MA 354: Data Analysis – Fall 2021 – Due 10/8 at 5p
Homework 2:

Complete the following opportunities to use what we've talked about in class. These questions will be graded for correctness, communication and succinctness. Ensure you show your work and explain your logic in a legible and refined submission.

The starting jobs will be applied in alphabetical order (last name) for question two.

1. **Solver:** provide a solution, if possible, and reasoning for the solution. **Due to group 10/5 or earlier.**
2. **Code Checker:** provides a first check of the solver's worked solutions and ensures they are correct with a solid interpretation.
3. **Checker** checks the solution for completeness, proposes and implements changes if agreed upon by the group. Provides a short paragraph summarizing the discussion of proposals and their reason for acceptance or non-acceptance.
4. **Double Checker** checks the solution for completeness, communication and polish. The Double Checker ensures that the solution is correct and highly polished for submission.

For subsequent questions student roles will move down one position. The rolls change as follows.

1. **Solver $\Longrightarrow$ Code Checker**
2. **Code Checker $\Longrightarrow$ Checker**
3. **Checker $\Longrightarrow$ Double Checker**
4. **Double Checker $\Longrightarrow$ Solver**

While students have assigned jobs for each question I encourage students to help each other complete the homework in collaboration.

1. Select a continuous distribution (Not the uniform or exponential). It does not have to be one that we cover in the notes! To explore the PDF of your distribution, specify two sets of parameter(s) for your distribution.

   (a) **History** Discuss what types of random variables are modeled with your distribution. Be sure to include a discussion about the support and ensure to provide the density function, and CDF. This requires some internet research – what's the history of the distribution, why was it created and named? What are some exciting applications of this distribution?

   Cite all of your sources in LaTeX by adding a BibTeX citation to the .bib file. To help, I've cited R (R Core Team, 2021) in parentheses here. R Core Team (2021) provides helpful tools for the rest of the questions below. BibTeX citations are available through Google Scholar by clicking the cite button below the article of interest and selecting the BibTeX option.

   Solution:
   Our group has chosen to use a normal distribution for the `Q1` and `Q2`!
   To start with, I will define what a normal distribution is. A continuous variable X has a normal distribution — with mean μ and σ^2 — X N(μ, σ^2), if it has the following properties:

$$\mu \in \mathbb{R}; \sigma \in \mathbb{R}^+ \tag{\textbf{Parameters}}$$

$$X = \{x : x \in \mathbb{R}\} \tag{\textbf{Support}}$$

$$f_X(x|\mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \tag{\textbf{PDF}}$$

$$F_X(x|\mu, \sigma) = \int_{-\infty}^{\infty} \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \tag{\textbf{CDF}}$$

   Description: Normal distribution is one of the most important distributions in the world of statistics, and it's used in many fields. The graph of this probability function is a symmetric, bell-shaped curved. Symmetry of this function implies that its mean is going to be equal to its median value. We are going to prove it later on.
   History: One of the first major great discoveries in the field of statistics happened in 1713, as Jacob Bernoulli published his work on proving the Weak Law of Large Numbers. According to Patel and Read (1996), initially normal distribution appeared in 1733 as an approximation to the probability for sums of binominally distributed quantities to lie between two values.
   Today: According to Ahsanullah et al. (2014), normal distribution plays an important role in many applied problems in biology, economics, engineering, genetics, hydrology, mechanics, medicine, number theory, statistics, physics, psychology and so on.

   (b) Show that you have a valid PDF. You will find the `integrate()` function in `R` helpful.

   Solution:
   Here's the equation of PDF for normal distribution!

$$f_X(x|\mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \tag{\textbf{PDF}}$$

   Now, let's prove that it's valid!
   Since normal (or gaussian — whatever you prefer!) distribution is a continous probability distribution, it implies that the area under the curve is equal to 1 (or 100%!). This statement takes its roots from the second Kolmogorov axiom that states that the entirety of sample space is equal to one.
   Therefore, let's use CDF to prove that our PDF formula is going to return one! Since the support for normal distribution contain all rational numbers, our integral is going to be from negative infinity to positive infinity.

$$F_X(x|\mu, \sigma) = \int_{-\infty}^{\infty} \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \tag{\textbf{CDF}}$$

   I am going to use `integrate()` function in order to compute this equation.

```r
mean<-1 #mu
sd<-3 #sigma
pdf <- function(x){
  (1/(sd*sqrt(2*pi)))*exp((-(x-mean)^2)/(2*sd^2)) #our PDF
}
integrate(pdf, -Inf, Inf) #CDF for total area under the curve

## 1 with absolute error < 2.1e-07
```

(c) Find the median for your two sets of parameter(s). Conduct some research to find the median based on our PDF to confirm that your numerical approach is correct.

Solution:
As we have established in part `(a)` of this problem, one of the key features of the normal distribution is the fact that it's symmetrical. Let's set out to prove it through the direct proof!
Let `m` be median!

$$P(X \leq m) = P(X \geq m) = \frac{1}{2} \qquad \textbf{(Definition of the median)}$$

Let normal distribution be symmetric. If it's symmetric, then the statement $\mu = m$ holds true.
Therefore, the following equation should also hold true:

$$\int_{-\infty}^{\mu} \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} = \int_{\mu}^{\infty} \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} = \frac{1}{2} \qquad \textbf{(Assumption)}$$

Let's check it through R!

```
        mean<-1 #mu
        sd<-3 #sigma
        func <- function(x){
          (1/(sd*sqrt(2*pi)))*exp((-(x-mean)^2)/(2*sd^2))
        }
        firstPart <- integrate(func, -Inf, mean)
        secondPart <- integrate(func, mean, Inf)

        (firstPart$value)

## [1] 0.5

        (firstPart$value==secondPart$value)

## [1] TRUE
```

It would appear that $\mu = m$! Therefore, mean is equal to median within the normal distribution. Therefore, since $\mu_1 = 1$ and $\mu_2 = -2$, the medians are 1 and -2 respectively!

(d) Graph the PDF for several values of the parameter(s) including the two sets you specified. What does changing the parameter(s) do to the shape of the PDF?

Solution:
Let's take various values and plot different PDFs! I am going to use the ggplot2 (Wickham, 2016) library for it!

```
library(ggplot2)
plot.df <- data.frame(
  x=seq(-15, 15, 0.001),
  f1=dnorm(x=seq(-15, 15, 0.001), mean=1, sd=1),
  f2=dnorm(x=seq(-15, 15, 0.001), mean=-2, sd=4),
  f3=dnorm(x=seq(-15, 15, 0.001), mean=0, sd=5),
  f4=dnorm(x=seq(-15, 15, 0.001), mean=3, sd=3)
)

ggplot(plot.df, aes(x=x))+
  geom_line(aes(y=f1, color="m=1, sd=1"))+
  geom_line(aes(y=f2, color="m=-2, sd=4"))+
  geom_line(aes(y=f3, color="m=0, sd=5"))+
  geom_line(aes(y=f4, color="m=3, sd=3"))+
  theme_bw()
```

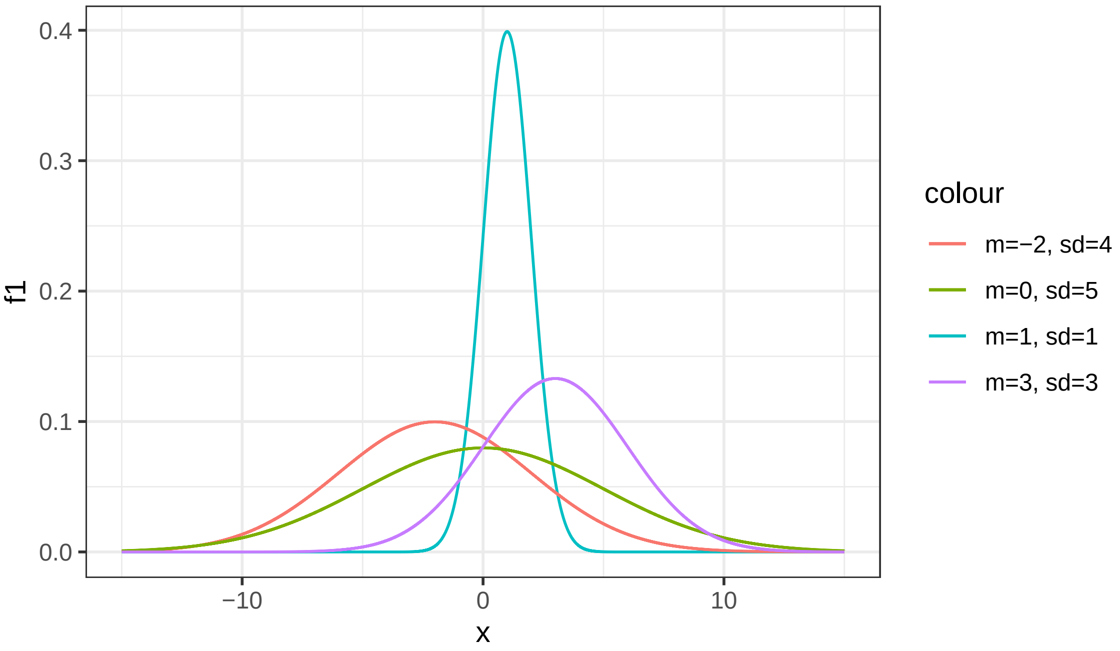


Figure 1: Gaussian PDF with various sets of parameters

As we can see on Figure 1, changing our parameters does indeed change the graph we have. Changing the μ transforms graph to the right (if the mean is positive) and to the left (if the mean is negative). Changing σ changes the kurtosis of the graph. Thus, distributions with lower value of σ show higher peaks.

(e) Graph the CDF for the same values of the parameter(s) as you did in Question 1d. What does changing the parameter(s) do to the shape of the CDF? Comment on the aspects of the CDFs that show that the CDF is valid.

Solution:

```
plot.df <- data.frame(
  x=seq(-15, 15, 0.001),
  f1=pnorm(q=seq(-15, 15, 0.001), mean=1, sd=1),
  f2=pnorm(q=seq(-15, 15, 0.001), mean=-2, sd=4),
  f3=pnorm(q=seq(-15, 15, 0.001), mean=0, sd=5),
  f4=pnorm(q=seq(-15, 15, 0.001), mean=3, sd=3)
)

ggplot(plot.df, aes(x=x))+
  geom_line(aes(y=f1, color="m=1, sd=1"))+
  geom_line(aes(y=f2, color="m=-2, sd=4"))+
  geom_line(aes(y=f3, color="m=0, sd=5"))+
  geom_line(aes(y=f4, color="m=3, sd=3"))+
  theme_bw()
```

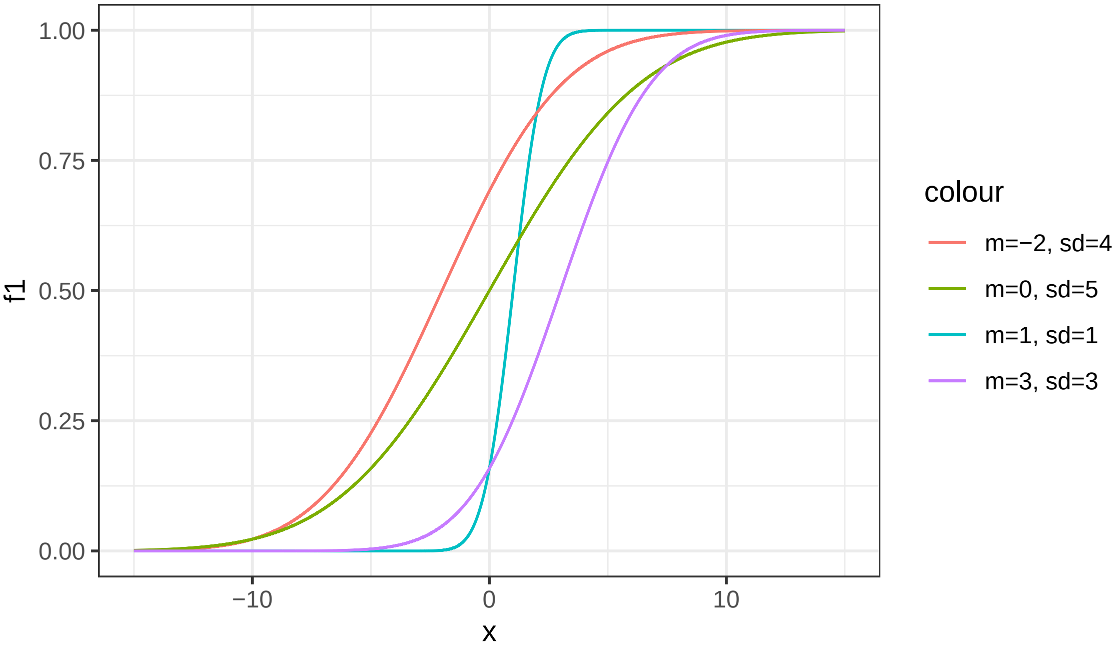


Figure 2: Gaussian CDF with various sets of parameters

We see that Figure 2 is a graph of CDF function with different sets of parameters. This function shows the probability that a probability that a random variable X will take a value less than or equal to x. We can see that our PDF is valid, for when we went over all values of X, the probability is 1.00 or 100%. Changing the value of μ changes the position of a graph (moves it to the right or to the left), while the change in σ increases the slope of the function: the higher the value of σ, the faster the function will go over all the values and reach 100%.

(f) Generate a random sample of size $n = 10, 25, 100$, and 1000 for your two sets of parameter(s). In a 4×2 grid, plot a histogram of each set of data and superimpose the true density function at the specified parameter values. Interpret the results.

Solution:

```
library(patchwork)
library(tidyverse)

sample.df <- list(x1=rnorm(10, mean=1, sd=1),
                  x2=rnorm(25, mean=1, sd=1),
                  x3=rnorm(100, mean=1, sd=1),
                  x4=rnorm(1000, mean=1, sd=1),
                  y1=rnorm(10, mean=-2, sd=4),
                       y2=rnorm(25, mean=-2, sd=4),
                       y3=rnorm(100, mean=-2, sd=4),
                       y4=rnorm(1000, mean=-2, sd=4))

buildingPlot <- function(source, mu, sigma){
  df <- data.frame(value=source) #turning values from the list into a df
  colnames(df) <- "value" #changing the name of the column

  answer<-ggplot(df, aes(value))+
    geom_histogram(aes(y=..density..), bins=10,
                 color="black")+ #building a histogram
    geom_function(fun=dnorm, args = list(mean = mu, sd = sigma),
        color="red")+
    theme_bw() +#superimposing the function
    labs(x="Value", y="Density")
  answer
}

x1<-buildingPlot(sample.df[1], 1, 1)+labs(title="Sample=10",
                                          subtitle = "mean=1, sd=1")
x2<-buildingPlot(sample.df[2], 1, 1)+labs(title="Sample=25",
                                          subtitle = "mean=1, sd=1")
x3<-buildingPlot(sample.df[3], 1, 1)+labs(title="Sample=100",
                                          subtitle = "mean=1, sd=1")
x4<-buildingPlot(sample.df[4], 1, 1)+labs(title="Sample=1000",
                                          subtitle = "mean=1, sd=1")

y1<-buildingPlot(sample.df[5], -2, 4)+labs(title="Sample=10",
                                          subtitle = "mean=-2, sd=4")
y2<-buildingPlot(sample.df[6], -2, 4)+labs(title="Sample=25",
                                          subtitle = "mean=-2, sd=4")
y3<-buildingPlot(sample.df[7], -2, 4)+labs(title="Sample=100",
                                          subtitle = "mean=-2, sd=4")
y4<-buildingPlot(sample.df[8], -2, 4)+labs(title="Sample=1000",
                                          subtitle = "mean=-2, sd=4")

(x1+y1)/(x2+y2)/(x3+y3)/(x4+y4)
```

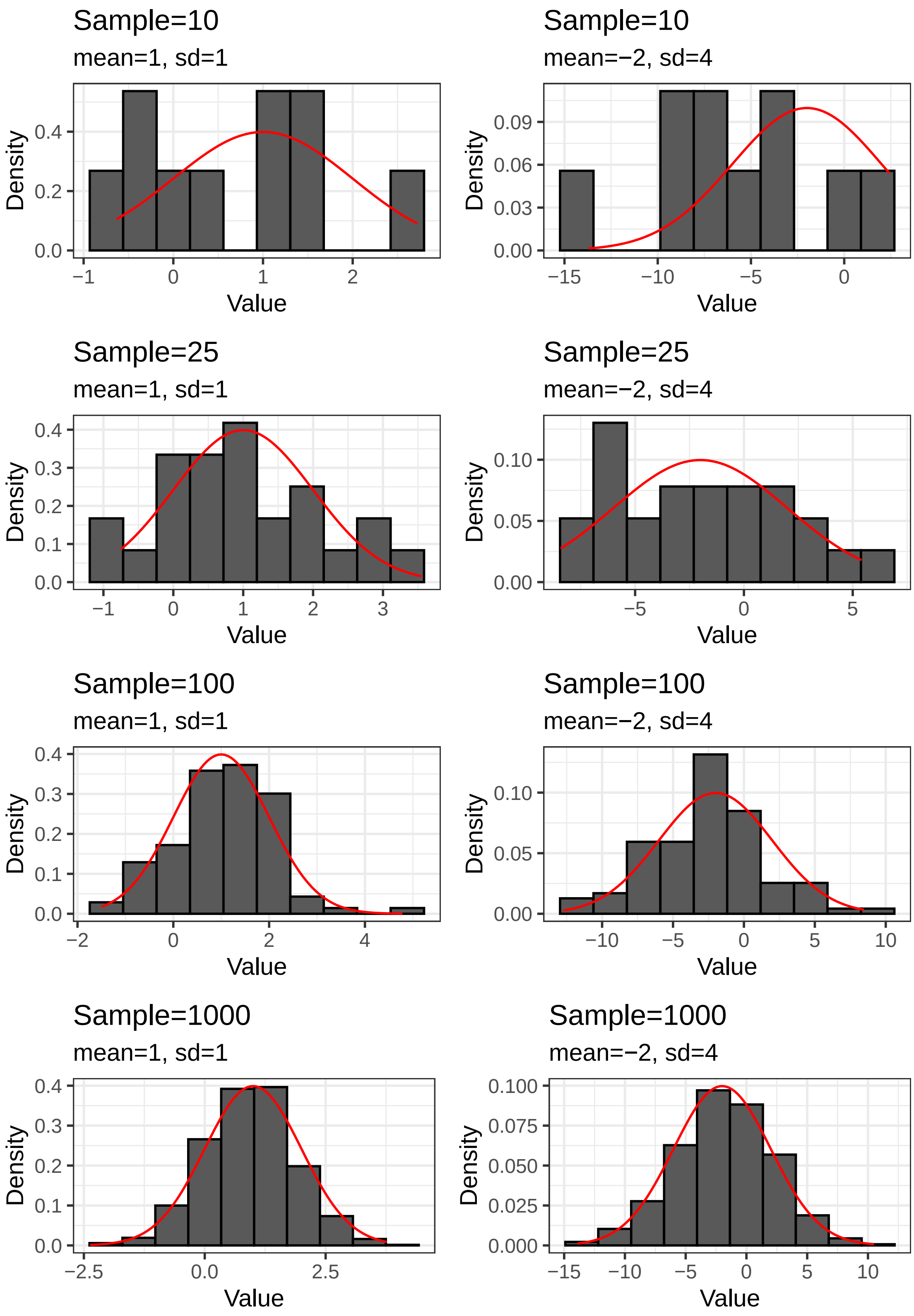


Figure 3: Histograms of each set of data with superimposed true density function

As we can see, the bigger sample we have, the closer the resulting histogram is to the real density function that we built on our set of parameters. We can notice that it's particularly true for $n > 30$.

i. Our PDF is such a good fit for the data because we are using `rnorm()` function that randomly generates numbers for the normal distribution.

ii. A good fit can also be explained via Central Limit Theorem: as our sample enlarges, the distribution of our random variable follows a normal distribution.

Checker Response (Ishraque): After our group meeting on Thursday I analyzed Chris' work and gave a few suggestions. I suggested some edits to the formatting of equations and descriptions in question 1. Next, I suggested some edits to the aesthetics of the plots of PDF and CDF and the histograms for the sections d, e,and f. These proposals were accepted promptly since they would improve the overall readability of the code and the solutions. I also suggested that for problem 1(f), he switches his method of plotting to a function based method instead. In general the code and the initial responses were already very well thought our and formatted and further edits only helped make it more concsise and clear.

2. Continue with the continuous distribution you selected for Question 1.

(a) Provide the mean, standard deviation, skewness, and kurtosis of the PDF. Ensure to interpret each.

$$
\begin{aligned}
E(x) &= \mu && \textbf{[Mean]} \\
var(X) &= \sigma^2 && \textbf{[Variance]} \\
skew(X) &= 0 && \textbf{[Skewness]} \\
kurt(X) &= 0 && \textbf{[Kurtosis]}
\end{aligned}
$$

The population skewness of the PDF is 0, which indicates that the Normal distribution is symmetric (with line of symmetry at the mean). The population excess kurtosis of the PDF is 0, which indicates that the Normal distribution is mesokurtic.

(b) Generate a random sample of size $n = 10, 25, 100,$ and 1000 for your two sets of parameter(s). Calculate the sample mean, standard deviation, skewness, and kurtosis. Interpret the results.

```r
library(e1071)
library(tidyverse)
library(patchwork)
obs <- c(10, 25, 100, 1000)
```

```r
s1 <- data.frame(x = rnorm(n = obs[1], mean = 0, sd = 1))
s1_stats <- s1 %>%
summarize(Mean = mean(x), SD = sd(x),
        Skewness = skewness(x),
        "Excess Kurtosis" = kurtosis(x))
s1_stats

##         Mean        SD  Skewness Excess Kurtosis
## 1 -0.0673861 0.6562915 -1.200612       0.1396012
```

Interpretation:
Center: The sample mean of a normal distribution is the balancing point of the distribution. In this case, this will be equal to around -

```r
s1_stats$Mean

## [1] -0.0673861
```

Spread: The average distance from the sample mean is -

```r
s1_stats$SD

## [1] 0.6562915
```

Skewness: The skewness of a dataset describes the symmetry of a distribution.

```r
s1_stats$Skewness

## [1] -1.200612
```

Since skewness ¿ 0, there are more observations for low values and fewer with high values.
Kurtosis describes how the peaked data is in relation to the Gaussian distribution (the bell-shaped curve).

```
s1_stats$`Excess Kurtosis`

## [1] 0.1396012
```

Since the excess kurtosis ¡ 0, it indicates that the data is platykurtic.

```
s2 <- data.frame(x = rnorm(n = obs[2], mean = 0, sd = 1))
s2_stats <- s2 %>% summarize(Mean = mean(x),
                             SD = sd(x), Skewness = skewness(x),
                             "Excess Kurtosis" = kurtosis(x))
s2_stats

##          Mean       SD   Skewness Excess Kurtosis
## 1 -0.05454249 1.140622 -0.3255927      0.03932679
```

Interpretation:
Center: The sample mean of a normal distribution is the balancing point of the distribution. In this case, this will be equal to around -

```
s2_stats$Mean

## [1] -0.05454249
```

Spread: The average distance from the sample mean is -

```
s2_stats$SD

## [1] 1.140622
```

Skewness: The skewness of a dataset describes the symmetry of a distribution.

```
s2_stats$Skewness

## [1] -0.3255927
```

Since skewness is slightly greater than 0, there are more observations for low values and fewer with high values. However, note that the skewness for this sample is closer to 0 than s1.
Kurtosis describes how the peaked data is in relation to the Gaussian distribution (the bell-shaped curve).

```
s2_stats$`Excess Kurtosis`

## [1] 0.03932679
```

Since the excess kurtosis ¡ 0, it indicates that the data is platykurtic. However, note that the excess kurtosis for this sample is closer to 0 than s1.

```
s3 <- data.frame(x = rnorm(n = obs[3], mean = 0, sd = 1))
s3_stats <- s3 %>% summarize(Mean = mean(x), SD = sd(x),
                             Skewness = skewness(x),
                             "Excess Kurtosis" = kurtosis(x))
s3_stats

##         Mean       SD   Skewness Excess Kurtosis
## 1 0.09983758 1.046754 0.04521973      -0.6180954
```

Interpretation:
Center: The sample mean of a normal distribution is the balancing point of the distribution. In this case, this will be equal to around -

```
s3_stats$Mean

## [1] 0.09983758
```

Spread: The average distance from the sample mean is -

```
s3_stats$SD

## [1] 1.046754
```

Skewness: The skewness of a dataset describes the symmetry of a distribution.

```
s3_stats$Skewness

## [1] 0.04521973
```

Since skewness is slightly greater than 0, there are more observations for low values and fewer with high values. However, note that the skewness for this sample is closer to 0 than s2.
Kurtosis describes how the peaked data is in relation to the Gaussian distribution (the bell-shaped curve).

```
s3_stats$‘Excess Kurtosis‘

## [1] -0.6180954
```

Since the excess kurtosis ¡ 0, it indicates that the data is platykurtic. However, note that the excess kurtosis for this sample is closer to 0 than s2.

```
s4 <- data.frame(x = rnorm(n = obs[4], mean = 0, sd = 1))
s4_stats <- s4 %>% summarize(Mean = mean(x),
                             SD = sd(x), Skewness = skewness(x),
                             "Excess Kurtosis" = kurtosis(x))
```

Interpretation:
Center: The sample mean of a normal distribution is the balancing point of the distribution. In this case, this will be equal to around -

```
s4_stats$Mean

## [1] -0.05839676
```

Spread: The average distance from the sample mean is -

```
s4_stats$SD

## [1] 0.9959243
```

Skewness: The skewness of a dataset describes the symmetry of a distribution.

```
s4_stats$Skewness

## [1] 0.09558234
```

Since skewness is almost 0 (slightly greater), the distribution is almost symmetric.
Kurtosis describes how the peaked data is in relation to the Gaussian distribution (the bell-shaped curve).

```
s4_stats$‘Excess Kurtosis‘

## [1] 0.09031246
```

Since the excess kurtosis is almost 0 (slightly lesser), it indicates that the data is almost mesokurtic.
As the sample size increases, the distribution moves towards becoming more Normal - the mean tends to equal 0, the SD tends to equal 1, and both the skewness and excess kurtosis tend to equal 0.

(c) Generate a random sample of size $n = 10$ for your two sets of parameter(s).

```
dat1 <- data.frame(x = rnorm(n = obs[1], mean = 0, sd = 1))
```

Calculate the method of moments estimator(s) and maximum likelihood estimator(s).

```
library(nleqslv)
#########################################################
# MOM estimator
```

```r
#######################################################
norm.mom<-function(par, data){
  mu <- par[1]
  sigma <- par[2]

  EX1 <- mu               # Expected value of a normal distribution
  EX2 <- sigma            # Variance of a normal distribution

  xbar1 <- mean(data)
  xbar2 <- mean(data^2)

  c(EX1-xbar1, EX2-xbar2)
}

# Entering the starting guess, the function(s) we want to solve for c(0, 0),
# and the dataframe a arguments of the non-linear equation solver
mom1 <- nleqslv(x = c(0,1),fn = norm.mom, data = dat1$x)

#######################################################
# MLE
#######################################################
norm.ll<-function(par, data, neg=T){
  mu <- par[1]
  sigma <- par[2]
  ll <- sum(dnorm(x=data, mean=mu, sd=sigma, log = T))
  ifelse(neg, -ll, ll)
  # Since the optim() function minimizes, we use neg because on multiplying by negative,     it will
}

mle1 <- optim(par = c(0,1), fn = norm.ll, data=dat1$x)
```

In a 1 × 2 grid, plot a histogram of each set of data with (1) the method of moments estimated distribution, (2) the maximum likelihood estimated distribution, and superimpose the true distribution in both.

```r
mom_dat1 <- data.frame(x = rnorm(n = obs[1], mean = mom1$x[1], sd = mom1$x[2]))
mom1_p <- ggplot(data=mom_dat1, aes(x=x))+
  geom_histogram(color="black", fill = "blue",
                 alpha = 0.5, aes(y=..density..))+
  geom_hline(yintercept=0)+ geom_density(alpha=.2, fill="purple") +
  theme_bw()+ xlab("x")+ ylab("Density")+
  ggtitle("Method Of Moments (n = 10)") +
  geom_histogram(aes(x = dat1$x), color="black",
                 fill = "green", alpha = 0.2) +
  geom_vline(xintercept = mean(dat1$x), alpha=0.35, color="red",size=1)

mle_dat1 <- data.frame(x = rnorm(n = obs[1],
                                 mean = mle1$par[1], sd = mle1$par[2]))
mle1_p <- ggplot(data=mle_dat1, aes(x=x))+
  geom_histogram(color="black", fill = "blue",
                 alpha = 0.5, aes(y=..density..))+
  geom_hline(yintercept=0)+
  geom_density(alpha=.2, fill="purple") +
  theme_bw()+ xlab("x")+ ylab("Density")+
  ggtitle("Maximum Likelihood Estimator (n = 10)")  +
  geom_histogram(aes(x = dat1$x, y=..density..),
                 color="black", fill = "green", alpha = 0.2) +
  geom_vline(xintercept = mean(dat1$x), alpha=0.35, color="red",size=1)

mom1_p + mle1_p
```

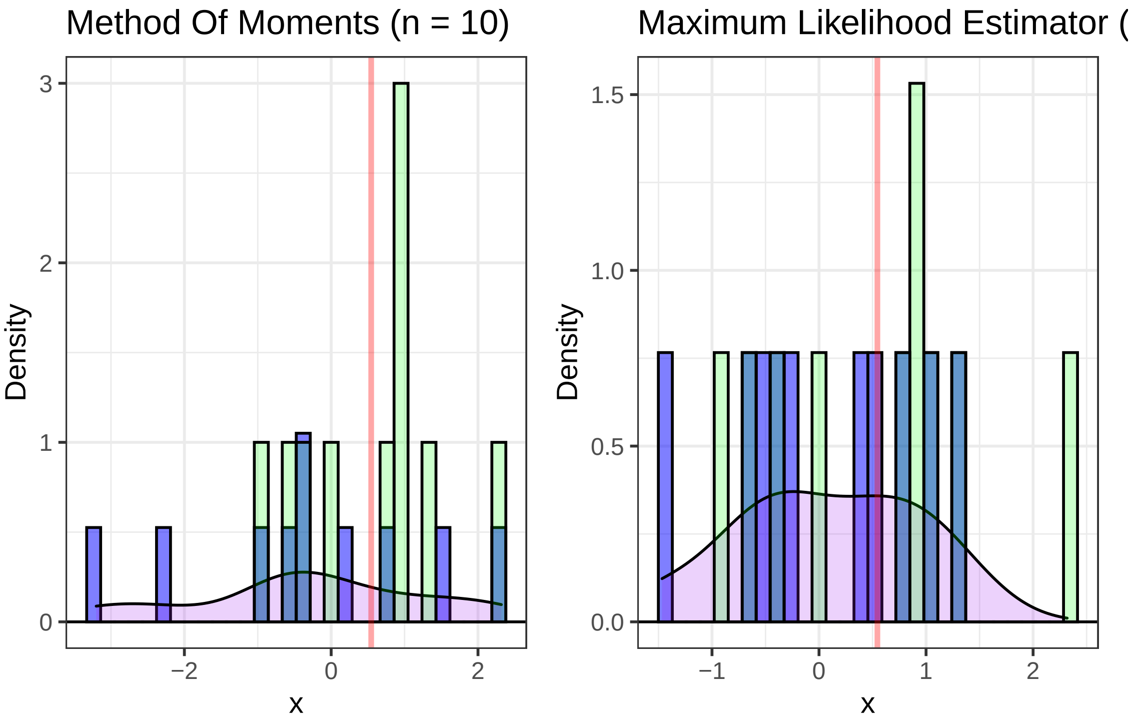


Figure 4: Cool caption!

```
dat1_h <- ggplot(data=dat1, aes(x=x))+
  geom_histogram(color="black", fill = "lightblue",
                 alpha = 0.5, binwidth = 0.2, aes(y=..density..))+
  geom_hline(yintercept=0)+
  geom_density(alpha=.2, fill="purple") +
  theme_bw()+ xlab("x")+ ylab("Density")+
  ggtitle("Normal Distrubution (n = 10)") +
  geom_vline(xintercept = mean(dat1$x), alpha=0.35,
             color="red",size=1)

dat1<-dat1 %>% mutate(n = seq(1,obs[1],1))

dat1_p <- ggplot(data=dat1, aes(x=n, y=x))+
  geom_line()+
  geom_hline(yintercept=0)+
  geom_hline(yintercept = mean(dat1$x), alpha=0.35, color="red",size=1) +
  theme_bw()+ xlab("n")+ ylab("x")+
  ggtitle("Normal Distrubution (n = 10)")

dat1_h + dat1_p
```

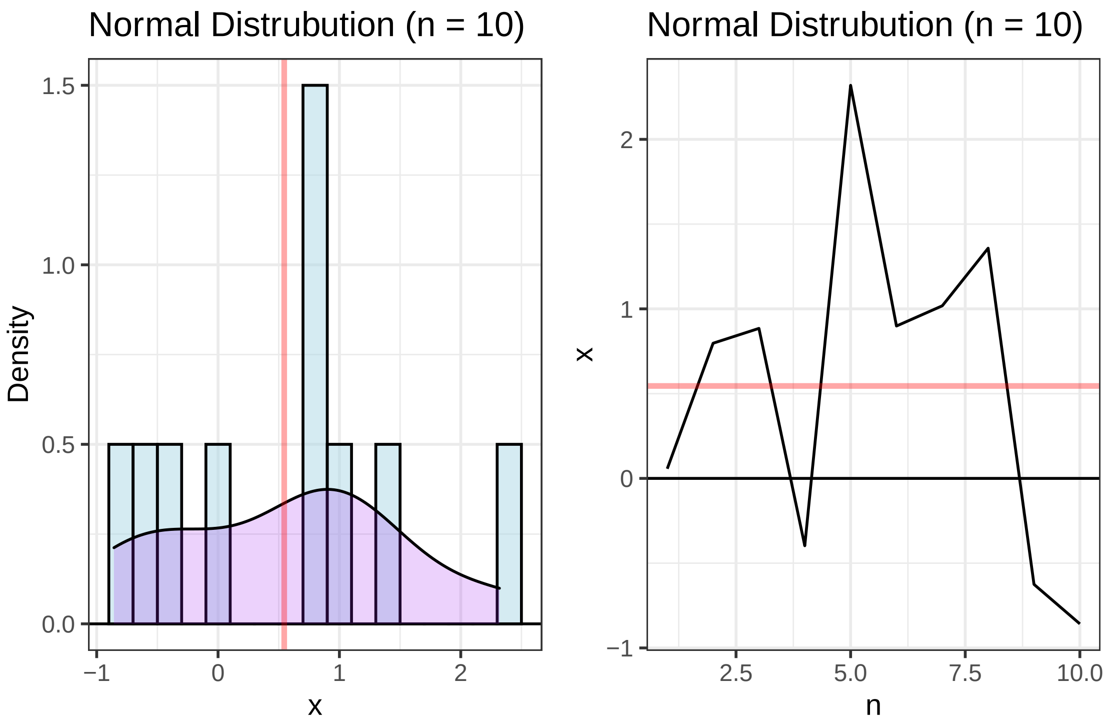


Figure 5: Cool caption!

(d) Generate a random sample of size $n = 25$ for your two sets of parameter(s).

```
dat2 <- data.frame(x = rnorm(n = obs[2], mean = 0, sd = 1))
```

Calculate the method of moments estimator(s) and maximum likelihood estimator(s).

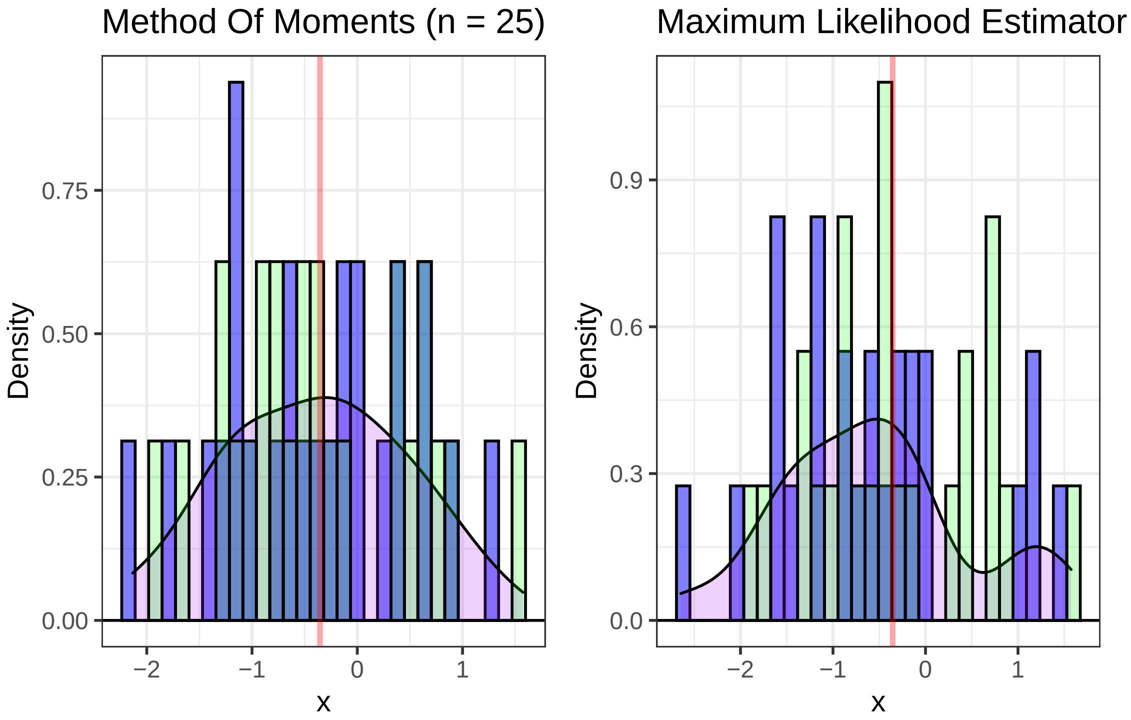
```
mom2 <- nleqslv(x = c(0,1),fn = norm.mom, data = dat2$x)
mle2 <- optim(par = c(0,1), fn = norm.ll, data=dat2$x)
```

In a 1×2 grid, plot a histogram of each set of data with (1) the method of moments estimated distribution, (2) the maximum likelihood estimated distribution, and superimpose the true distribution in both.

```
mom_dat2 <- data.frame(x = rnorm(n = obs[2],
                                 mean = mom2$x[1], sd = mom2$x[2]))
mom2_p <- ggplot(data=mom_dat2, aes(x=x))+
  geom_histogram(color="black", fill = "blue",
                 alpha = 0.5, aes(y=..density..))+
  geom_hline(yintercept=0)+ geom_density(alpha=.2, fill="purple") +
  theme_bw()+ xlab("x")+ ylab("Density")+
  ggtitle("Method Of Moments (n = 25)") +
  geom_histogram(aes(x = dat2$x, y=..density..),
                 color="black", fill = "green", alpha = 0.2) +
  geom_vline(xintercept = mean(dat2$x),
             alpha=0.35, color="red",size=1)

mle_dat2 <- data.frame(x = rnorm(n = obs[2],
                                 mean = mle2$par[1], sd = mle2$par[2]))
mle2_p <- ggplot(data=mle_dat2, aes(x=x))+
  geom_histogram(color="black", fill = "blue",
                 alpha = 0.5, aes(y=..density..))+
  geom_hline(yintercept=0)+
  geom_density(alpha=.2, fill="purple") +
  theme_bw()+ xlab("x")+ ylab("Density")+
  ggtitle("Maximum Likelihood Estimator (n = 25)") +
  geom_histogram(aes(x = dat2$x, y=..density..),
                 color="black", fill = "green", alpha = 0.2) +
  geom_vline(xintercept = mean(dat2$x), alpha=0.35, color="red",size=1)

mom2_p + mle2_p
```


Method Of Moments (n = 25)

Maximum Likelihood Estimator

Figure 6: Cool caption!

```
dat2_h <- ggplot(data=dat2, aes(x=x))+
  geom_histogram(color="black", fill = "lightblue",
                 alpha = 0.5, binwidth = 0.2,
                 aes(y=..density..))+
  geom_hline(yintercept=0)+
  geom_density(alpha=.2, fill="purple") +
  theme_bw()+
  xlab("x")+
  ylab("Density")+
  ggtitle("Normal Distrubution (n = 25)") +
  geom_vline(xintercept = mean(dat2$x), alpha=0.35, color="red",size=1)

dat2 <-dat2 %>% mutate(n = seq(1,obs[2],1))
dat2_p <- ggplot(data=dat2, aes(x=n, y=x))+
  geom_line()+
  geom_hline(yintercept=0)+
  geom_hline(yintercept = mean(dat2$x), alpha=0.35, color="red",size=1) +
  theme_bw()+
  xlab("n")+
  ylab("x")+
  ggtitle("Normal Distrubution (n = 25)")

dat2_h + dat2_p
```

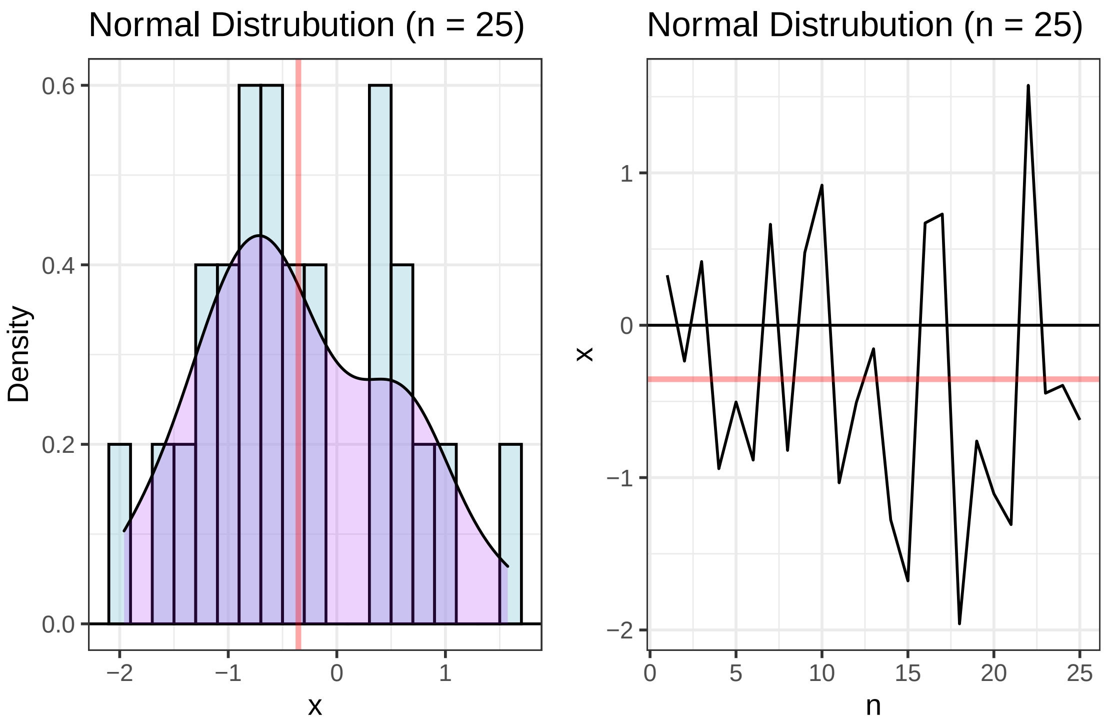


Figure 7: Cool caption!

(e) Generate a random sample of size $n = 100$ for your two sets of parameter(s).

```
dat3 <- data.frame(x = rnorm(n = obs[3], mean = 0, sd = 1))
```

Calculate the method of moments estimator(s) and maximum likelihood estimator(s).

```
mom3 <- nleqslv(x = c(0,1),fn = norm.mom, data = dat3$x)
mle3 <- optim(par = c(0,1), fn = norm.ll, data=dat3$x)
```

In a 1×2 grid, plot a histogram of each set of data with (1) the method of moments estimated distribution, (2) the maximum likelihood estimated distribution, and superimpose the true distribution in both.

```
mom_dat3 <- data.frame(x = rnorm(n = obs[3], mean = mom3$x[1], sd = mom3$x[2]))
mom3_p <- ggplot(data=mom_dat3, aes(x=x))+
  geom_histogram(color="black",
                 fill = "blue", alpha = 0.5, aes(y=..density..))+
  geom_hline(yintercept=0)+
  geom_density(alpha=.2, fill="purple") +
  theme_bw()+ xlab("x")+
  ylab("Density")+
  ggtitle("Method of Moments (n = 100)") +
  geom_histogram(aes(x = dat3$x, y=..density..),
                 color="black", fill = "green", alpha = 0.2) +
  geom_vline(xintercept = mean(dat3$x), alpha=0.35, color="red",size=1)

mle_dat3 <- data.frame(x = rnorm(n = obs[3],
                                 mean = mle3$par[1], sd = mle3$par[2]))
mle3_p <- ggplot(data=mle_dat3, aes(x=x))+
  geom_histogram(color="black", fill = "blue",
                 alpha = 0.5, aes(y=..density..))+
  geom_hline(yintercept=0)+
  geom_density(alpha=.2, fill="purple") +
  theme_bw()+ xlab("x")+ ylab("Density")+
  ggtitle("Maximum Likelihood Estimator (n = 100)") +
  geom_histogram(aes(x = dat3$x, y=..density..),
                 color="black", fill = "green", alpha = 0.2) +
  geom_vline(xintercept = mean(dat3$x), alpha=0.35, color="red",size=1)

mom3_p + mle3_p
```

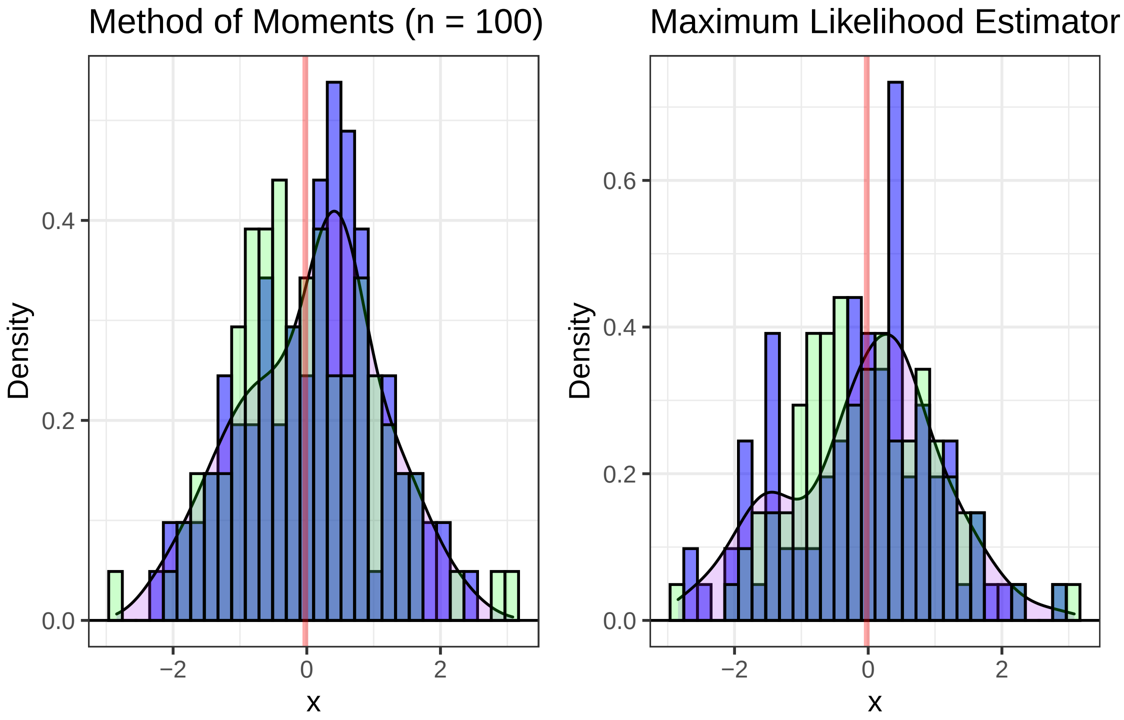


Figure 8: Cool caption!

```
dat3_h <- ggplot(data=dat3, aes(x=x))+
  geom_histogram(color="black", fill = "lightblue",
                 alpha = 0.5, binwidth = 0.2, aes(y=..density..))+
  geom_hline(yintercept=0)+
  geom_density(alpha=.2, fill="purple") +
  theme_bw()+ xlab("x")+ ylab("Density")+
  ggtitle("Normal Distrubution (n = 100)")

dat3 <-dat3 %>% mutate(n = seq(1,obs[3],1))
dat3_p <- ggplot(data=dat3, aes(x=n, y=x))+
  geom_line()+
  geom_hline(yintercept=0)+
  geom_hline(yintercept = mean(dat3$x), alpha=0.35, color="red",size=1) +
  theme_bw()+ xlab("n")+ ylab("x")+
  ggtitle("Normal Distrubution (n = 100)")

dat3_h + dat3_p
```

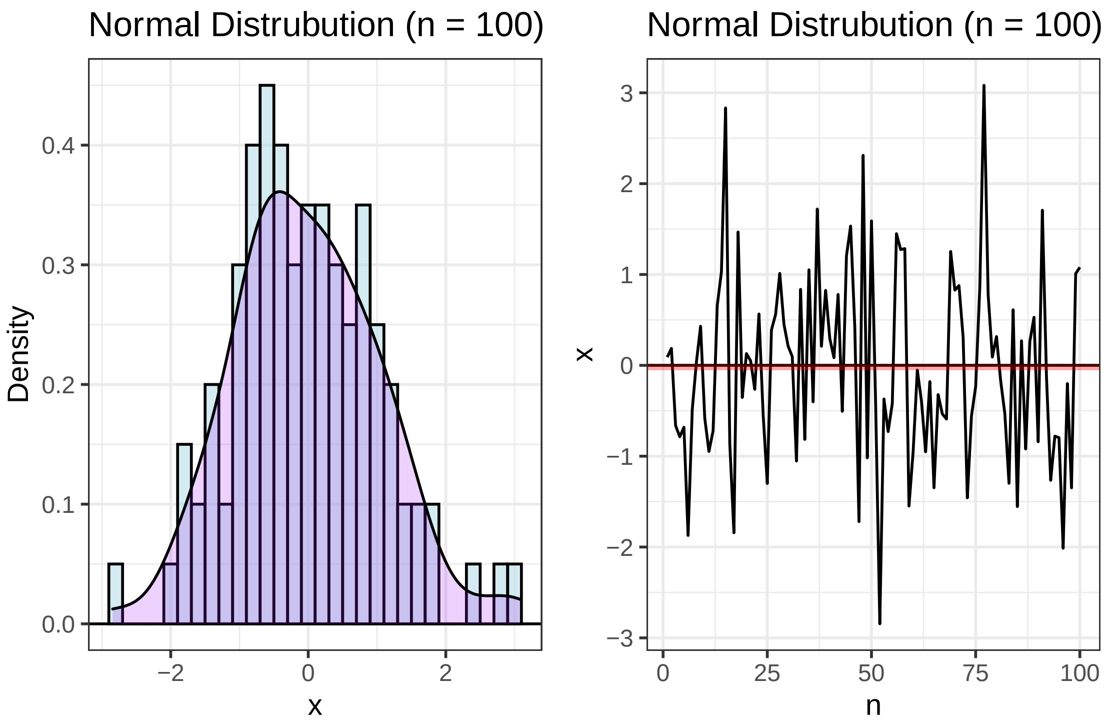


Figure 9: Cool caption!

(f) Generate a random sample of size $n = 100$ for your two sets of parameter(s).

```r
dat4 <- data.frame(x = rnorm(n = obs[4], mean = 0, sd = 1))
```

Calculate the method of moments estimator(s) and maximum likelihood estimator(s).

```r
mom4 <- nleqslv(x = c(0,1),fn = norm.mom, data = dat4$x)
mle4 <- optim(par = c(0,1), fn = norm.ll, data=dat4$x)
```

In a 1×2 grid, plot a histogram of each set of data with (1) the method of moments estimated distribution, (2) the maximum likelihood estimated distribution, and superimpose the true distribution in both.

```r
mom_dat4 <- data.frame(x = rnorm(n = obs[4], mean = mom4$x[1], sd = mom4$x[2]))
mom4_p <- ggplot(data=mom_dat4, aes(x=x))+
  geom_histogram(color="black", fill = "blue",
                 alpha = 0.5, aes(y=..density..))+
  geom_hline(yintercept=0)+
  geom_density(alpha=.2, fill="purple") +
  theme_bw()+ xlab("x")+ ylab("Density")+
  ggtitle("Method of Moments (n = 1000)") +
  geom_histogram(aes(x = dat4$x, y=..density..),
                 color="black", fill = "green", alpha = 0.2) +
  geom_vline(xintercept = mean(dat4$x), alpha=0.35, color="red",size=1)

mle_dat4 <- data.frame(x = rnorm(n = obs[4],
                                 mean = mle4$par[1], sd = mle4$par[2]))
mle4_p <- ggplot(data=mle_dat4, aes(x=x))+
  geom_histogram(color="black", fill = "blue",
                 alpha = 0.5, aes(y=..density..))+
  geom_hline(yintercept=0)+ geom_density(alpha=.2, fill="purple") +
  theme_bw()+ xlab("x")+ ylab("Density")+
  ggtitle("Maximum Likelihood Estimator (n = 1000)") +
  geom_vline(xintercept = mean(dat4$x), alpha=0.35,
               color="red",size=1) +
  geom_histogram(aes(x = dat4$x, y=..density..),
                   color="black", fill = "green", alpha = 0.2)

mom4_p + mle4_p
```

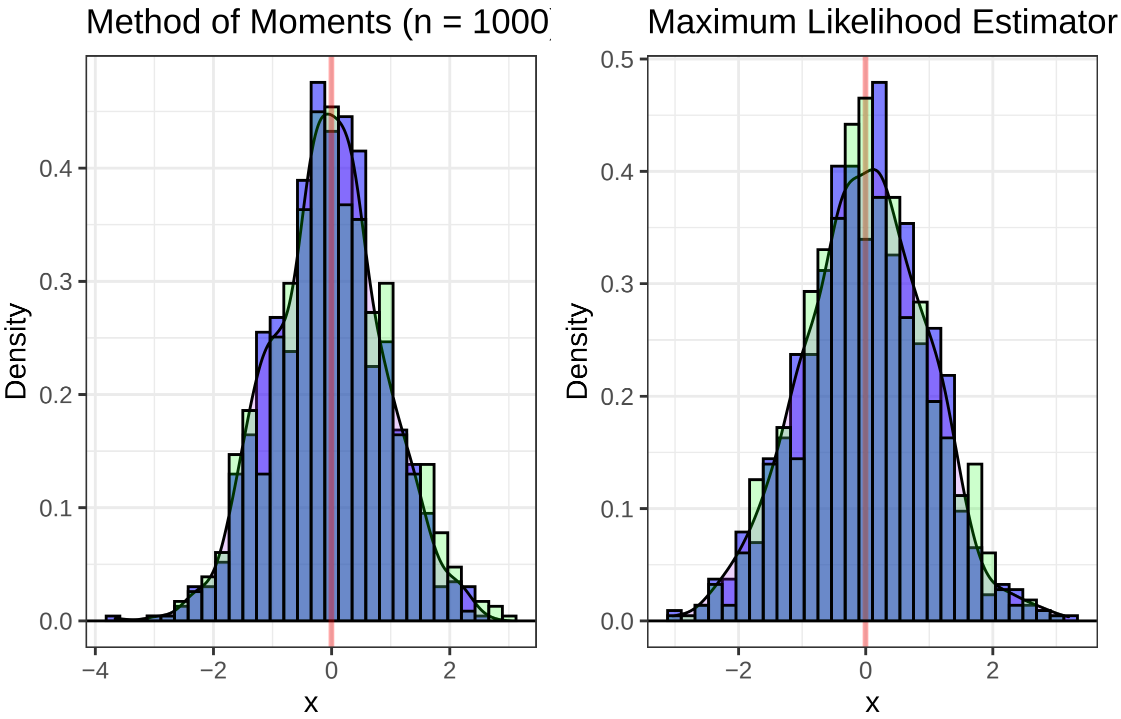


Figure 10: Cool caption!

```
dat4_h <- ggplot(data=dat4, aes(x=x))+
  geom_histogram(color="black", fill = "lightblue",
                   alpha = 0.5, aes(y=..density..))+
  geom_hline(yintercept=0)+ geom_density(alpha=.2, fill="purple") +
  theme_bw()+ xlab("x")+ ylab("Density")+
  ggtitle("Normal Distrubution (n = 1000)") +
  geom_vline(xintercept = mean(dat4$x), alpha=0.35, color="red",size=1)

dat4 <-dat4 %>% mutate(n = seq(1,obs[4],1))
dat4_p <- ggplot(data=dat4, aes(x=n, y=x))+
  geom_line()+ geom_hline(yintercept=0)+
  geom_hline(yintercept = mean(dat3$x), alpha=0.35, color="red",size=1) +
  theme_bw()+ xlab("n")+ ylab("x")+ ggtitle("Normal Distrubution (n = 1000)")

dat4_h + dat4_p
```

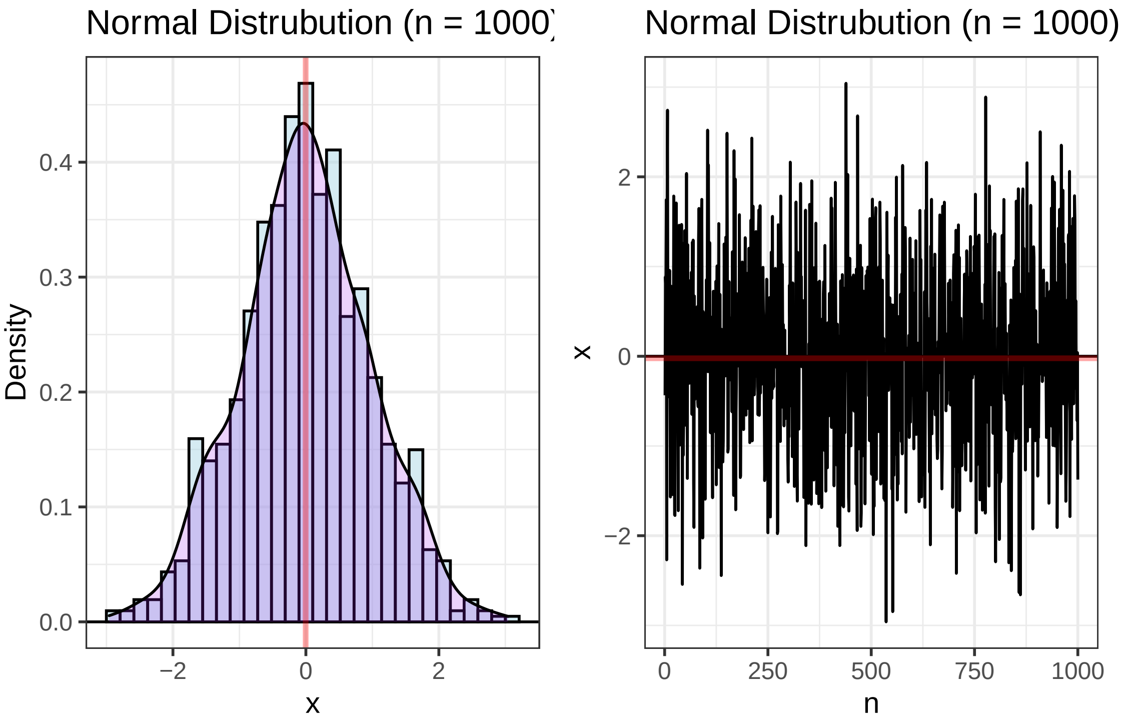


Figure 11: Cool caption!

(g) Comment on the results of parts (c)-(f).

Parts (c)-(f) illustrate the weak law of large numbers. As the sample size n increases, we see that both our computed estimators (MOM and MLE) tend to overlap more with their corresponding probability distributions - there is very less overlap when n = 10; however, at n = 1000, both the histograms are essentially superimposed on top of one another. Thus, as the sample size increases, our estimators do a better job at estimating the population statistics of the Normal distribution. We also notice that as the sample size increases, the sample mean gets closer to 0.

Checker Response (Peter): Part `(a)` of question 2 seems alright, though I would include some more information about what a distribution's mean, standard deviation, skewness, and kurtosis are exactly and what they tell us about the graph of our distribution. Part `(b)` also seems alright containing the computationally obtained mean, standard deviation, skewness, and kurtosis for both sets of parameters. It also contains some brief information about what those numbers mean about how the data is distributed but could benefit from comparing those numbers to what the calculated mean, standard deviation, skewness, and kurtosis are. Parts `(c)-(f)` also seem alright and the graphs look like what I would expect to see from a randomly generated data set that follows a normal distribution. However, some of the figures are unlabelled, and it is difficult to tell which graph corresponds to which set of parameters. Finally, I think that the interpretations put forwards in part `(g)` seem good and agree that these graphs demonstrate the weak law of large numbers. Overall, while most of the work here seems correct and is mostly complete, the poor formatting of the document makes it hard to follow and is the number 1 thing that could use improving upon.

3. Select a discrete distribution (not the Poisson). It does not have to be one that we cover in the notes! To explore the PMF of your distribution, specify two sets of parameter(s) for your distribution.

   (a) **History** Discuss what types of random variables are modeled with your distribution. Be sure to include a discussion about the support and ensure to provide the mass function, and CDF. This requires some internet research – what's the history of the distribution, why was it created and named? What are some exciting applications of this distribution? Cite all of your sources.

Solution:
The discrete distribution we have chosen is the Bernoulli distribution. The Bernoulli distribution is a discrete probability distribution for a Bernoulli trial - a probabilistic experiment that can have one of two outcomes, success (x = 1) and failure (x = 0), and in which the probability fo success is p. Often p is called the Bernoulli probability parameter (Forbes et al., 2011). In Bernoulli distribution, the random variable X can have only one of two values: 0 or 1. This means that the support of our discrete random variable is the set $\{0, 1\}$. This distribution can be summarized as follows:

$$\begin{aligned}
p &\in (0,1) && \textbf{[Parameter]} \\
\mathcal{X} &= \{x : x \in \{0,1\}\} && \textbf{[Support]} \\
f_X(x \mid p) &= p^x(1-p)^{1-x} I(x \in \{0,1\}) && \textbf{[PMF]} \\
F_X(x \mid p) &= P(X \leq \lfloor x \rfloor) && \\
&= [(1-p)I(\lfloor x \rfloor = 0)] + I(\lfloor x \rfloor \geq 1) && \textbf{[CDF]}
\end{aligned}$$

Simply, the Bernoulli distribution can be thought of as a model for the set of possible outcomes of any single experiment that asks a yes-no question (Vukadin et al., 2021). This distribution is named after the 17th century Swiss mathematician Jacob Bernoulli, because he was the one who explicitly defined the concept of Bernoulli trial (in his book *Ars Conjectandi*) described above. Bernoulli distribution serves as a building block for discrete distributions which model Bernoulli trials, such as Binomial distribution and geometric distribution (Cthaeh, 2020). Logistic regression, a widely used classification model that models a binary outcome, also takes advantage of the Bernoulli distribution (Kleinbaum and Klein, 2010).
Since the Bernoulli distribution is not cataloged by R, we have to introduce it into our calculations with the following functions:

```r
# Bernoulli PMF
dbern<-function(x,prob){
  if(prob<0 | prob>1){
    errormsg <- "This function is only valid for success probabilities between 0 and 1."
    stop(errormsg)
  }
  indicator <- rep(0, length(x))
  indicator[x==0] <- 1 # indicator should be one if x=0
  indicator[x==1] <- 1 # indicator should be one if x=1
  fx <- (prob^x * (1-prob)^(1-x)) * indicator # PMF formula
  return(fx)
}
# Bernoulli CDF
pbern<-function(q, prob){
  if(prob<0 | prob>1){
    errormsg<-"This function is only valid for success probabilities between 0 and 1."
    stop(errormsg)
  }
  indicator1 <- rep(1, length(q))
  indicator1[q != 0] <- 0 #indicator should be zero if x!=0
  indicator2 <- rep(1, length(q))
  indicator2[q < 1] <- 0 #indicator should be zero if x<1
  Fx <- (1-prob) * indicator1 + indicator2
  return(Fx)
}
```

The R packages Tidyverse (Wickham et al., 2019) and Patchwork (Pedersen, 2020) will be used in all of the following plots in question 3.

(b) Show that you have a valid PMF. You can show this approximately by calculating the series in a repeat loop until probability mass evaluations are infinitesimally small.

Solution: For a PMF to be valid, it has to fulfill the following statements:

1. $0 \leq f_X(x) \leq 1$ for all $x \in \mathbb{R}$
2. $\sum_{-\infty}^{\infty} f_X = \sum_{\mathcal{X}} f_X = 1$

By definition, the Bernoulli distribution satisfies statement 1, since the support is 0,1 and the probability parameter p $\in (0, 1)$, so the term for PMF (see above) cannot have a value smaller than 0 or greater than 1.

We can also show that the statement 2 is true for the Bernoulli distribution:

$$\begin{aligned}
\sum_{\mathcal{X}} f_X = \sum_{x=0}^{1} f_X(x) &= \sum_{x=0}^{1} p^x (1-p)^{1-x} \\
&= p^0(1-p)^{1-0} + p^1(1-p)^{1-1} \\
&= (1)(1-p) + p(1) \\
&= 1 - p + p \\
&= 1
\end{aligned}$$

Therefore, we have a valid PMF for our distribution.

(c) Find the median for your two sets of parameter(s). Conduct some research to find the median based on our PMF to confirm that your numerical approach is correct.

Solution: In the Bernoulli distribution, where p is the probability parameter and q = $1 - $p, when p>q, there are more ”successes” (that is when X is 1) than failure (when X is 0), so the median must also be a success. Similar, when q>p, there are more ”failures” than ”successes”, hence resulting in the median being 0. When p = q, there is an equal probability of X being 0 or 1, thus the median is ambiguous. Therefore, the median for when p = 0.4 is zero and the median for when p = 0.6 is 1. Further inquiry (Ber, 2021) suggests that this really is the case for the median of the Bernoulli distribution:

$$Median = \begin{cases} 0 & \text{if}\, p < 1/2 \\ [0, 1] & \text{if}\, p = 1/2 \\ 1 & \text{if}\, p > 1/2 \end{cases}$$

(d) Graph the PMF for several values of the parameter(s) including the two sets you specified. What does changing the parameter(s) do to the shape of the PMF?

Solution: We can plot the PMF of the Bernoulli distribution for different parameters.Note that the initially chosen parameters were p = 0.4 and p = 0.6. Let us also look at the PMF when the parameter values are 0.5, and 0.8.

```r
library(tidyverse)
library(patchwork)
plotbernPMF <- function(prob){ # Pass in the success probability
  ggdat <- data.frame(x = (-1:2),
                      f = dbern(x = (-1:2), prob = prob),
                      F = pbern(q = (-1:2), prob = prob))
  ## Plot PMF
  PMF <- ggplot(data = ggdat, aes(x = x)) +
    geom_linerange(aes(ymax = f), ymin = 0) +
    geom_hline(yintercept = 0) +
    theme_bw() +
    ylim(0, 1) +
    xlab("X") +
    ylab(bquote(f[x](x))) +
    ggtitle("Bernoulli Distribution",subtitle = paste("p =", prob))

  return(PMF)
}

plotbernPMF(0.4) + plotbernPMF(0.5) + plotbernPMF(0.6) + plotbernPMF(0.8)
```

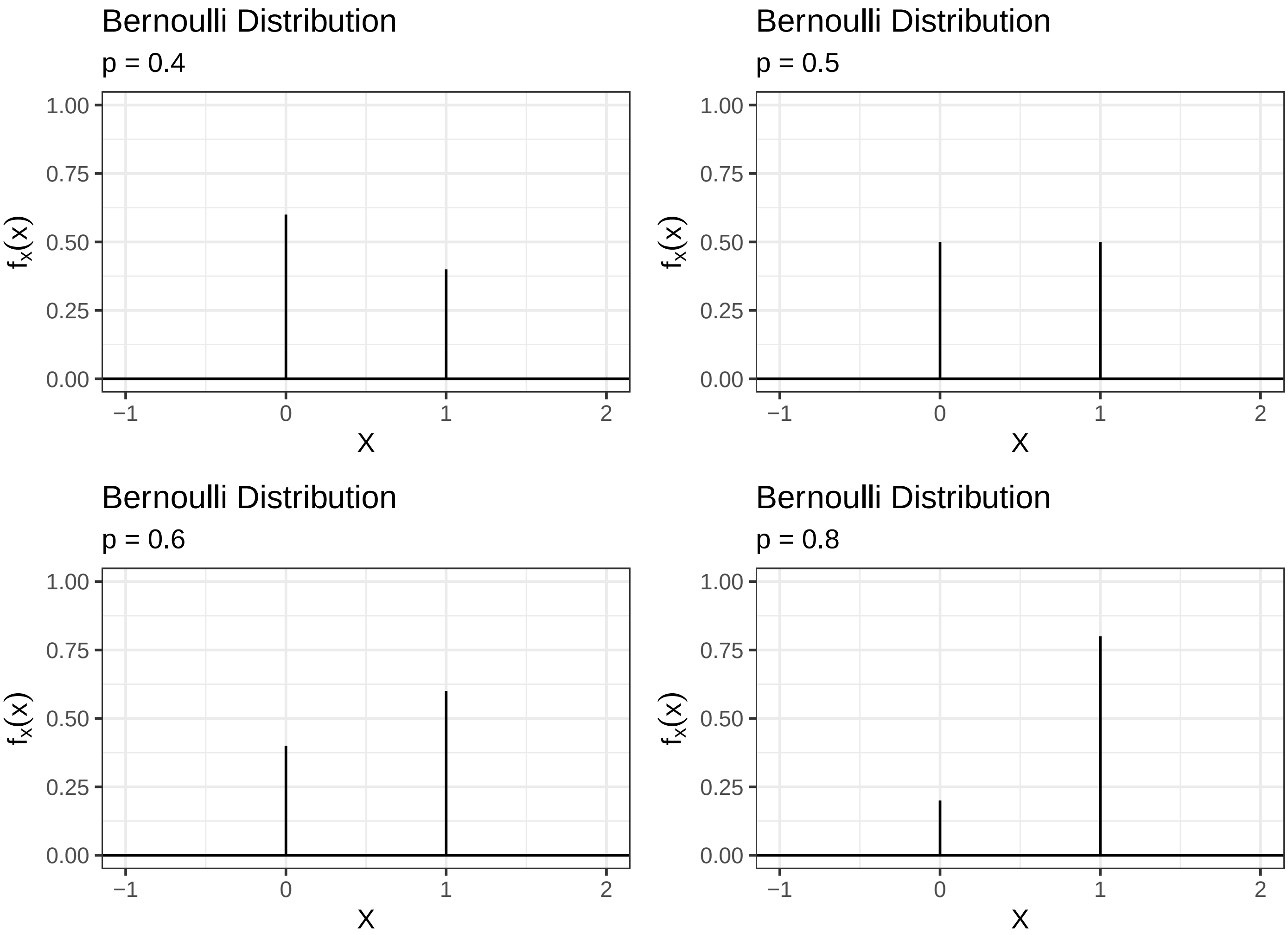


Figure 12: The CDF of the Bernoulli distribution for different probability parameters p

We observe the same characteristic of the Bernoulli distribution in Figure 12 as we discussed about its median above. If the $p < 0.5$ then there are more failures than successes, indicated by the taller PMF at 0. On the other hand, if If the $p > 0.5$ then there are more successes than failures, which is reflected in the PMF plot being taller at 1. The plot of the PMF when $p = 0.5$ suggests that there is an equal chance for success as there is for failure.

(e) Graph the CDF for the same values of the parameter(s) as you did in Question 3d. What does changing the parameter(s) do to the shape of the CDF? Comment on the aspects of the CDFs that show that the CDF is valid.
Solution: We can plot the CDF of the Bernoulli distribution for the different parameters 0.4, 0.5, 0.6, and 0.8 below:

```r
plotbernCDF <- function(prob){ # Pass in the success probability
  ggdat <- data.frame(x = (-1:2),
                      f = dbern(x = (-1:2), prob = prob),
                      F = pbern(q = (-1:2), prob = prob))
  ggdat.openpoints <- data.frame(x = ggdat$x,
                                 y = pbern(ggdat$x-1, prob = prob))
  ggdat.closedpoints <- data.frame(x = ggdat$x,
                                   y = pbern(ggdat$x,prob=prob))
  CDF<-ggplot(data = ggdat, aes(x = x, y = F)) +
    geom_step()+
    geom_point(data = ggdat.openpoints, aes(x = x, y = y), shape = 1) +
    geom_point(data = ggdat.closedpoints, aes(x = x, y = y)) +
    geom_hline(yintercept = 0.5, linetype="dotted", color="red")+
    theme_bw()+
    xlab("X")+
    ylab(bquote(F[x](x)))+
    ggtitle("Bernoulli CDF",subtitle=(paste("p =", prob)))
  return(CDF)
}
plotbernCDF(0.4) + plotbernCDF(0.5) + plotbernCDF(0.6) + plotbernCDF(0.8)
```

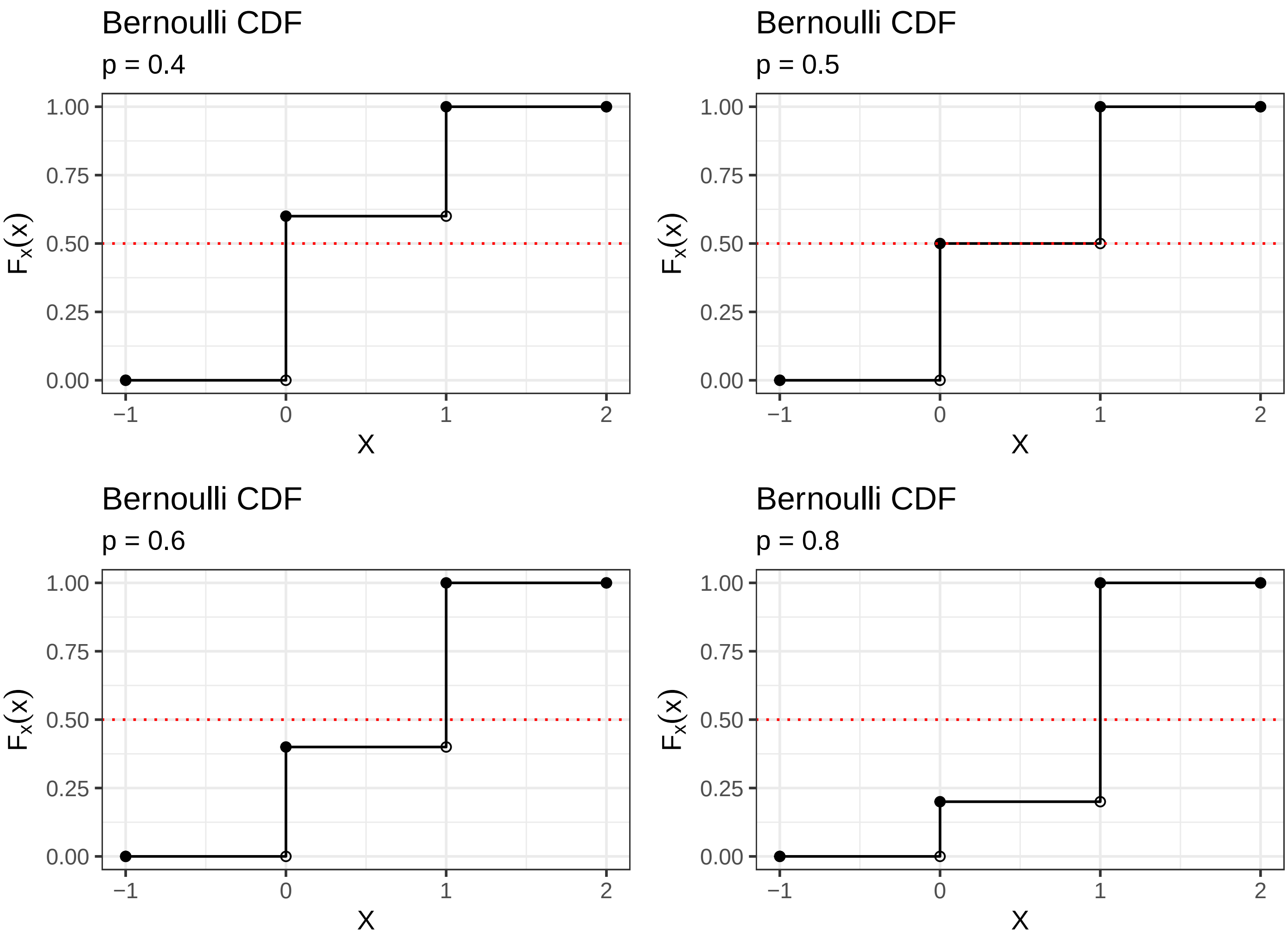


Figure 13: The CDF of the Bernoulli distribution for different probability parameters p

In the Figure 13, we can see that with increasing p, the most of the ”area” under the CDF is moving towards 1, which correctly indicates that the number of successes increase as the value of p increases. Furthermore, we can see that in all the CDF plots, the functions always add up to 1. The CDF are also monotonically increasing towards the right. All of these above characteristics together indicate that our CDF is valid. Notice that in all of the plots of Figure 13, we have drawn a horizontal line at 1/2. This is another way we can determine the median of the Bernoulli distribution using the CDF. Where the horizontal line intersects the CDF indicates where the median is for the given value of p. These results agree with our results from problem 3(c).

(f) Generate a random sample of size $n = 10, 25, 100$, and 1000 for your two sets of parameter(s). In a 4×2 grid, plot a histogram (with bin size 1) of each set of data and superimpose the true mass function at the specified parameter values. Interpret the results.

Solution: The two parameters we are working with are $\mathrm{p} = 0.4$ and $\mathrm{p} = 0.6$

```r
# Using sampling with replacement to generate arrays of 0 and 1

library(tidyverse)
library(patchwork)

plot_df <- list( x1 = bern_sample(10, 0.4),
                 x2 = bern_sample(25, 0.4),
                 x3 = bern_sample(100, 0.4),
                 x4 = bern_sample(1000, 0.4),
                 y1 = bern_sample(10, 0.6),
                 y2 = bern_sample(25, 0.6),
                 y3 = bern_sample(100, 0.6),
                 y4 = bern_sample(1000, 0.6))


buildingPlot2 <- function(source, prob, sample_size){
  df <- data.frame(value=source) #turning values from the list into a df
  colnames(df) <- "value" #changing the name of the column
  df_PMF <- data.frame(x = (-1:2),
                      f = dbern(x = (-1:2), prob = prob))
  answer<-ggplot(df, aes(value))+
     geom_histogram(data = df, aes(y=..density..), binwidth=1,
                 color="black")+
  geom_linerange(data=df_PMF, aes(x=x, ymax = f), ymin = 0, size=2, color="red")+
  theme_bw() +
  ggtitle("Bernoulli Distribution",subtitle = paste("Sample Size =", sample size, ", Prob =", prob))
  answer
}

x1 <- buildingPlot2(plot_df[1], 0.4, 10)
x2 <- buildingPlot2(plot_df[2], 0.4, 25)
x3 <- buildingPlot2(plot_df[3], 0.4, 100)
x4 <- buildingPlot2(plot_df[4], 0.4, 1000)

y1 <- buildingPlot2(plot_df[5], 0.6, 10)
y2 <- buildingPlot2(plot_df[6], 0.6, 25)
y3 <- buildingPlot2(plot_df[7], 0.6, 100)
y4 <- buildingPlot2(plot_df[8], 0.6, 1000)


(x1+y1)/(x2+y2)/(x3+y3)/(x4+y4)
```

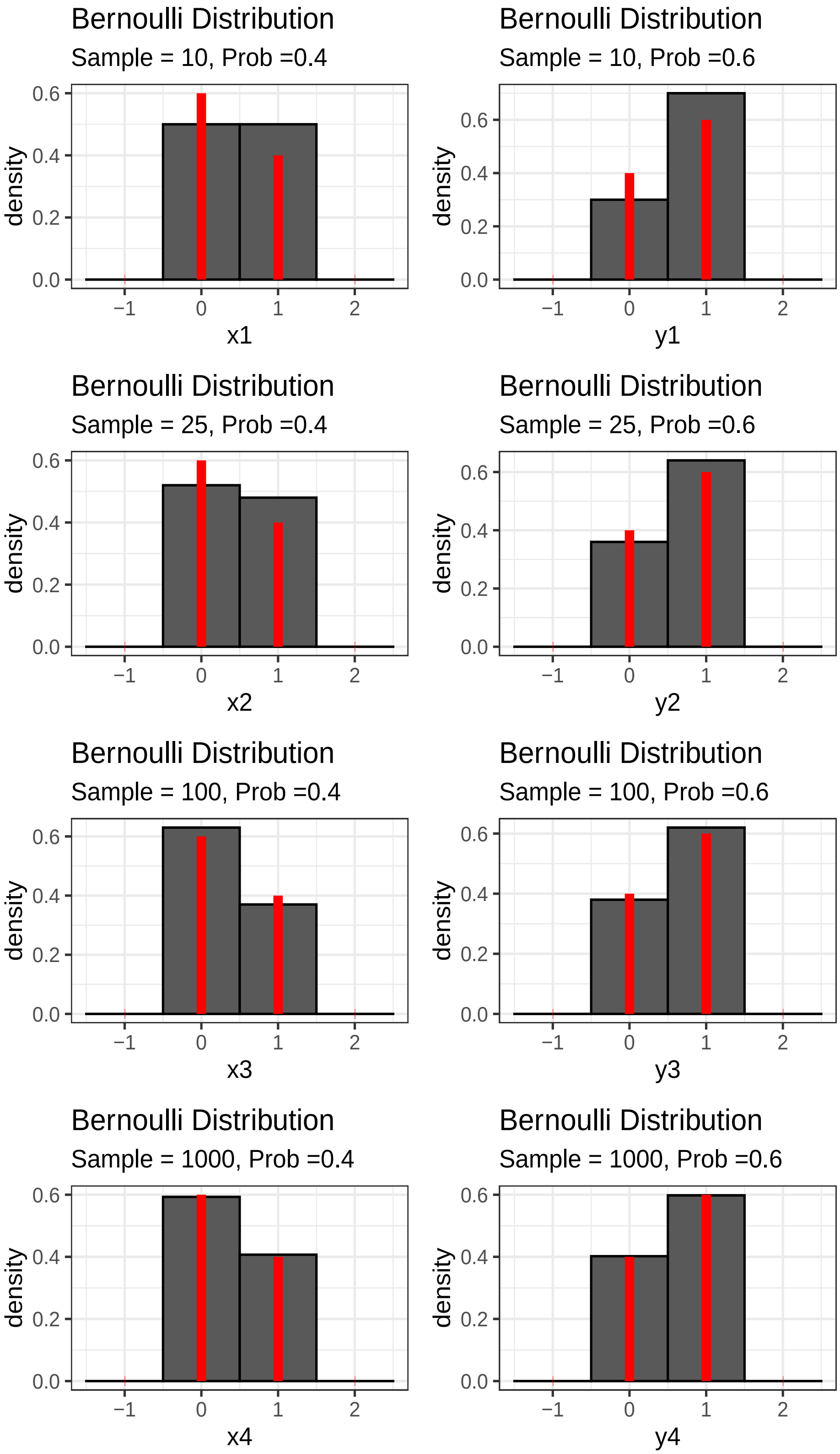


Figure 14: (Left Column) Samples of 10, 25, 100, 1000 for the Bernoulli distribution with p = 0.4. (Right Column) Samples of 10, 25, 100, 1000 for the Bernoulli distribution with p = 0.6. In each plot the red lines indicate the true PMF function of the corresponding Bernoulli distribution. Notice that the y-axis indicates the prportional frequency in the histograms

For each of the cases of p (see the first and the second row of plots in Figure 14), the more we sample, the better our histograms agree with the the true PMF of the Bernoulli distribution. These histograms above represent only one variation of the random sampling done on the Bernoulli distribution. While the histograms for n=1000 does not stray much from the PMF as we take newere samples, the smaller samples (10, 25) vary significantly from the PMF. From sample to sample, the results of the histograms vary (in terms of agreement with the true PMF). The larger the size of the sample, the more consistent is its agreement with the true PMF.

Checker Response (Chris): After having a group meeting on Thursday, I analyzed Ishraque's work and gave a couple of suggestions on the best way to improve his part of the assignment. First off, I suggested putting his separate data frames into one list instead: it would clean up his code and make it more readable. Also, instead of making new plots for each variable he's creating, I suggested writing a function that would take in variables and give him a plot as an output. This would improve the overall readability of the code. The team agreed with this decision, and we decided to implement them!

4. Continue with the discrete distribution you selected for Question

   (a) Provide the mean, standard deviation, skewness, and kurtosis of the PMF. Ensure to interpret each.

   Bernoulli Distribution:

   PMF:

$$f\left(k;p\right)=p^{k}\left(1-p\right)^{1-k} \qquad \text{for } k\in\{0,1\}$$

   Mean:

$$E(X)=p$$

   The mean represents the average or typical value that we would expect to see in a data set.

   Standard Deviation:

$$\sigma=\sqrt{p\left(1-p\right)}$$

   The standard deviation tells us how much the data in our data set varies from the mean. Data sets with lower standard deviations will have their data points grouped more tightly together, while data sets with higher standard deviations will have their data points more spread out.

   Skewness:

$$\xi_X=\frac{\left(1-p\right)-p}{\sqrt{p\left(1-p\right)}}$$

   The skewness tells us how symmetrical the data will be. If our data is positively or right-skewed, that means there will be a large grouping of data towards the left side of our distribution, with a long tail extending out to the right. Conversely, negatively or left-skewed data means there will be a large grouping of data towards the right side of our distribution, with a long tail extending out to the left.

   Kurtosis:

$$\kappa_Y=3+\frac{1-6p\left(1-p\right)}{p\left(1-p\right)}$$

   The kurtosis is used to tell us how thick or thin the peak and tails of our data distribution are. A data set with kurtosis greater than 3 will have a distribution that has thicker tails and a thinner peak. Conversely, a data set with kurtosis less than 3 will have thinner tails and a thicker peak.

   (b) Generate a random sample of size $n = 10, 25, 100,$ and 1000 for your two sets of parameter(s). Calculate the sample mean, standard deviation, skewness, and kurtosis. Interpret the results.

```
library(e1071)

even.10 <- rbinom(n=10,          #number of observations
              size=1,            #number of trials (size=1 for Bernoulli)
              prob=.5)           #probability of success

mean(even.10)                    #mean

## [1] 0.5

sd(even.10)                      #standard deviation

## [1] 0.5270463

skewness(even.10)                #skewness

## [1] 0

kurtosis(even.10)+3              #kurtosis

## [1] 0.81

even.25  <- rbinom(n=25,         #number of observations
              size=1,            #number of trials (size=1 for Bernoulli)
              prob=.5)           #probability of success

mean(even.25)                    #mean

## [1] 0.52

sd(even.25)                      #standard deviation

## [1] 0.509902

skewness(even.25)                #skewness

## [1] -0.0753086

kurtosis(even.25)+3              #kurtosis

## [1] 0.9275077

even.100 <- rbinom(n=100,        #number of observations
              size=1,            #number of trials (size=1 for Bernoulli)
              prob=.5)           #probability of success

mean(even.100)                   #mean

## [1] 0.44

sd(even.100)                     #standard deviation

## [1] 0.4988877

skewness(even.100)               #skewness

## [1] 0.2381298

kurtosis(even.100)+3             #kurtosis

## [1] 1.037379

even.1000 <- rbinom(n=1000,      #number of observations
              size=1,            #number of trials (size=1 for Bernoulli)
              prob=.5)           #probability of success

mean(even.1000)                  #mean

## [1] 0.521
```

```
sd(even.1000)                    #standard deviation

## [1] 0.4998088

skewness(even.1000)              #skewness

## [1] -0.08394811

kurtosis(even.1000)+3            #kurtosis

## [1] 1.005055

uneven.10 <- rbinom(n=10,        #number of observations
                  size=1,        #number of trials (size=1 for Bernoulli)
                  prob=.7)       #probability of success

mean(uneven.10)                  #mean

## [1] 0.9

sd(uneven.10)                    #standard deviation

## [1] 0.3162278

skewness(uneven.10)              #skewness

## [1] -2.27684

kurtosis(uneven.10)+3            #kurtosis

## [1] 6.57

uneven.25  <- rbinom(n=25,       #number of observations
                   size=1,       #number of trials (size=1 for Bernoulli)
                   prob=.7)      #probability of success

mean(uneven.25)                  #mean

## [1] 0.8

sd(uneven.25)                    #standard deviation

## [1] 0.4082483

skewness(uneven.25)              #skewness

## [1] -1.410906

kurtosis(uneven.25)+3            #kurtosis

## [1] 2.9952

uneven.100 <- rbinom(n=100,      #number of observations
                   size=1,       #number of trials (size=1 for Bernoulli)
                   prob=.7)      #probability of success

mean(uneven.100)                 #mean

## [1] 0.65

sd(uneven.100)                   #standard deviation

## [1] 0.4793725

skewness(uneven.100)             #skewness

## [1] -0.61956

kurtosis(uneven.100)+3           #kurtosis
```

```
## [1] 1.367832

uneven.1000 <- rbinom(n=1000,      #number of observations
                      size=1,      #number of trials (size=1 for Bernoulli)
                      prob=.7)     #probability of success

mean(uneven.1000)                  #mean

## [1] 0.67

sd(uneven.1000)                    #standard deviation

## [1] 0.470448

skewness(uneven.1000)              #skewness

## [1] -0.7219927

kurtosis(uneven.1000)+3            #kurtosis

## [1] 1.519796
```

Calculated Mean, Standard Deviation, Skewness and Kurtosis:

For $p = .5$:

Mean:

$$E(X) = .5$$

Standard Deviation:

$$\sigma = \sqrt{.5\,(1 - .5)}$$

$$= .5$$

Skewness:

$$\xi_X = \frac{(1 - .5) - .5}{\sqrt{.5\,(1 - .5)}}$$

$$= 0$$

Kurtosis:

$$\kappa_Y = 3 + \frac{1 - 6\,(.5)\,(1 - .5)}{.5\,(1 - .5)}$$

$$= 1$$

For $p = .7$:

Mean:

$$E(X) = .7$$

Standard Deviation:

$$\sigma = \sqrt{.7\,(1 - .7)}$$

$$\approx .45825$$

Skewness:

$$\xi_X = \frac{(1 - .7) - .7}{\sqrt{.7\,(1 - .7)}}$$

$$\approx -.87287$$

Kurtosis:

$$\kappa_Y = 3 + \frac{1 - 6\,(.7)\,(1 - .7)}{.7\,(1 - .7)}$$

$$\approx 1.76190$$

By comparing the values we received computationally to the values that we calculated we can see that as our random sample size n increases, the computational values that we received for our mean, standard deviation, skewness, and kurtosis tended to approach the actual values for the distribution that we calculated.

(c) Generate a random sample of size $n = 10$ for your two sets of parameter(s). Calculate the method of moments estimator(s) and maximum likelihood estimator(s). In a 1×2 grid, plot a histogram (with bin size 1) of each set of data with (1) the method of moments estimated distribution, (2) the maximum likelihood estimated distribution, and superimpose the true distribution in both.

```r
library(tidyverse)
library(patchwork)

second.moment <- function(data, x.bar){  #function to find the second moment
  temp = 0
  for(i in data){
    temp = temp + (i - x.bar)^2
  }
  return (temp/length(data))
}

MOM.bernoulli <- function(data){           #function to find the MOM Estimator for p in a Bernoulli di
  mu.hat <- mean(data)
  var.hat <- second.moment(data,mu.hat)
  p.hat <- (mu.hat-var.hat)/mu.hat
  return(p.hat)
}

# Negative of the Likelihood function
bernoulliLogLik.neg <- function(par, data){
  -sum(dbinom(x=data, prob=par, size=1, log=TRUE))} #Sum is negative, as we are looking for a min wit
```

```
MLE.bernoulli <- function(data){
  MLE <- optim(par = .5,               # best guess for the parameter
               fn = bernoulliLogLik.neg,   # function to minimize
               data = data,       # data (an argument to fn)
               method = "Brent",       # required for univariate optimization
               lower = 0,              # lowest possible lambda
               upper = 1)              # reasonable upper bound

  # Note that our mle is hat(lambda) = xbar
  return(MLE$par)
}

bernoulli.plot <- function(data){

  dfActual <- data.frame(matrix(nrow = 2, ncol = 2))
  colnames(dfActual) <- c("Prop","Value")
  rownames(dfActual) <- c("0","1")

  ActualTab <- as.data.frame(data) %>% group_by(data) %>%
    summarize(count = n()) # Calculate the counts

  dfActual[1,1] <- prop.table(ActualTab$count)[1]
  dfActual[2,1] <- prop.table(ActualTab$count)[2]
  dfActual[1,2] <- 0
  dfActual[2,2] <- 1


  dfEstimate <- data.frame(matrix(nrow = 2, ncol = 2))
  colnames(dfEstimate) <- c("Prop","Value")
  rownames(dfEstimate) <- c("0","1")

  EstimateTab <- as.data.frame(data) %>% group_by(data) %>%
    summarize(count = n()) # Calculate the counts

  dfEstimate[1,1] <- 1-MOM.bernoulli(data)
  dfEstimate[2,1] <- MOM.bernoulli(data)
  dfEstimate[1,2] <- 0
  dfEstimate[2,2] <- 1


  p1 <- ggplot(mapping = aes(y=Prop, x=Value)) +
    geom_col(data = dfActual, position = "dodge", fill = "lightblue") +
    geom_col(data = dfEstimate, aes(group = Value),
             fill = "black", width = 0.01, position = position_dodge(width = 0.9))+
    ggtitle("MOM")

  dfActual <- data.frame(matrix(nrow = 2, ncol = 2))
  colnames(dfActual) <- c("Prop","Value")
  rownames(dfActual) <- c("0","1")

  ActualTab <- as.data.frame(data) %>% group_by(data) %>%
    summarize(count = n()) # Calculate the counts

  dfActual[1,1] <- prop.table(ActualTab$count)[1]
  dfActual[2,1] <- prop.table(ActualTab$count)[2]
  dfActual[1,2] <- 0
  dfActual[2,2] <- 1
  dfEstimate <- data.frame(matrix(nrow = 2, ncol = 2))
  colnames(dfEstimate) <- c("Prop","Value")
  rownames(dfEstimate) <- c("0","1")

  EstimateTab <- as.data.frame(data) %>% group_by(data) %>%
    summarize(count = n()) # Calculate the counts

  dfEstimate[1,1] <- 1-MLE.bernoulli(data)
  dfEstimate[2,1] <- MLE.bernoulli(data)
  dfEstimate[1,2] <- 0
  dfEstimate[2,2] <- 1


  p2 <- ggplot(mapping = aes(y=Prop, x=Value)) +
    geom_col(data = dfActual, position = "dodge", fill = "lightblue") +
    geom_col(data = dfEstimate, aes(group = Value),
             fill = "black", width = 0.01, position = position_dodge(width = 0.9))+
    ggtitle("MLE")

  return(p1+p2)

}
```

```
even.10 <- rbinom(n=10,          #number of observations
                  size=1,        #number of trials (size=1 for Bernoulli)
                  prob=.5)       #probability of success

paste("MOM: ", MOM.bernoulli(uneven.10),
      "MLE: ", MLE.bernoulli(uneven.10))

## [1] "MOM:  0.9 MLE:  0.900000000113646"
```

Plot is Given Below:

```
bernoulli.plot(even.10)
```

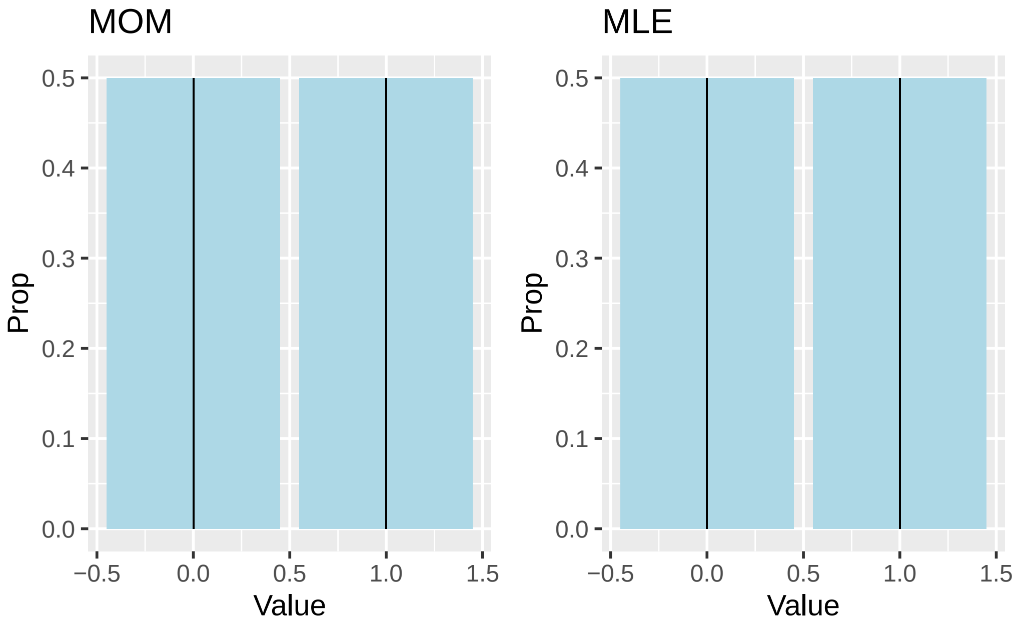


Figure 15: Bernoulli($n = 10$, $p = .5$)

```
uneven.10 <- rbinom(n=10,         #number of observations
                    size=1,         #number of trials (size=1 for a Bernoulli distribution)
                    prob=.7)        #probability of success

paste("MOM: ", MOM.bernoulli(uneven.10),
      "MLE: ", MLE.bernoulli(uneven.10))

## [1] "MOM:  0.5 MLE:  0.5"
```

Plot is Given Below:

```
bernoulli.plot(uneven.10)
```

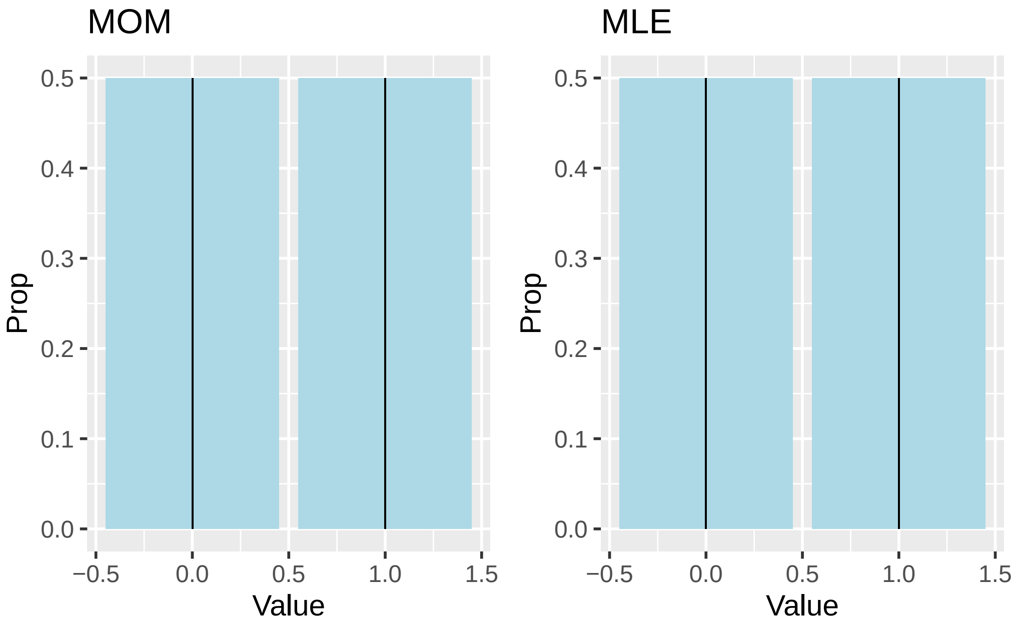


Figure 16: Bernoulli($n = 10$, $p = .7$)

(d) Generate a random sample of size $n = 25$ for your two sets of parameter(s). Calculate the method of moments estimator(s) and maximum likelihood estimator(s). In a 1×2 grid, plot a histogram (with bin size 1) of each set of data with (1) the method of moments estimated distribution, (2) the maximum likelihood estimated distribution, and superimpose the true distribution in both.

```
even.25 <- rbinom(n=25,         #number of observations
                  size=1,         #number of trials (size=1 for Bernoulli)
                  prob=.5)        #probability of success

paste("MOM:", MOM.bernoulli(even.25),
      "MLE:", round(MLE.bernoulli(even.25),2))

## [1] "MOM: 0.44 MLE: 0.44"
```

Plot is Given Below:

```
bernoulli.plot(even.25)
```

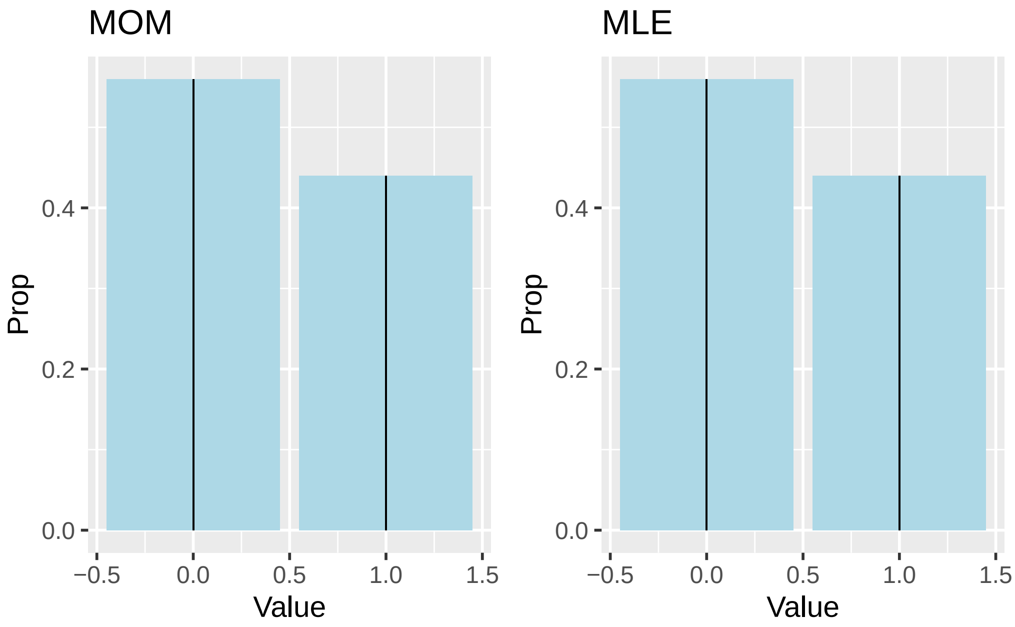


Figure 17: Bernoulli($n = 25$, $p = .5$)

```
uneven.25 <- rbinom(n=25,        #number of observations
                    size=1,        #number of trials (size=1 for Bernoulli)
                    prob=.7)       #probability of success

paste("MOM:", MOM.bernoulli(uneven.25),
      "MLE:", round(MLE.bernoulli(uneven.25),2))

## [1] "MOM: 0.72 MLE: 0.72"
```

Plot is Given Below:

```
bernoulli.plot(uneven.25)
```

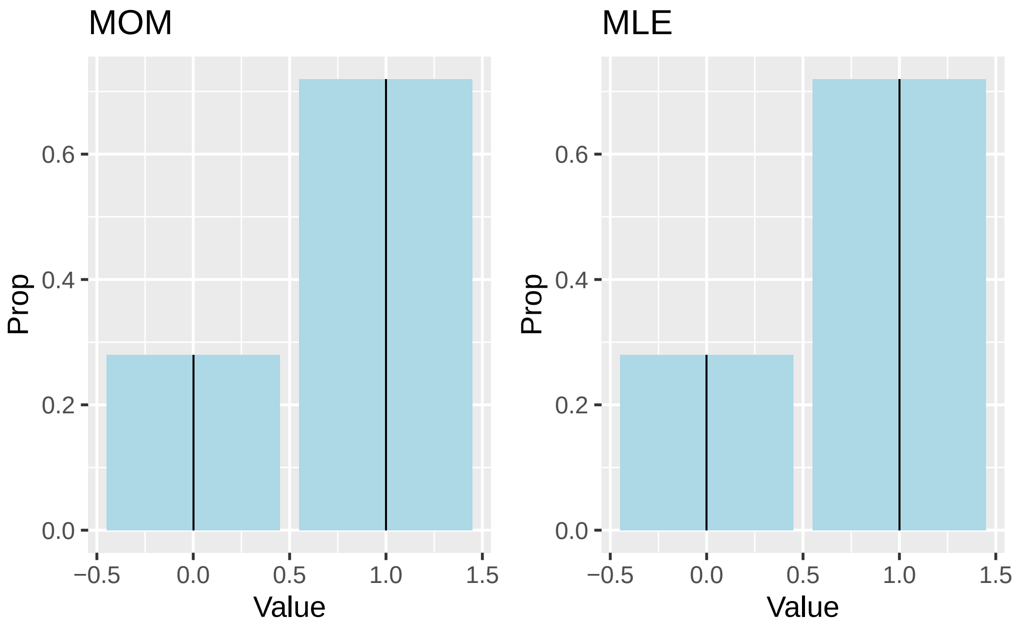


Figure 18: Bernoulli($n = 25$, $p = .7$)

(e) Generate a random sample of size $n = 100$ for your two sets of parameter(s). Calculate the method of moments estimator(s) and maximum likelihood estimator(s). In a 1×2 grid, plot a histogram (with bin size 1) of each set of data with (1) the method of moments estimated distribution, (2) the maximum likelihood estimated distribution, and superimpose the true distribution in both.

```
even.100 <- rbinom(n=100,           #number of observations
                   size=1,          #number of trials (size=1 for Bernoulli)
                   prob=.5)         #probability of success

paste("MOM:", round(MOM.bernoulli(even.100),2),
      "MLE:", round(MLE.bernoulli(even.100), 2))

## [1] "MOM: 0.48 MLE: 0.48"
```

Plot is Given Below:

```
bernoulli.plot(even.100)
```

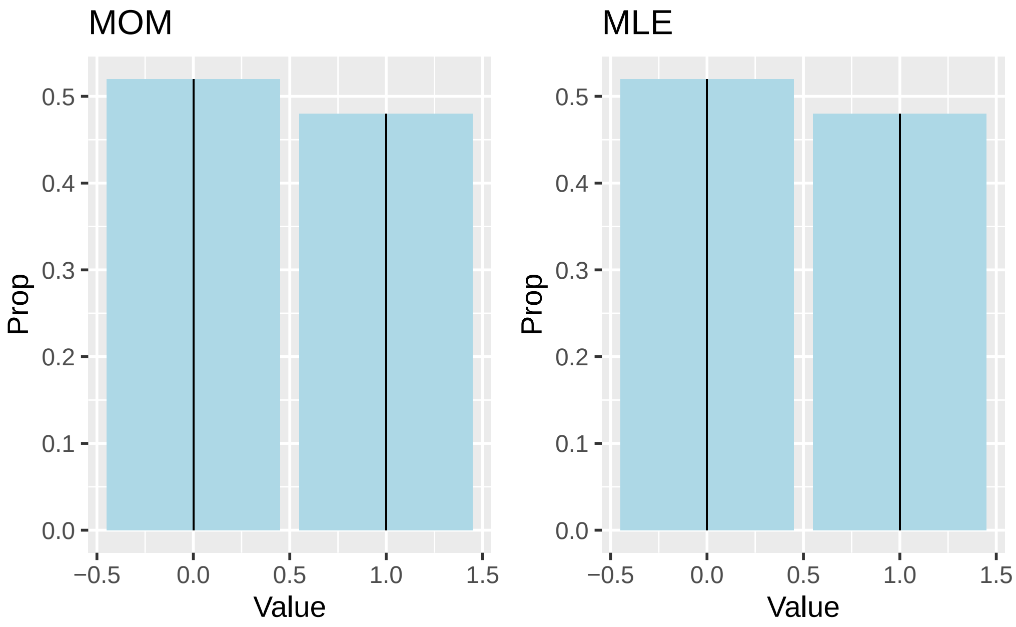


Figure 19: Bernoulli($n = 100$, $p = .5$)

```
uneven.100 <- rbinom(n=100,         #number of observations
                    size=1,         #number of trials (size=1 for Bernoulli)
                    prob=.7)        #probability of success

paste("MOM:", round(MOM.bernoulli(uneven.100),2),
      "MLE:", round(MLE.bernoulli(uneven.100), 2))

## [1] "MOM: 0.83 MLE: 0.83"
```

Plot is Given Below:

```
bernoulli.plot(uneven.100)
```

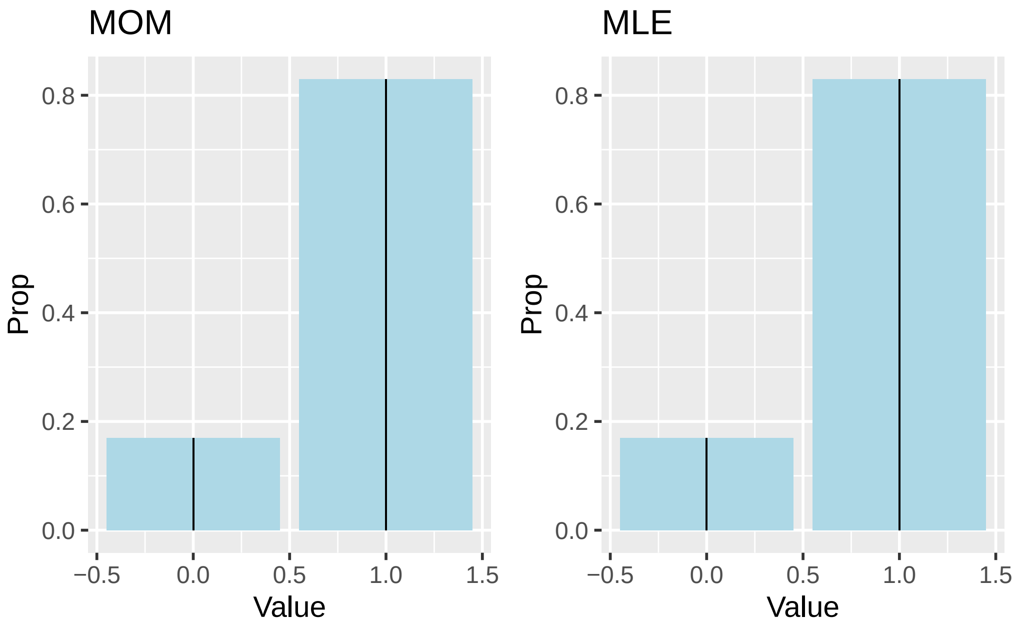


Figure 20: Bernoulli($n = 100$, $p = .7$)

(f) Generate a random sample of size $n = 100$ for your two sets of parameter(s). Calculate the method of moments estimator(s) and maximum likelihood estimator(s). In a 1×2 grid, plot a histogram (with bin size 1) of each set of data with (1) the method of moments estimated distribution, (2) the maximum likelihood estimated distribution, and superimpose the true distribution in both.

```
even.1000 <- rbinom(n=1000,         #number of observations
                  size=1,        #number of trials (size=1 for Bernoulli)
                  prob=.5)       #probability of success

paste("MOM:", round(MOM.bernoulli(even.1000),2),
      "MLE:", round(MLE.bernoulli(even.1000), 2))

## [1] "MOM: 0.54 MLE: 0.54"
```

Plot is given below:

```
bernoulli.plot(even.1000)
```

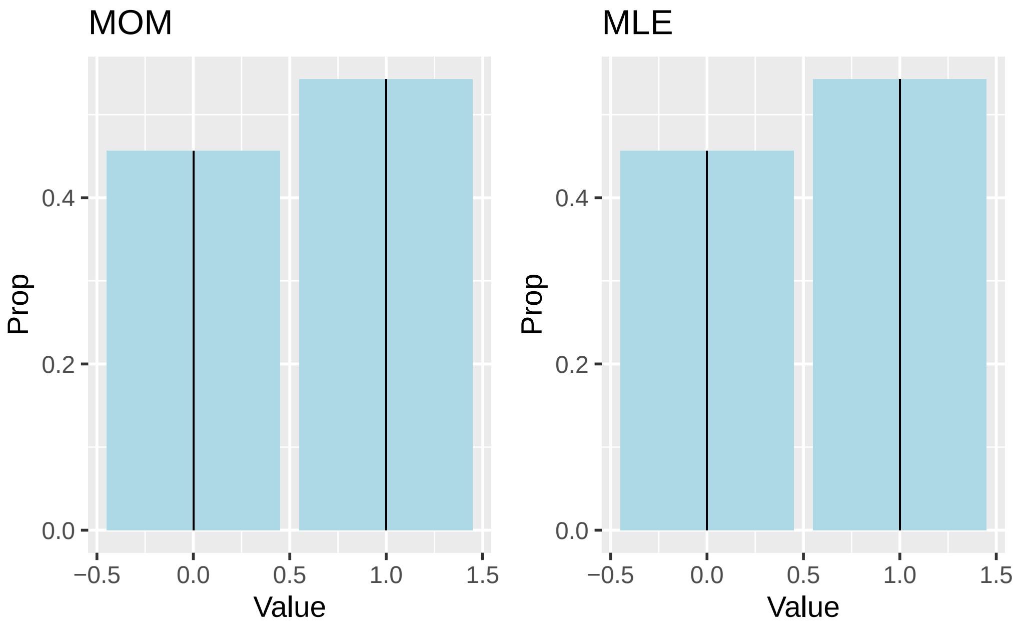


Figure 21: Bernoulli($n = 1000$, $p = .5$)

```
uneven.1000 <- rbinom(n=1000,          #number of observations
                      size=1,         #number of trials (size=1 for Bernoulli)
                      prob=.7)        #probability of success

MOM.bernoulli(uneven.1000)

## [1] 0.708

MLE.bernoulli(uneven.1000)

## [1] 0.708

paste("MOM:", round(MOM.bernoulli(uneven.1000),2),
      "MLE:", round(MLE.bernoulli(uneven.1000), 2))

## [1] "MOM: 0.71 MLE: 0.71"
```

Plot is given below:

```
bernoulli.plot(uneven.1000)
```

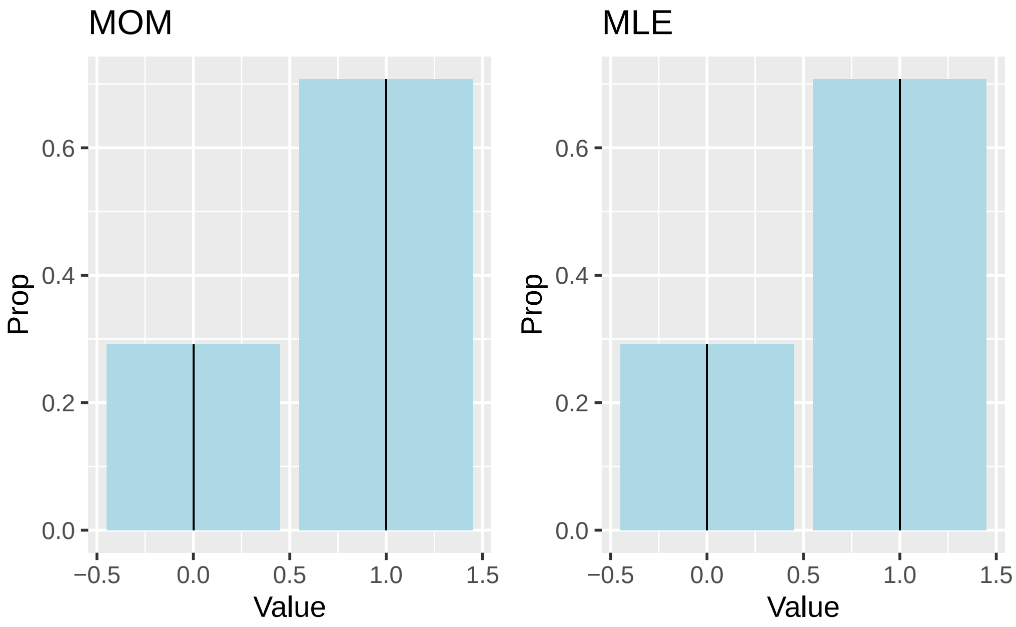


Figure 22: Bernoulli($n = 1000$, $p = .7$)

(g) Comment on the results of parts (c)-(f).

From the plots in the results of parts (c)-(f), we can see that both the Method of Moments estimator and the Maximum Likelihood Estimator were able to provide a perfect estimate for the parameter p regardless of our random sample size n. Therefore, I feel it is reasonable to conclude that using larger random sample sizes to find the MOM and MLE of a Bernoulli distribution is unnecessary.

References

(2021).

Ahsanullah, M., Kibria, B. G., and Shakil, M. (2014). Normal distribution. In *Normal and Student st Distributions and Their Applications*, pages 7–50. Springer.

Cthaeh, T. (2020). The bernoulli distribution: Intuitive understanding.

Forbes, C., Evans, M., Hastings, N., and Peacock, B. (2011). *Statistical Distributions.* John Wiley & Sons, Inc., Hoboken, New Jersey.

Kleinbaum, D. G. and Klein, M. (2010). Introduction to Logistic Regression. In Kleinbaum, D. G. and Klein, M., editors, *Logistic Regression: A Self-Learning Text*, Statistics for Biology and Health, pages 1–39. Springer, New York, NY.

Patel, J. K. and Read, C. B. (1996). *Handbook of the normal distribution*, volume 150. CRC Press.

Pedersen, T. L. (2020). *patchwork: The Composer of Plots.* R package version 1.1.1.

R Core Team (2021). *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria.

Vukadin, O., Lee, S., Williams, C., and Khim, J. (2021). Bernoulli distribution | Brilliant.

Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis.* Springer-Verlag New York.

Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D., François, R., Grolemund, G., Hayes, A., Henry, L., Hester, J., Kuhn, M., Pedersen, T. L., Miller, E., Bache, S. M., Müller, K., Ooms, J., Robinson, D., Seidel, D. P., Spinu, V., Takahashi, K., Vaughan, D., Wilke, C., Woo, K., and Yutani, H. (2019). Welcome to the tidyverse. *Journal of Open Source Software*, 4(43):1686.