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. 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:

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

   Mean:

$$E(X) = p$$

   Standard Deviation:

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

   Skewness:

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

   Kurtosis:

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

   (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)

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

mean(even.10)                     #mean

## [1] 0.7

sd(even.10)                       #standard deviation

## [1] 0.4830459

skewness(even.10)                 #skewness

## [1] -0.7452708

kurtosis(even.10)                 #kurtosis

## [1] -1.572857
```

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

mean(even.25)                     #mean

## [1] 0.56

sd(even.25)                       #standard deviation

## [1] 0.5066228

skewness(even.25)                 #skewness

## [1] -0.2273881

kurtosis(even.25)                 #kurtosis

## [1] -2.02454

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

mean(even.100)                    #mean

## [1] 0.45

sd(even.100)                      #standard deviation

## [1] 0.5

skewness(even.100)                #skewness

## [1] 0.198

kurtosis(even.100)                #kurtosis

## [1] -1.9803

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

mean(even.1000)                   #mean

## [1] 0.467

sd(even.1000)                     #standard deviation

## [1] 0.4991595

skewness(even.1000)               #skewness

## [1] 0.1320901

kurtosis(even.1000)               #kurtosis

## [1] -1.984534
```

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

mean(uneven.10)                    #mean

## [1] 0.6

sd(uneven.10)                      #standard deviation

## [1] 0.5163978

skewness(uneven.10)                #skewness

## [1] -0.3485685

kurtosis(uneven.10)                #kurtosis

## [1] -2.055

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

mean(uneven.25)                    #mean

## [1] 0.76

sd(uneven.25)                      #standard deviation

## [1] 0.4358899

skewness(uneven.25)                #skewness

## [1] -1.145243

kurtosis(uneven.25)                #kurtosis

## [1] -0.7121684

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

mean(uneven.100)                   #mean

## [1] 0.72

sd(uneven.100)                     #standard deviation

## [1] 0.4512609

skewness(uneven.100)               #skewness

## [1] -0.9652953

kurtosis(uneven.100)               #kurtosis

## [1] -1.078693
```

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

mean(uneven.1000)                  #mean
```

```
## [1] 0.69
```

```r
sd(uneven.1000)                    #standard deviation
```

```
## [1] 0.4627247
```

```r
skewness(uneven.1000)              #skewness
```

```
## [1] -0.8204015
```

```r
kurtosis(uneven.1000)              #kurtosis
```

```
## [1] -1.328267
```

(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(ggplot2)

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 Berno
  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 

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))

  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) %>%
```

```r
    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))

  return(p1+p2)



}


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

MOM.bernoulli(even.10)

## [1] 0.4

MLE.bernoulli(even.10)

## [1] 0.4

bernoulli.plot(even.10)

## Error in as.data.frame(data) %>% group_by(data) %>% summarize(count = n()):  could
not find function "%>%"

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

MOM.bernoulli(uneven.10)

## [1] 0.6

MLE.bernoulli(uneven.10)

## [1] 0.6

bernoulli.plot(uneven.10)

## Error in as.data.frame(data) %>% group_by(data) %>% summarize(count = n()):  could
not find function "%>%"
```

(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.

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

MOM.bernoulli(even.25)

## [1] 0.56

MLE.bernoulli(even.25)

## [1] 0.56

bernoulli.plot(even.25)

## Error in as.data.frame(data) %>% group_by(data) %>% summarize(count = n()):  could
not find function "%>%"

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

MOM.bernoulli(uneven.25)

## [1] 0.6

MLE.bernoulli(uneven.25)

## [1] 0.6

bernoulli.plot(uneven.25)

## Error in as.data.frame(data) %>% group_by(data) %>% summarize(count = n()):  could
not find function "%>%"
```

(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.

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

MOM.bernoulli(even.100)

## [1] 0.56

MLE.bernoulli(even.100)

## [1] 0.56

bernoulli.plot(even.100)

## Error in as.data.frame(data) %>% group_by(data) %>% summarize(count = n()):  could
not find function "%>%"

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

MOM.bernoulli(uneven.100)
```

```
## [1] 0.68

MLE.bernoulli(uneven.100)

## [1] 0.68

bernoulli.plot(uneven.100)

## Error in as.data.frame(data) %>% group_by(data) %>% summarize(count = n()):  could
not find function "%>%"
```

(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 a Bernoulli distribution)
                  prob=.5)         #probability of success

MOM.bernoulli(even.1000)

## [1] 0.5

MLE.bernoulli(even.1000)

## [1] 0.5

bernoulli.plot(even.1000)

## Error in as.data.frame(data) %>% group_by(data) %>% summarize(count = n()):  could
not find function "%>%"

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

MOM.bernoulli(uneven.1000)

## [1] 0.68

MLE.bernoulli(uneven.1000)

## [1] 0.68

bernoulli.plot(uneven.1000)

## Error in as.data.frame(data) %>% group_by(data) %>% summarize(count = n()):  could
not find function "%>%"
```

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