘package:base’:

    union
```



2. From the Dataset Diabetes, construct contingency tables for the following variable combinations:

A: location Vs gender B: Gender Vs frame C: Gender Vs Age (Convert age to an discrete ordinal variable with three categories) D: Cholesterol Vs Age (Convert age and cholesterol to an discrete ordinal variable with three categories)

calculate the joint and marginal probabilities, and from the above contingency tables choose 5 conditional probability examples with the probabilities calculations and one or two sentences explaining the results.