# DSAN 5100 Lab 3

AUTHOR
Dr. Purna Gamage

## Initialization Code

Here we make sure to set R's random seed to `5100`, then import the `ggplot2` library and change its default theme (since `theme_classic()` more closely mirrors base R's plot style!)

```
set.seed(5100)
library(ggplot2)
ggplot2::theme_set(ggplot2::theme_classic())
```

## Question 1

Use either the base R `plot()` function or the `ggplot2` library to plot probability mass histograms for the following discrete distributions, using the parameter values provided within each part (to avoid ambiguity in parameter names, a link to the Wikipedia page for each distribution is provided, so that you can use the sidebar in the article to find definitions for each parameter).

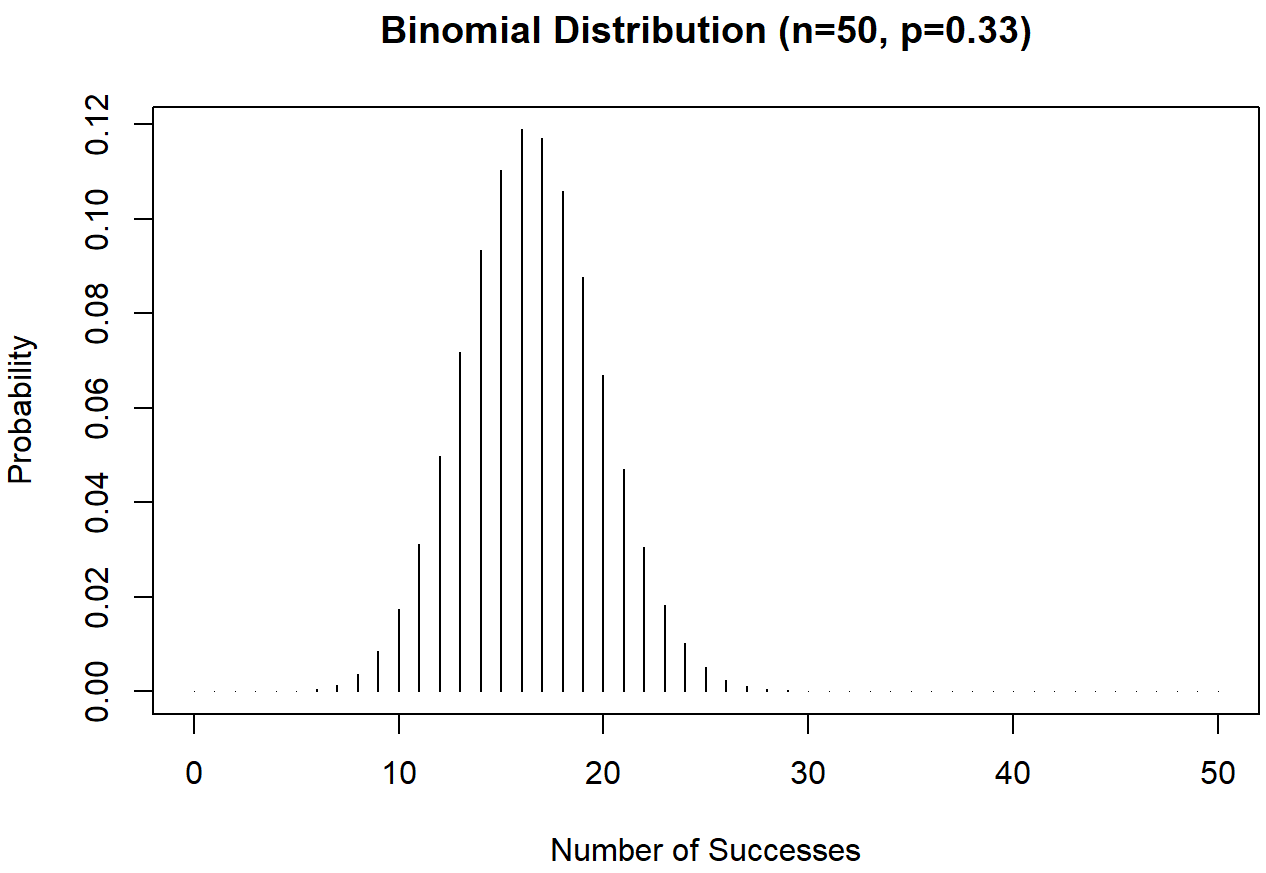
To help you see what we're looking for, the solution to Question 1.1 is already provided for you, using both the base R `plot()` function as well as the `ggplot2` library. Note that you do **not** need to generate two histogram plots like this for each subsequent question! The examples are there to demonstrate the two possible approaches.

(*Hint*: For each distribution, you'll need to find the appropriate `d<distribution name>()` function.).

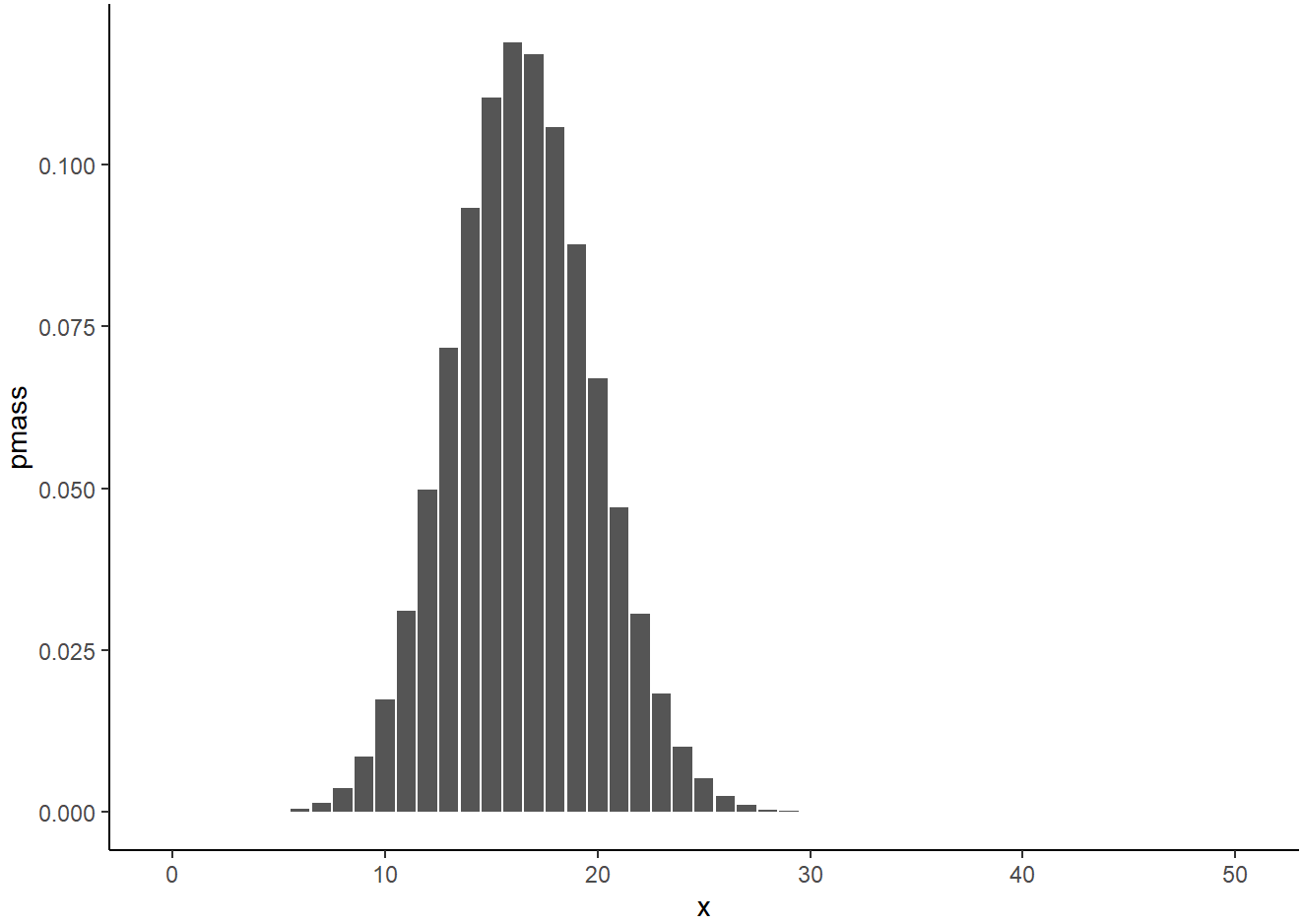
### Question 1.1: Binomial Distribution

Construct a plot of the probability mass histogram for a Binomially-distributed random variable $X$, with parameter values $n = 50$ and $p = 0.33$.

```
# Your code here
x <- 0:50
masses <- dbinom(x, size=50, prob=0.33)
plot(x, masses, type="h", xlab="Number of Successes", ylab="Probability", main="Binomial Distribu
```




```
binom_df <- tibble::tibble(
  x = 0:50,
  pmass = dbinom(x, size=50, prob=0.33)
)
binom_df |> ggplot(aes(x=x, y=pmass)) +
  geom_bar(stat='identity')
```




For the remaining parts of the question, you only need to produce **one** plot, using either the base R `plot()` function **or** the `ggplot2` library.

## Question 1.2: Discrete Uniform

Construct a plot of the probability mass histogram for a Discrete Uniform random variable $X$, with parameter values $a = 10$ and $b = 20$.

To make this and subsequent questions a bit easier, you can install the `extraDistr` package by executing the following R command:

```
# Set CRAN mirror
options(repos = c(CRAN = "https://cloud.r-project.org"))

# Install extraDistr package if not already installed
if (!require(extraDistr)) {
  install.packages("extraDistr")
}
```

```
Loading required package: extraDistr
```

And then you can use the additional functions provided by this library by starting your R code cell with:

```
library(extraDistr)
```

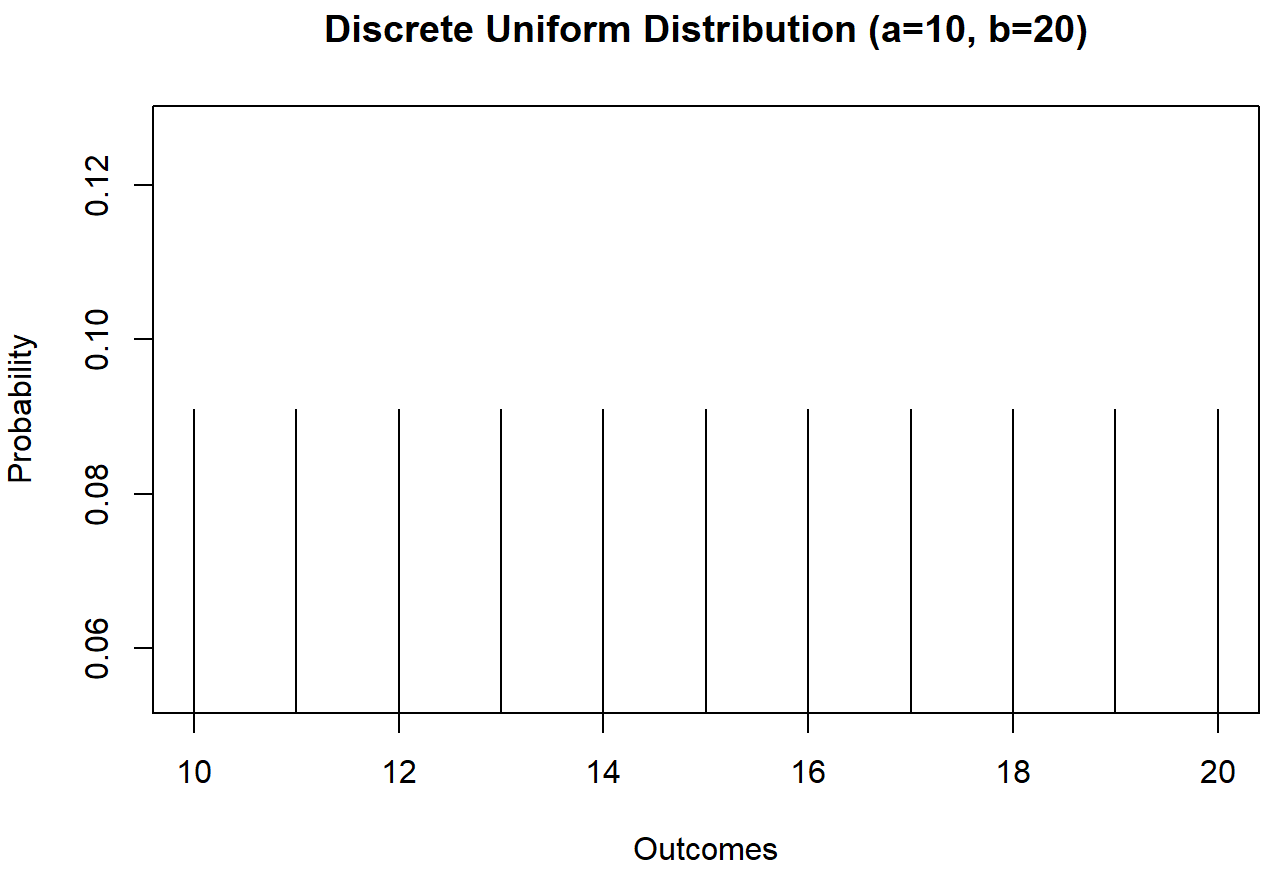
```
library(extraDistr)

# Define parameters
a <- 10
b <- 20

# range of possible outcomes (10 to 20)
x <- a:b

# probability masses using the discrete uniform distribution
masses <- ddunif(x, min = a, max = b)

plot(x, masses, type="h", xlab="Outcomes", ylab="Probability",
     main="Discrete Uniform Distribution (a=10, b=20)")
```




## Question 1.3: Bernoulli Distribution

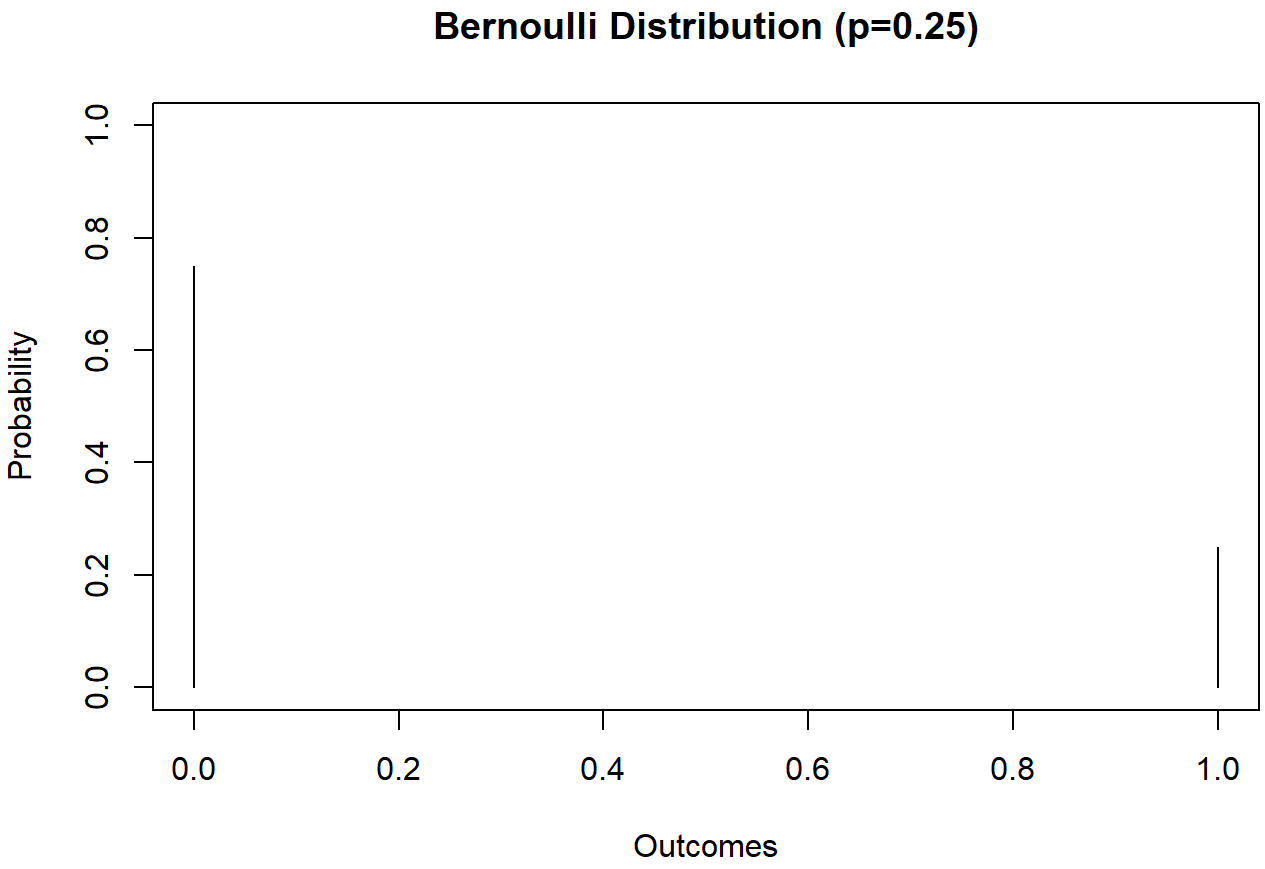
Construct a plot of the probability mass histogram for a Bernoulli random variable $X$, with parameter value $p = 0.25$.

```r
# Parameter for Bernoulli distribution
p <- 0.25

# Outcomes are 0 and 1
x <- c(0, 1)

# probability masses for 0 and 1
masses <- c(1 - p, p)

#using base R
plot(x, masses, type="h", xlab="Outcomes", ylab="Probability",
     main="Bernoulli Distribution (p=0.25)", ylim=c(0, 1))
```




## Question 1.4: Poisson Distribution

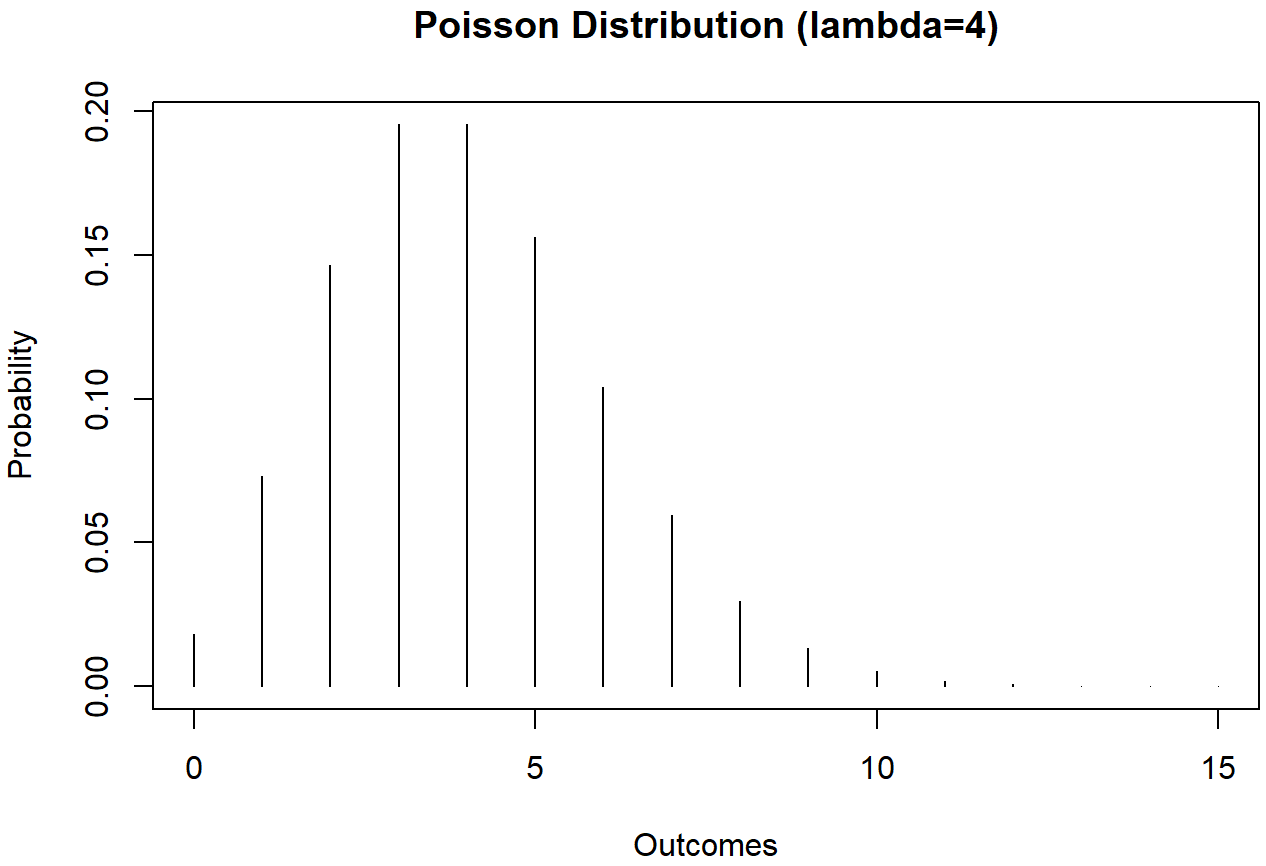
Construct a plot of the probability mass histogram for a Poisson random variable $X$, with parameter value $\lambda = 4$.

```r
lambda <- 4

x <- 0:15

masses <- dpois(x, lambda = lambda)
```

```
plot(x, masses, type="h", xlab="Outcomes", ylab="Probability",
     main="Poisson Distribution (lambda=4)")
```




## Question 1.5: Geometric Distribution

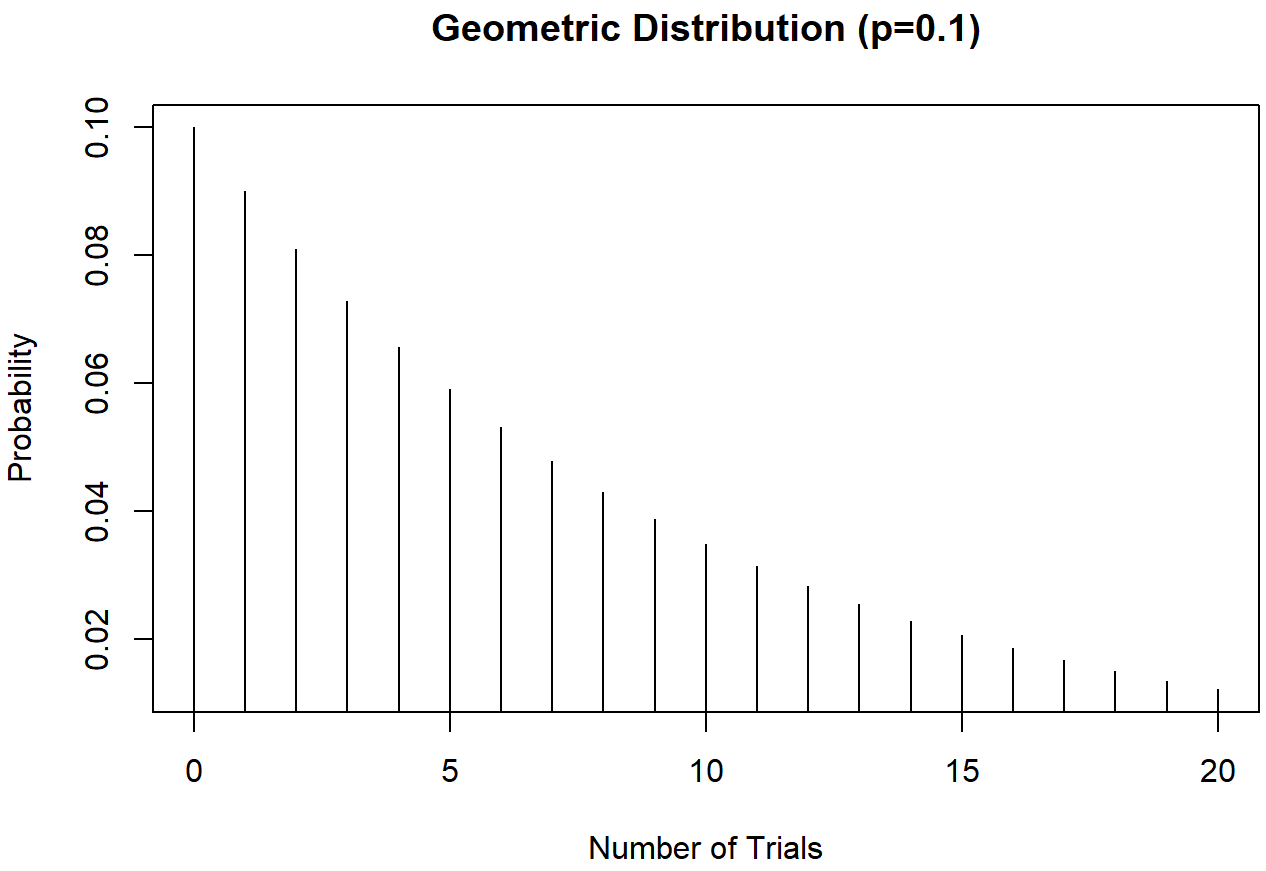
Construct a plot of the probability mass histogram for a Geometric random variable $X$, with parameter values $p = 0.1$. Please use the **first** interpretation in the Wikipedia article, whereby $X$ is the number of **trials** required before obtaining a success.

```
p <- 0.1

x <- 0:20

masses <- dgeom(x, prob = p)

plot(x, masses, type="h", xlab="Number of Trials", ylab="Probability",
     main="Geometric Distribution (p=0.1)")
```




## Question 1.6: Hypergeometric Distribution

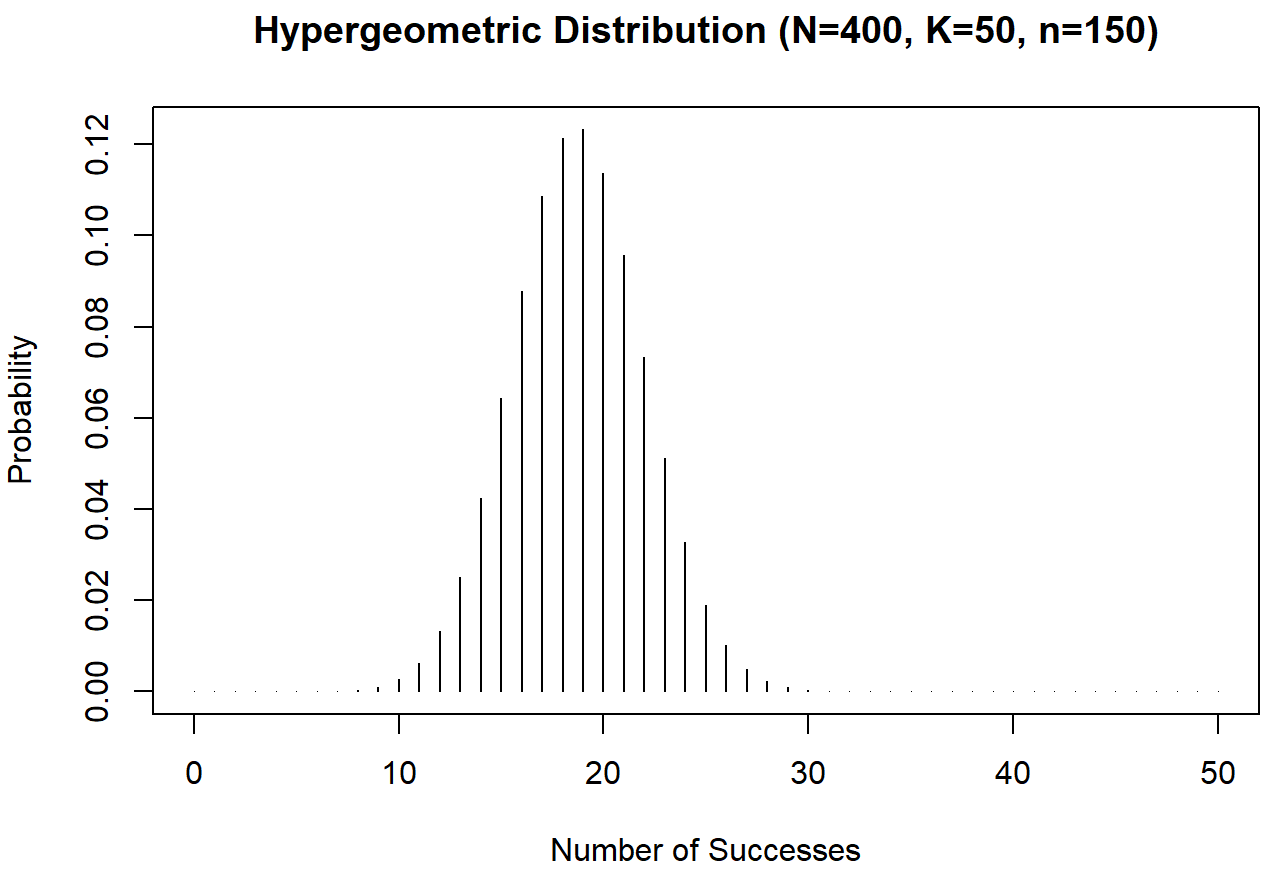
Construct a plot of the probability mass histogram for a Hypergeometric random variable $X$, with parameter values $N = 400$, $K = 50$, and $n = 150$.

```
# Parameters for Hypergeometric distribution
N <- 400
K <- 50
n <- 150

# (number of successes in n draws)
x <- 0:min(K, n)

#probability masses using the Hypergeometric distribution
masses <- dhyper(x, m = K, n = N - K, k = n)

plot(x, masses, type="h", xlab="Number of Successes", ylab="Probability",
     main="Hypergeometric Distribution (N=400, K=50, n=150)")
```




## Question 1.7: Negative Binomial Distribution

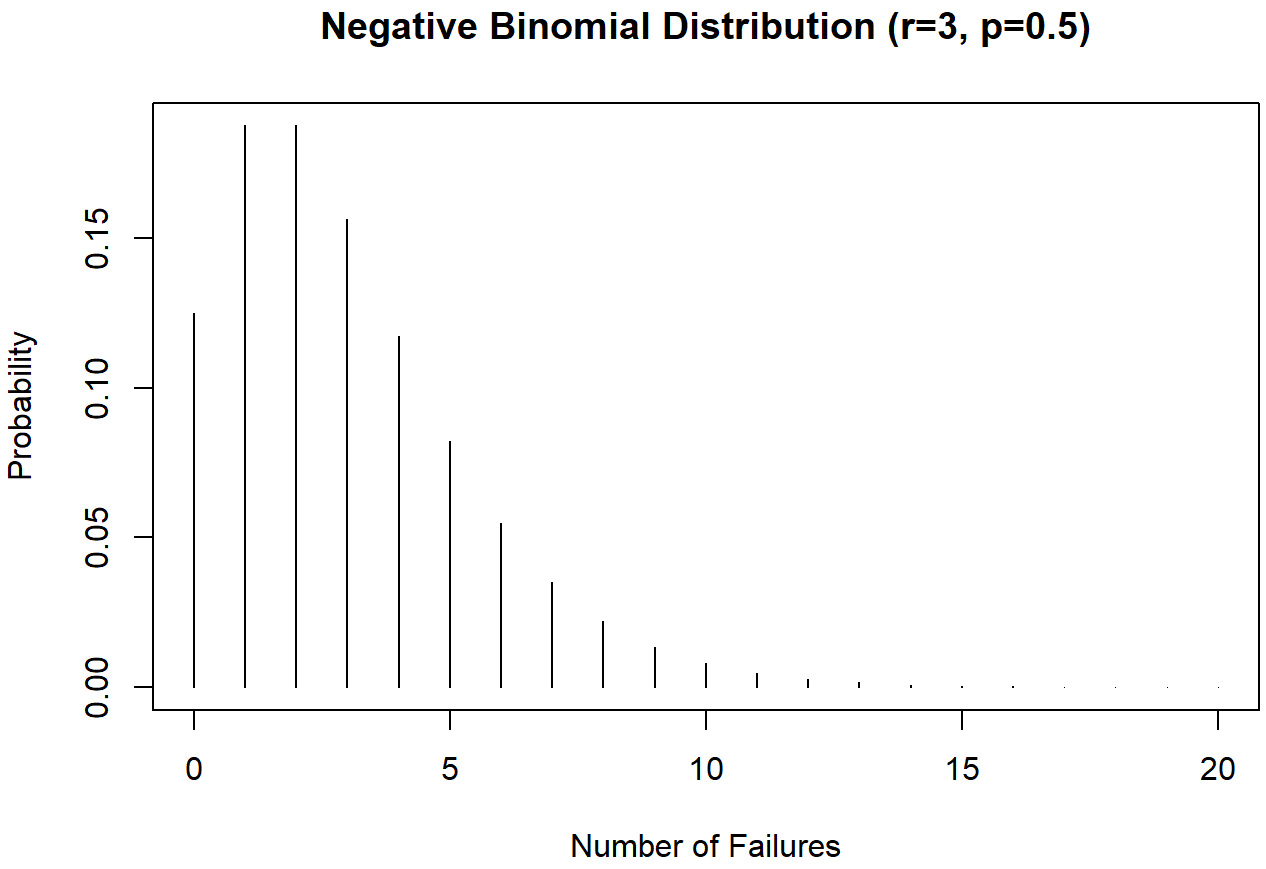
Construct a plot of the probability mass histogram for a Negative Binomial random variable $X$, with parameter values $r = 3$ and $p = 0.5$.

```
# Negative Binomial distribution
r <- 3
p <- 0.5

# failures ranging from 0 to 20)
x <- 0:20

#  probability masses using the Negative Binomial distribution
masses <- dnbinom(x, size = r, prob = p)

plot(x, masses, type="h", xlab="Number of Failures", ylab="Probability",
     main="Negative Binomial Distribution (r=3, p=0.5)")
```




## Question 2

Suppose there are fifteen multiple choice questions in `DSAN-5100` midterm test. Each question has four possible answers, and only one of them is correct. Find the probability of having four or less correct answers if a student attempts to answer every question at random, in the following two ways:

### Question 2.1

Compute the probability of having exactly 4 correct answers by random attempts using `dbinom()` and `pbinom()`.

```
#  binomial distribution
n <- 15
p <- 0.25
x <- 4

# probability of getting exactly 4 correct answers
prob_exact_4 <- dbinom(x, size = n, prob = p)

# cumulative probability of getting 4 or fewer correct answers
prob_4_or_less <- pbinom(x, size = n, prob = p)
```

```
prob_exact_4
```

```
[1] 0.2251991
```

```
prob_4_or_less
```

```
[1] 0.6864859
```

## Question 2.2

Find the the probability of having four or less correct answers by random attempts using `dbinom()`.

```
# Parameters for  binomial distribution
n <- 15
p <- 0.25
x <- 0:4

# probability of getting  0 to 4 correct answers
prob_4_or_less <- sum(dbinom(x, size = n, prob = p))

# Display  result
prob_4_or_less
```

```
[1] 0.6864859
```

## Question 2.3

Compute the above probability (from Question 2.2) using `pbinom()`.

```
# Parameters for binomial distribution
n <- 15
p <- 0.25
x <- 4

# cumulative probability of getting 4 or fewer correct answers
prob_4_or_less <- pbinom(x, size = n, prob = p)

# Display the result
prob_4_or_less
```

```
[1] 0.6864859
```

# Question 3

Use R to compute the following probabilities.

## Question 3.1

Assume an insurance company receives 3 motor vehicle insurance claims per week. What is the probability $\Pr(X \leq 11)$ that they recieve 11 or fewer claims during a month?

```
# Poisson rate for a month
lambda_month <- 3 * 4

# probability of receiving 11 or fewer claims
prob_11_or_less <- ppois(11, lambda = lambda_month)

prob_11_or_less
```

```
[1] 0.4615973
```

## Question 3.2

While you are at the Georgetown library terrace, you notice that airplanes fly at an average rate of 1 every 4 hours. What is the probability that you will see at least one plane in the next hour?

```
# Poisson rate per hour
lambda_per_hour <- 1 / 4

# probability of seeing at least one plane
prob_at_least_one <- 1 - ppois(0, lambda = lambda_per_hour)

prob_at_least_one
```

```
[1] 0.2211992
```

# Question 4

Try using the `nbinom()` function to solve this example. (This relates to Problem-1 in the Lab-1 Assignment)

Mike decides to flip a coin until obtaining three successes (heads). Treating each coin flip as a trial, the successes occurred in trials 6, 8, and 9. This means that he had to perform 9 trials (coin flips) in total to obtain 3 successes (heads) in total.

Mathematically, we can model this procedure by defining a Random Variable $X$, whose distribution depends on a parameter $p$: $X$ will represent the number of total coin flips that are required before obtaining 3 successes, for a given success probability $p$.

Given the observed data (9 coin flips required for 3 successes), do you think that Mike is using a fair coin, so that his probability of heads is $p = 0.5$? Or is it an unfair coin with $p > 0.5$? Can a simulation give an answer? Let's try.

## Question 4.1

If Mike's success probability is $p = 0.5$ What is the probability that he will obtain these 3 successes within 9 trials?

```
r <- 3
n <- 9
p <- 0.5

prob_3_successes_9_trials <- dnbinom(n - r, size = r, prob = p)

prob_3_successes_9_trials
```

```
[1] 0.0546875
```

## Question 4.2

Run $N = 10000$ simulations with this success probability ($p = 0.5$), and use the results to estimate this same probability $\Pr(X = 6)$ *Hint: Use* `rnbinom()`

```
# parameters
r <- 3
p <- 0.5
N <- 10000

# 10,000 simulations, generating the number of failures before achieving 3 success
simulations <- rnbinom(N, size = r, prob = p)

# probability of getting exactly 6 failures
estimated_prob <- mean(simulations == 6)

estimated_prob
```

```
[1] 0.0567
```

## Question 4.3

If Mike's success probability $p$ was in fact greater than $0.5$ he would not need a lot of trials before obtaining 3 successes.

Using the two different approaches described below, find the probability $\Pr(X \geq 6)$ that three successes were reached after 9 tosses or more by somebody with success probability $p = 0.5$.

### Question 4.3.1

First, calculate the probabiliy using `dnbinom()` and `pnbinom()`.

```
# Negative Binomial distribution
r <- 3
p <- 0.5
```

```
# cumulative probability of getting fewer than 6 failures

prob_less_than_6 <- pnbinom(5, size = r, prob = p)

# probability of getting 6 or more failures (which corresponds to 9 or more trials)
prob_6_or_more <- 1 - prob_less_than_6

prob_6_or_more
```

```
[1] 0.1445312
```

## Question 4.3.2

Now, calculate this probability without using `dnbinom()`, by simulating $N = 10000$ different runs of Mike's procedure.

```
# parameters
r <- 3
p <- 0.5
N <- 10000

# 10,000 simulations, generating the number of failures before achieving 3 successes
simulations <- rnbinom(N, size = r, prob = p)

# Eprobability of getting 6 or more failures (which corresponds to 9 or more trials)
estimated_prob_6_or_more <- mean(simulations >= 6)

estimated_prob_6_or_more
```

```
[1] 0.1447
```

## Question 4.3.3

Is this probability (part b) the same as you got using `myattempts()`: Lab 1 Assignment Problem 1 part 3?

> result obtained in this assignment being 0.1447, differs significantly from when i used the myattemps function where i got 0.0018. This could mean that the two methods are looking at slightly different situations, or that there's some randomness in the simulation. Because the difference is so big, it might help to double-check the setup or try running more simulations to get a more accurate result.