Prerequisite

# Good practice to remove all objects from the workspace
rm(list = ls())

# Use library() for packages you need, or source() for other R files.
library(tidyverse)

# Setting the seed ensures that we get the same random draw over and over again.
set.seed(20201009)

rnorm(5) # Check

## [1] 0.8315079 -0.9818884 1.1522644 -0.4687453 -0.8344489

0. Calculate the following operations by hand (... meaning by R)

a)
\[ \sum_{i=1}^{5} i = \]

b)
\[ \prod_{i=1}^{5} i = \]

c)
\[ 5! \times 10^3 \times e^4 = \]

# a)
sum(1:5)

## [1] 15

# b)
prod(1:5)

## [1] 120

#Check
1*2*3*4*5

## [1] 120
# c)

factorial(5)

## [1] 120

5*4*3*2*1

## [1] 120

factorial(5) * 10^factorial(3) * exp(4)

## [1] 6551778004

1. Build a Bernoulli distribution using the sample() function, where the probability of “success” is 0.7. Run “?sample” if you are unsure how the function works.

```r
# Create an imaginary person to flip the coin once for you
sample(x = c(0, 1),
       size = 1,
       prob = c(0.3, 0.7))

## [1] 1
```

2. How do you know if it is working properly? Conduct simulation to check if the assigned probabilities are matched with the empirics

```r
# Specify the number of simulations
sims <- 10000

# Specify the probability
ProbSuccess <- 0.7

# Create an empty vector as "container"
BernResult <- vector(mode = "numeric", length = sims)

# For loop
for (i in 1:sims) {
  BernResult[i] <- sample(c(0, 1),
                          size = 1,
                          prob = c(1 - ProbSuccess, ProbSuccess))
}

mean(BernResult)
```
```r
## [1] 0.6947

## Faster way w/o loop...but we want to build the intuition
sample(c(0, 1),
       size = sims,  
       replace = TRUE,  
       prob = c(0.3, 0.7)
)

## Or use rbinom (rmb: bernoulli is just a special case of binomial when N = 1)
rbinom(sims,  
       size = 1,  
       prob = 0.7)
```

3. Plot the above Bernoulli distribution

```r
# Base graphics
hist(BernResult)
```

![Histogram of BernResult](image)

```r
# ggplot2
BernDF <- tibble(Outcome = BernResult)

ggplot(BernDF, aes(x = Outcome)) +
```
# Adding: `aes(y = stat(count / sum(count)))` in `geom_hist`; What does it do?

```r
ggplot(BernDF, aes(x = Outcome)) +
  geom_histogram(aes(y = stat(count/sum(count)))) +
  scale_x_continuous(breaks = c(0, 1), expand = c(1, 0)) +
  labs(y = "Prob", x = "x",
       subtitle = "Bernoulli Distribution: Probability") +
  theme_bw()
```
# y = stat() means...
BernResult %>%
as_tibble %>%
count(value) %>%
mutate(TotalNum = sum(n),
    Prob = n/TotalNum)

## # A tibble: 2 x 4
##   value n TotalNum  Prob
##    <dbl> <int>      <int> <dbl>
## 1     0    3053  10000 0.305
## 2     1    6947  10000 0.695

4. Based on the above, generate a binomial distribution, with number of trials equal to 10, without using rbinom()

# Create an imaginary person to flip the coin ten times for you
# Let’s test it outside of the loop:
sample(c(0, 1),
 size = 10,
 replace = TRUE,
 prob = c(1- ProbSuccess,
         ProbSuccess)
)

# Create number of simulations and an empty vector as container

```r
BinoResult <- vector(mode = "numeric", length = sims)
```

```r
for (i in 1:sims) {

# Create an imaginary person to flip the coin ten times for you
flips <- sample(c(0, 1),
    size = 10,
    replace = TRUE,
    prob = c(1 - ProbSuccess, ProbSuccess))

# Sum up the number of "success" for that person
count <- sum(flips)

# Store it into the container; repeat 10,000
BinoResult[i] <- count
}
```

# Faster way w/o loop and sample()...but we want to build the intuition

```r
# rbinom(n = sims, 
# size = 10, 
# prob = ProbSuccess)
```

5. Plot the above binomial distribution

```r
# Base graphics
hist(BinoResult)
```
# ggplot2

BinoResult %>%
  as_tibble %>%  # Convert to data frame
  ggplot(aes(x = value)) +
  geom_histogram(aes(y = stat(count/sum(count)))) +
  scale_x_continuous(breaks = 1:10) +
  labs(y = "Prob", x = "x",
       subtitle = "Binomial Distribution") +
  theme_bw()

## 'stat_bin()' using ‘bins = 30’. Pick better value with ‘binwidth’. 
6. Explore the `rbinom`, `dbinom`, `pbinom` functions. What do they do? Answer the following questions:

a) The probability of a coin landing on head is 0.7. If you were to flip the coin 10 times, what is the probability of getting exactly 7 heads?

b) What is the probability of getting 7 heads or less?

c) How do you know (b) is true?

```r
da) Pr(exactly 7 heads) \to PDF
``` 

dbinom(x = 7, size = 10, prob = 0.7)

```
## [1] 0.2668279
```

```r
# b) Pr(7 heads or less) \to CDF
``` 
pbinom(q = 7, size = 10, prob = 0.7)

```
## [1] 0.6172172
``` 

```r
pbinom(q = 7, size = 10, prob = 0.7, lower.tail = FALSE)
```

```
## [1] 0.3827828
```
```r
1 - pbinom(q = 7, size = 10, prob = 0.7)
```

```r
## [1] 0.3827828
```

```r
# c) Double check
dbinom(x = c(0:7), size = 10, prob = 0.7) %>%
sum()
```

```r
## [1] 0.6172172
```

### How to plot?

```r
# PDF
# Create x axis in data.frame
tibble(x = seq(from = 0, to = 10, by = 1)) %>%
# Create y axis
mutate(y = dbinom(x = x, size = 10, prob = 0.7)) %>%
# Plot, map data in aesthetic
ggplot(aes(x=x,y=y))+
# Specify the type of plot
geom_line()+
# Specify x axis breaks
#scale_x_continuous(breaks = 0:10)+
scale_x_continuous(breaks = 0:10)+
# Add vertical line at x = 0.7
geom_vline(xintercept = 7, color = "red", linetype = "dashed")+
# Change label on y axis
labs(y="Prob", subtitle = "PDF")+
# (Optional: specify theme )
theme_bw()
```
# CDF

Create x axis in data.frame
tibble(x = seq(from = 0, to = 10, by =1)) %>%

Create y axis
mutate(y = pbinom(q = x, size =10, prob=.7)) %>%

Plot, map data in aesthetic
 ggplot(aes(x=x,y=y)) +

Specify the type of plot
 geom_line() +

Specify x axis breaks
 scale_x_continuous(breaks = 0:10) +

Add vertical line at x = 0.7
 geom_vline(xintercept = 7, color = "red", linetype = "dashed") +

Change label on y axis
 labs(y="Prob", subtitle = "CDF") +

(Optional: specify theme )
 theme_bw()