ECE Final Project Writing

Question 1

B. The probability is 0.5. This is because all outcomes are equally likely. There are 10 possible outcomes and 5 of those outcomes are odd rolls. Therefore the probability is 5/10.
C. Part A agrees with my theoretical result as 0.5007 is very close to 0.5.
D. The probability that X is a odd value given that odds are twice as likely to come up as evens is 2/3. In this case we can think of the example of a 15 sided dice with all numbers equally likely, the dice is similar to the 10 sided dice in every way however the extra 5 sides are reserved for the 5 odd numbers. This will have the same probability as the 10 sided dice with twice the probability of getting an odd number. In this case there are 10 odd values that can be rolled and 15 values in total. Therefore 10/15 = 2/3. This agrees with my computed value from my code. My code gave me the value 0.6677 which is very close to 2/3.

ECE131A_Project

Olivia Smith

2023-03-21

1.

a

```
set.seed(123) # So you can check the replication of this simulation

#Rolling the samples
samp_rolls50 <- replicate(50, sample(1:10, 1))
samp_rolls100 <- replicate(100, sample(1:10, 1))
samp_rolls1000 <- replicate(1000, sample(1:10, 1))
samp_rolls2000 <- replicate(2000, sample(1:10, 1))
samp_rolls3000 <- replicate(3000, sample(1:10, 1))
samp_rolls10000 <- replicate(10000, sample(1:10, 1))
samp_rolls100000 <- replicate(100000, sample(1:10, 1))

#Finding the probability of rolling odds for each one

p50<-mean(samp_rolls50 == 1|samp_rolls50==3|samp_rolls50 == 5|samp_rolls50==7|samp_rolls50 == 9)
p100<-mean(samp_rolls100 == 1|samp_rolls100==3|samp_rolls100 == 5|samp_rolls100==7|samp_rolls100
== 9)
p1000<-mean(samp_rolls1000 == 1|samp_rolls1000==3|samp_rolls1000 == 5|samp_rolls1000==7|samp_rol
ls1000 == 9)
p2000<-mean(samp_rolls2000 == 1|samp_rolls2000==3|samp_rolls2000 == 5|samp_rolls2000==7|samp_rol
ls2000 == 9)
p3000<-mean(samp_rolls3000 == 1|samp_rolls3000==3|samp_rolls3000 == 5|samp_rolls3000==7|samp_rol
ls3000 == 9)
p10000<-mean(samp_rolls10000 == 1|samp_rolls10000==3|samp_rolls10000 == 5|samp_rolls10000==7|sam
p_rolls10000 == 9)
p100000<-mean(samp_rolls100000 == 1|samp_rolls100000==3|samp_rolls100000 == 5|samp_rolls100000==
7|samp_rolls100000 == 9)

#Printing those probabilities
p50 # 50 tosses
```

```
## [1] 0.6
```

```
p100 # 100 tosses
```

```
## [1] 0.47
```

```
p1000 # 1000 tosses
```

```
## [1] 0.522
```

```
p2000 # 2000 tosses
```

```
## [1] 0.484
```

```
p3000 # 3000 tosses
```

```
## [1] 0.5083333
```

```
p10000 # 10000 tosses
```

```
## [1] 0.5007
```

Due to the idea of the law of large numbers. The closer that you get to infinite the more likely you are going to obtain the actual probability. In my simulation therefore, we would take the largest number as the closest number to the probability of obtaining an odd number. In this case that probability is 0.5007.

d

```
double <-c(2,1,2,1,2,1,2,1,2,1) #vector for weights of 1-10

#generating sample rolls with probability of double for odd numbers
samp2_rolls50 <- replicate(50, sample(1:10, 1, prob = double))
samp2_rolls100 <- replicate(100, sample(1:10, 1,prob = double))
samp2_rolls1000 <- replicate(1000, sample(1:10, 1,prob = double))
samp2_rolls2000 <- replicate(2000, sample(1:10, 1,prob = double))
samp2_rolls3000 <- replicate(3000, sample(1:10, 1,prob = double))
samp2_rolls10000 <- replicate(10000, sample(1:10, 1,prob = double))
samp2_rolls100000 <- replicate(100000, sample(1:10, 1,prob = double))

#function for calculating and printing the mean
mean.funct <- function(a) {
   mean<-mean(a == 1|a==3|a == 5|a==7|a == 9)
   print(mean)
}

#calculating and printing the mean
mean.funct(samp2_rolls50)
```

```
## [1] 0.7
```

2) Maximum Likelihood Estimation

a. $\log(f_{x_1,x_2,\dots x_n}(x_1,x_2\dots x_n|M,\sigma)) = -\frac{n}{2}\log(2\pi\sigma^2) - \sum_{i=1}^{n}\frac{(x_i-M)^2}{2\sigma^2}$

We can use the property that if $X \perp\!\!\!\perp Y$ independent

$f_{X,Y}(x,y) = f_X(x)\cdot f_Y(y)$

$\log(f_{x_1}(x_1|M,\sigma)\cdot f_{x_2}(x_2|M,\sigma)\cdot\dots\cdot f_{x_n}(x_n|M,\sigma))$

Since $x_1,x_2\dots x_n$ are gaussian and i.i.d.

$f_{x_1}(x_1|M,\sigma) = \frac{1}{\sqrt{2\pi}\sigma}\cdot e^{-\frac{(x_1-M)^2}{2\sigma^2}} = f_{x_2}(x_2|M,\sigma)$

Therefore

$\log\left(\frac{1}{\sqrt{2\pi}\sigma}\cdot e^{-\frac{(x_1-M)^2}{2\sigma^2}}\cdot\frac{1}{\sqrt{2\pi}\sigma}\cdot e^{-\frac{(x_2-M)^2}{2\sigma^2}}\cdot\dots\cdot\frac{1}{\sqrt{2\pi}\sigma}\cdot e^{-\frac{(x_n-M)^2}{2\sigma^2}}\right) = \log\left(\left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^n\right) + \log\left(e^{-\sum_{i=1}^{n}\frac{x_i-M}{2\sigma^2}}\right)$

$$\boxed{= -\frac{n}{2}\log(2\pi\sigma^2) - \sum_{i=1}^{n}\frac{(x_i-M)^2}{2\sigma^2}}$$

b)

$M_{MLE} = \arg\max_M[\log(f_{x_1,x_2,\dots,x_n}(x_1,x_2,x_3\dots,x_n|M,\sigma))]$

$= \arg_M\max[-\frac{n}{2}\log(2\pi\sigma^2) - \sum_{i=1}^{n}\frac{(x_i-M)^2}{2\sigma^2}]$

To find max set derivative to 0

$0 = \frac{\partial}{\partial M}\left(-\frac{n}{2}\log(2\pi\sigma^2) - \sum_{i=1}^{n}\frac{(x_i-M)^2}{2\sigma^2}\right)$ 0

$\frac{\partial}{\partial M}\left(\sum_{i=1}^{n}\frac{(x_i-M)^2}{2\sigma^2}\right) = \sum_{i=1}^{n}\frac{x_i-M}{\sigma^2} = \sum_{i=1}^{n}x_i = nM$ $\boxed{M_{MLE} = \frac{1}{n}\sum_{i=1}^{n}x_i}$

Therefore M_{MLE} is equal to the average of all x_i's

$\sigma_{MLE} = \arg_\sigma\max[\log(f_{x_1,x_2,\dots,x_n}(x_1,x_2,x_3\dots,x_n|M_{MLE},\sigma))]$

$= \arg_\sigma\max[-\frac{n}{2}\log(2\pi\sigma^2) - \sum_{i=1}^{n}\frac{(x_i-M_{MLE})^2}{2\sigma^2}]$

$\frac{d}{d\sigma_{MLE}} = \frac{d}{d\sigma}\left(-\frac{n}{2}\log(2\pi\sigma^2) - \sum_{i=1}^{n}\frac{(x_i-M_{MLE})^2}{2\sigma^2}\right)$

$u = 2\pi\sigma^2$

$du = 4\pi\sigma$

$\frac{d}{du}\log(u) = \frac{1}{u}$

$-\frac{n}{2}\left[\frac{1}{2\pi\sigma^2}\cdot 4\pi\sigma\right] - \sum_{i=1}^{n}\frac{-4\sigma(x_i-M_{MLE})^2}{4\sigma^4} = -\frac{n}{\sigma} + \sum_{i=1}^{n}\frac{(x_i-M_{MLE})^2}{\sigma^3} = 0$

$\frac{n}{\sigma} = \sum_{i=1}^{n}\frac{(x_i-M_{MLE})^2}{\sigma^3}$

$\downarrow$

σ is not dependent on counter i so we can take out

$$n\sigma^2 = \sum_{i=1}^{n} (x_i - \mu_{MLE})^2$$

$$\boxed{\sigma_{MLE} = \sqrt{\frac{\sum (x_i - \mu_{MLE})^2}{n}}}$$

```
mean.funct(samp2_rolls100)
```

```
## [1] 0.68
```

```
mean.funct(samp2_rolls1000)
```

```
## [1] 0.671
```

```
mean.funct(samp2_rolls2000)
```

```
## [1] 0.675
```

```
mean.funct(samp2_rolls3000)
```

```
## [1] 0.6643333
```

```
mean.funct(samp2_rolls10000)
```

```
## [1] 0.6707
```

```
mean.funct(samp2_rolls100000)
```

```
## [1] 0.66647
```

Based on my simulation of odd value having twice as likely probability of showing up than an even number the probability of obtaining an odd number is 0.6677. Based on mathematical analysis and probability theory we expect the probability that X has an odd value to be 2/3. This agrees with empirical probability as the theoretical is equal to 0.666667 which is very close to 0.6677.

2

```
data<-read.delim(file.choose(), header=FALSE)
```

b

```
mu_MLE = mean(data$V1)
print(paste("mu_MLE =", mu_MLE) )
```

```
## [1] "mu_MLE = 20.0134506689949"
```

```
sigma_MLE = sqrt(var(data$V1))
print(paste("sigma_MLE =", sigma_MLE) )
```

```
## [1] "sigma_MLE = 9.99246977093787"
```

C

```
library(mosaic)
```

```
## Registered S3 method overwritten by 'mosaic':
##   method                           from
##   fortify.SpatialPolygonsDataFrame ggplot2
```

```
##
## The 'mosaic' package masks several functions from core packages in order to add
## additional features.  The original behavior of these functions should not be affected by thi
s.
```

```
##
## Attaching package: 'mosaic'
```

```
## The following objects are masked from 'package:dplyr':
##
##     count, do, tally
```

```
## The following object is masked from 'package:Matrix':
##
##     mean
```

```
## The following object is masked from 'package:ggplot2':
##
##     stat
```

```
## The following objects are masked from 'package:stats':
##
##     binom.test, cor, cor.test, cov, fivenum, IQR, median, prop.test,
##     quantile, sd, t.test, var
```

```
## The following objects are masked from 'package:base':
##
##     max, mean, min, prod, range, sample, sum
```

3 c) $P(B=0, T=1, S=0, A\leq 55) = P(T, S, A \mid B=0)$

Definition of conditional Probability $\rightarrow$ $P(A \mid B) = \dfrac{P(A \cap B)}{P(B)}$

$$P(T, S, A \mid B=0) = \frac{P(T \cap S \cap A \cap B)}{P(B=0)}$$

From assumption of conditional independence

$$P(T, S, A \mid B=0) = P(T \mid B=0) \cdot P(S \mid B=0) \cdot P(A \mid B=0)$$

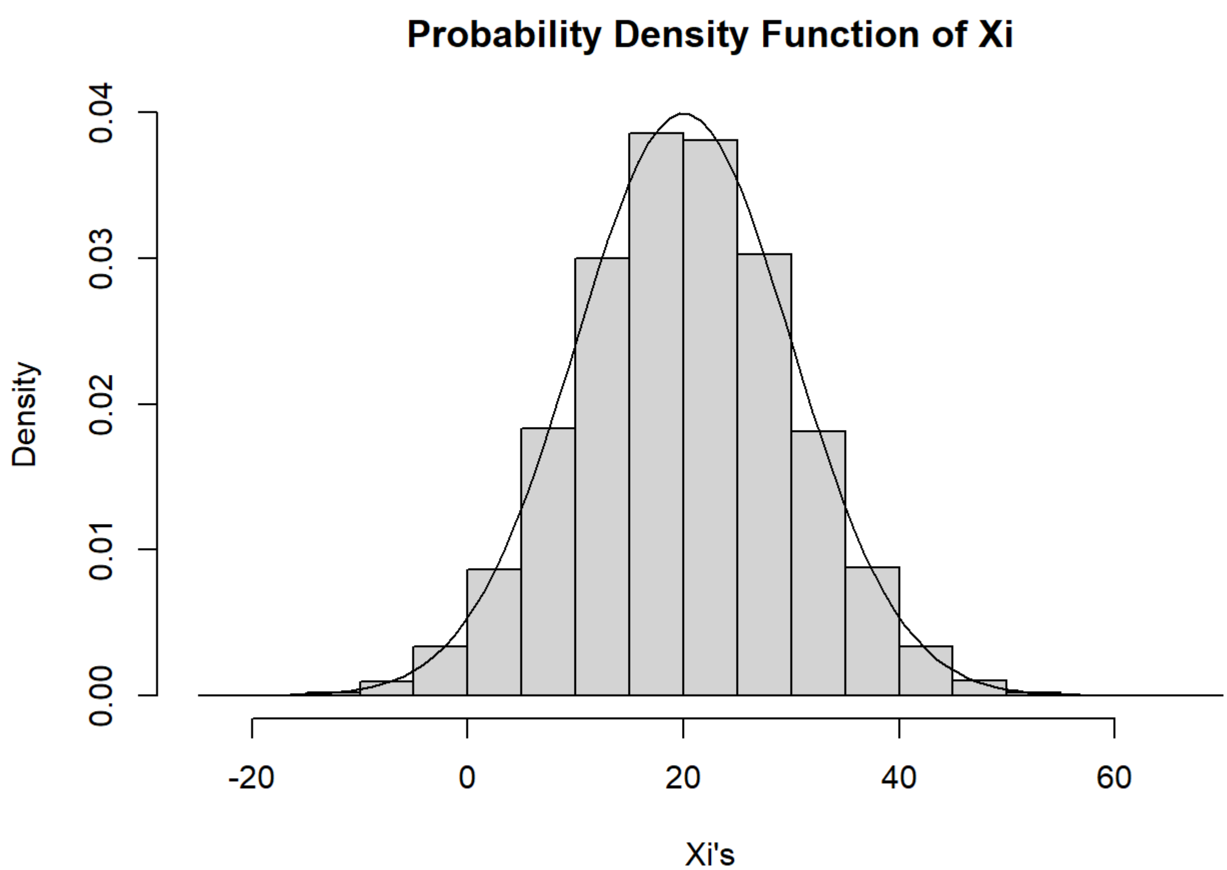
Therefore

$$P(T \cap S \cap A \cap B) = P(T \mid B=0) \cdot P(S \mid B=0) \cdot P(A \mid B=0) \cdot P(B=0)$$

3d) $P(B=0 \mid T=1, S=0, A\leq 55) = \dfrac{P(T=1, S=0, A\leq 55 \mid B=0) \cdot P(B=0)}{P(T=1, S=0, A\leq 55)}$

Based on the probabilities listed in my code we know that the probability of a female, who's age is below 55 and who is a large spender will buy this produce is 0.303. While the probability that she does not buy it is 0.523. Since the probability of her not buying it is above the probability of her buying it. I would predict that she would not buy this product.

```
hist(data$V1, main ="Probability Density Function of Xi", xlab ="Xi's", prob = TRUE) # maybe cha
nge the bin width
curve(dnorm(x, mu_MLE,sigma_MLE), add = TRUE)
```




```
userData <-read.csv("user_data.csv")
bought <- userData$Bought #B
spenderType <- userData$SpenderType #T
sex <- userData$Sex #S
age <- userData$Age #A
```

3

a

PMFs for B, T, S, and A

```
#B
boughtTable<-prop.table(table(bought))
print("The proportion table for people that bought is shown below where 1 = those that bought an
d 0 = those that did not buy")
```

```
## [1] "The proportion table for people that bought is shown below where 1 = those that bought a
nd 0 = those that did not buy"
```

```
boughtTable
```

```
## bought
##         0         1
## 0.6144307 0.3855693
```

```
#T
spenderTypeTable<-prop.table(table(spenderType))
print("The proportion table for spender type is shown below where 1 = larger spender, 2 = medium
spender, 3 = smaller spender")
```

```
## [1] "The proportion table for spender type is shown below where 1 = larger spender, 2 = mediu
m spender, 3 = smaller spender"
```

```
spenderTypeTable
```

```
## spenderType
##         1         2         3
## 0.2570462 0.2074408 0.5355130
```

```
#S
sexTable<-prop.table(table(sex))
print("The proportion table for sex type is shown below where 1 = Male and 0 = Female)")
```

```
## [1] "The proportion table for sex type is shown below where 1 = Male and 0 = Female)"
```

```
sexTable
```

```
## sex
##         0         1
## 0.6459977 0.3540023
```

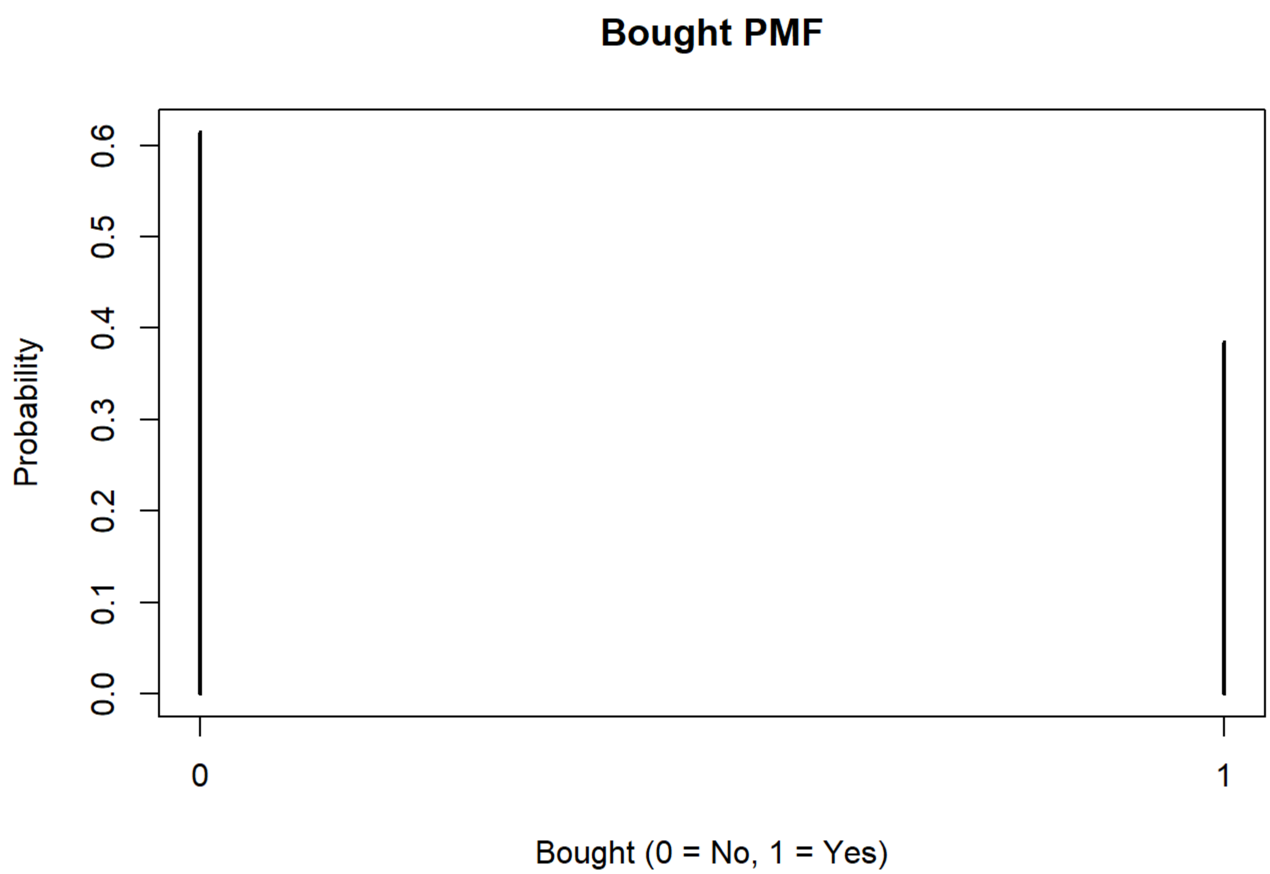
```
#A
ageTable<-prop.table(table(age))
print("The proportion table for age type is shown below")
```

```
## [1] "The proportion table for age type is shown below"
```

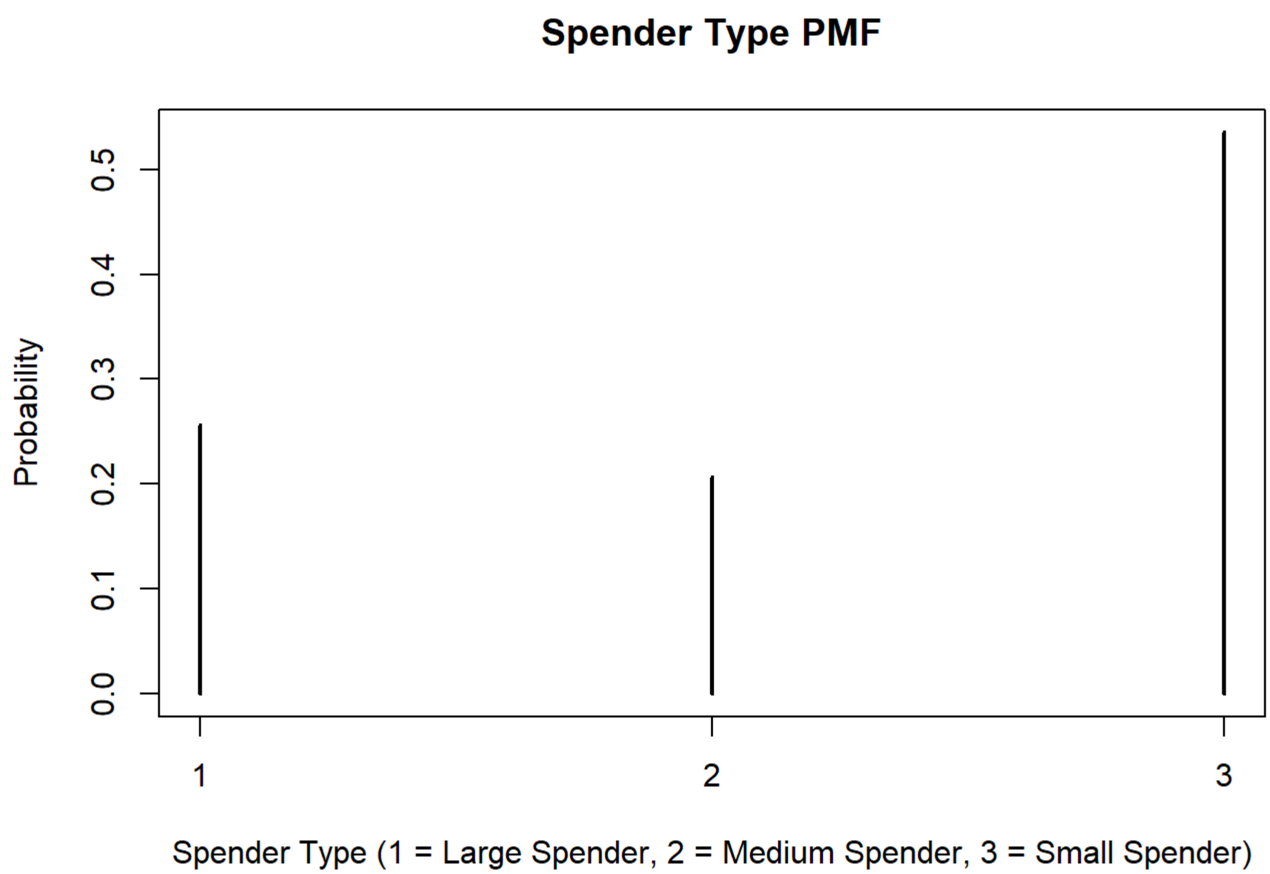
```
ageTable
```

```
## age
##          15          16          17          18          19          20
## 0.001127396 0.013528749 0.011273957 0.007891770 0.013528749 0.007891770
##          21          22          23          24          25          26
## 0.003382187 0.004509583 0.007891770 0.009019166 0.002254791 0.004509583
##          27          28          29          30          31          32
## 0.002254791 0.002254791 0.007891770 0.006764374 0.022547914 0.018038331
##          33          34          35          36          37          38
## 0.040586246 0.036076663 0.025930101 0.038331454 0.046223224 0.029312289
##          39          40          41          42          43          44
## 0.038331454 0.028184893 0.023675310 0.028184893 0.041713641 0.029312289
##          45          46          47          48          49          50
## 0.039458850 0.021420519 0.024802706 0.021420519 0.019165727 0.023675310
##          51          52          53          54          55          56
## 0.025930101 0.015783540 0.013528749 0.021420519 0.019165727 0.012401353
##          57          58          59          60          61          62
## 0.019165727 0.006764374 0.010146561 0.016910936 0.010146561 0.010146561
##          63          64          65          66          67          68
## 0.013528749 0.010146561 0.011273957 0.007891770 0.005636979 0.001127396
##          69          70          71          72          73          74
## 0.010146561 0.004509583 0.005636979 0.003382187 0.005636979 0.002254791
##          75          76          77          78          79          80
## 0.005636979 0.003382187 0.005636979 0.002254791 0.004509583 0.003382187
##          81          84          85          86          89          95
## 0.001127396 0.001127396 0.002254791 0.003382187 0.001127396 0.001127396
```

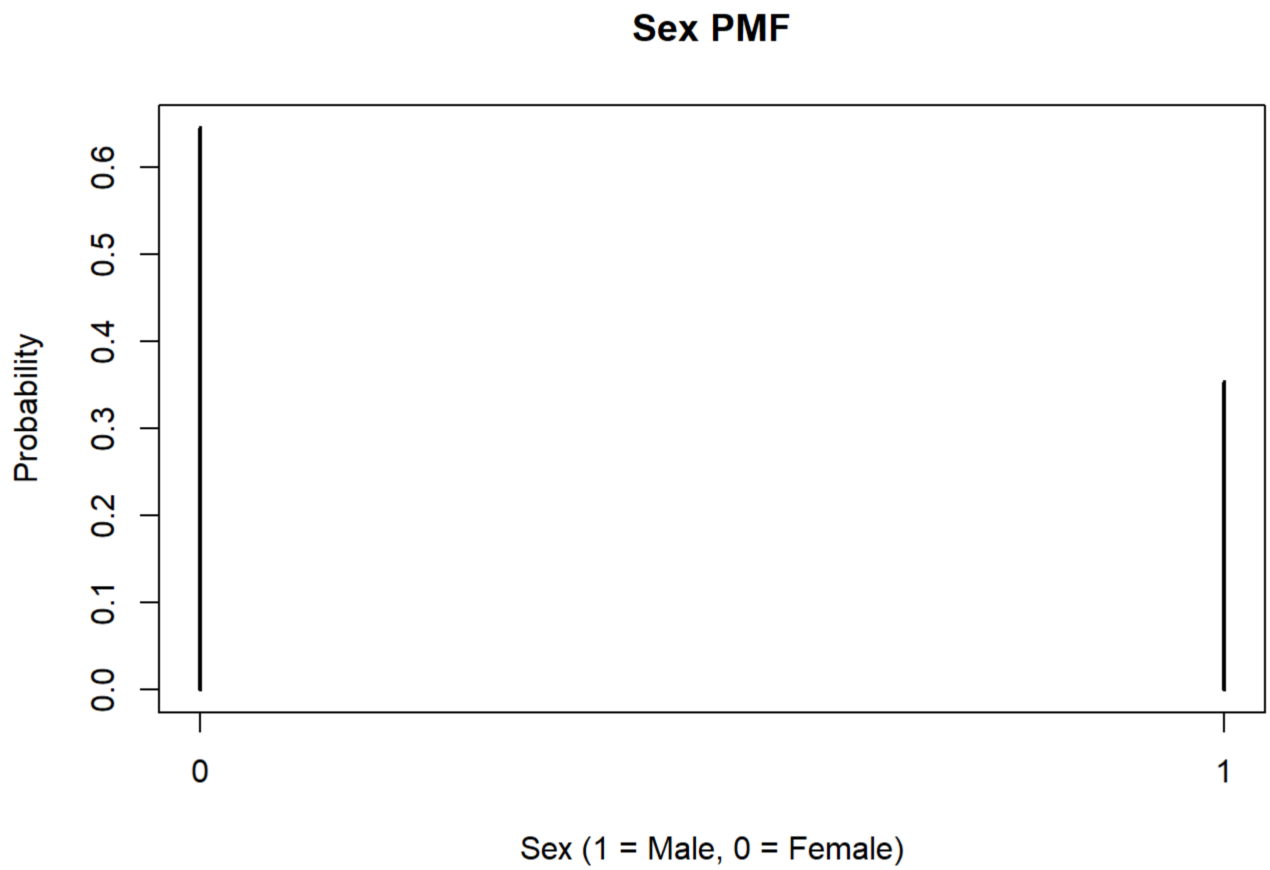
```
plot(boughtTable, main = "Bought PMF", ylab= "Probability",xlab="Bought (0 = No, 1 = Yes)")
```



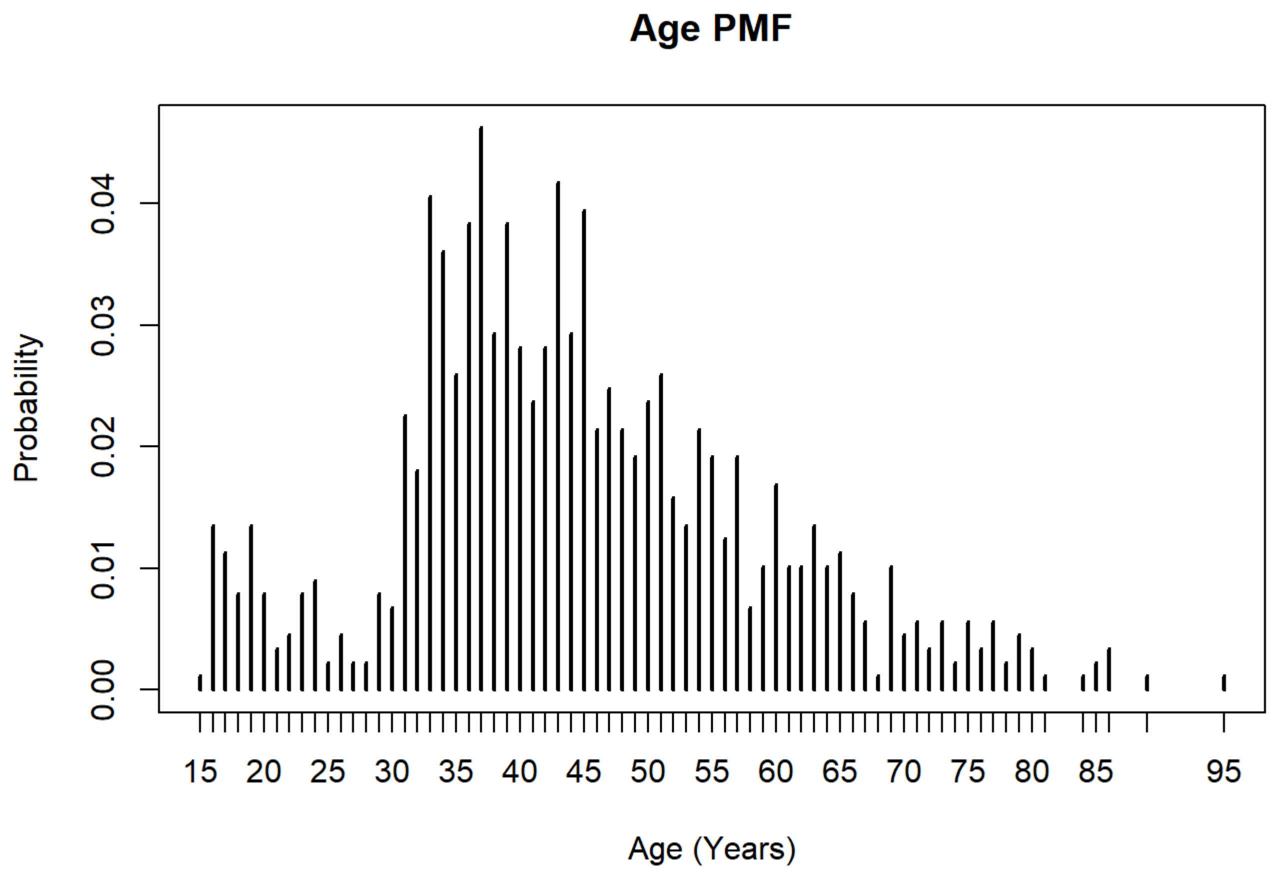

```
plot(spenderTypeTable, main = "Spender Type PMF", ylab= "Probability",xlab="Spender Type (1 = La
rge Spender, 2 = Medium Spender, 3 = Small Spender)")
```




```
plot(sexTable, main = "Sex PMF", ylab= "Probability",xlab="Sex (1 = Male, 0 = Female)")
```




```
plot(ageTable, main = "Age PMF", ylab= "Probability",xlab="Age (Years)")
```



b

```
#Sorts out the data into bought or not bought
B0<-userData[userData$Bought == 0,]
B1<-userData[userData$Bought == 1,]

#calculates the proportion tables to each condition
B0condSpenderType <- prop.table(table(B0$SpenderType))
B0condSex <- prop.table(table(B0$Sex))
B0condAge <- prop.table(table(B0$Age))

B1condSpenderType <- prop.table(table(B1$SpenderType))
B1condSex <- prop.table(table(B1$Sex))
B1condAge <- prop.table(table(B1$Age))

#prints out the tables for each condition
print("The proportion table for spender type conditioned on not bought is shown below where 1 = larger spender, 2 = medium spender, 3 = smaller spender")
```

```
## [1] "The proportion table for spender type conditioned on not bought is shown below where 1 = larger spender, 2 = medium spender, 3 = smaller spender"
```

```
B0condSpenderType
```

```
##
##         1         2         3
## 0.1688073 0.1926606 0.6385321
```

```
print("The proportion table for sex type conditioned on not bought is shown below where 1 = Male
and 0 = Female)")
```

```
## [1] "The proportion table for sex type conditioned on not bought is shown below where 1 = Mal
e and 0 = Female)"
```

```
B0condSex
```

```
##
##         0         1
## 0.8477064 0.1522936
```

```
print("The proportion table for age type conditioned on not bought is shown below")
```

```
## [1] "The proportion table for age type conditioned on not bought is shown below"
```

```
B0condAge
```

```
##
##          16          17          18          19          20          21
## 0.003669725 0.011009174 0.003669725 0.007339450 0.005504587 0.001834862
##          22          23          24          25          26          27
## 0.005504587 0.009174312 0.009174312 0.003669725 0.005504587 0.001834862
##          29          30          31          32          33          34
## 0.005504587 0.003669725 0.023853211 0.018348624 0.044036697 0.038532110
##          35          36          37          38          39          40
## 0.034862385 0.051376147 0.045871560 0.033027523 0.031192661 0.031192661
##          41          42          43          44          45          46
## 0.023853211 0.023853211 0.049541284 0.029357798 0.042201835 0.020183486
##          47          48          49          50          51          52
## 0.022018349 0.016513761 0.018348624 0.018348624 0.022018349 0.022018349
##          53          54          55          56          57          58
## 0.011009174 0.023853211 0.020183486 0.014678899 0.016513761 0.007339450
##          59          60          61          62          63          64
## 0.009174312 0.016513761 0.012844037 0.012844037 0.009174312 0.009174312
##          65          66          67          69          70          71
## 0.009174312 0.009174312 0.005504587 0.011009174 0.003669725 0.005504587
##          72          73          74          75          76          77
## 0.005504587 0.001834862 0.003669725 0.005504587 0.005504587 0.005504587
##          79          80          81          84          85          86
## 0.007339450 0.005504587 0.001834862 0.001834862 0.003669725 0.005504587
##          89
## 0.001834862
```

```
print("The proportion table for spender type conditioned on bought is shown below where 1 = larg
er spender, 2 = medium spender, 3 = smaller spender")
```

```
## [1] "The proportion table for spender type conditioned on bought is shown below where 1 = lar
ger spender, 2 = medium spender, 3 = smaller spender"
```

```
B1condSpenderType
```

```
##
##         1         2         3
## 0.3976608 0.2309942 0.3713450
```

```
print("The proportion table for sex type conditioned on bought is shown below where 1 = Male and
0 = Female)")
```

```
## [1] "The proportion table for sex type conditioned on bought is shown below where 1 = Male an
d 0 = Female)"
```

```
B1condSex
```

```
##
##         0         1
## 0.3245614 0.6754386
```

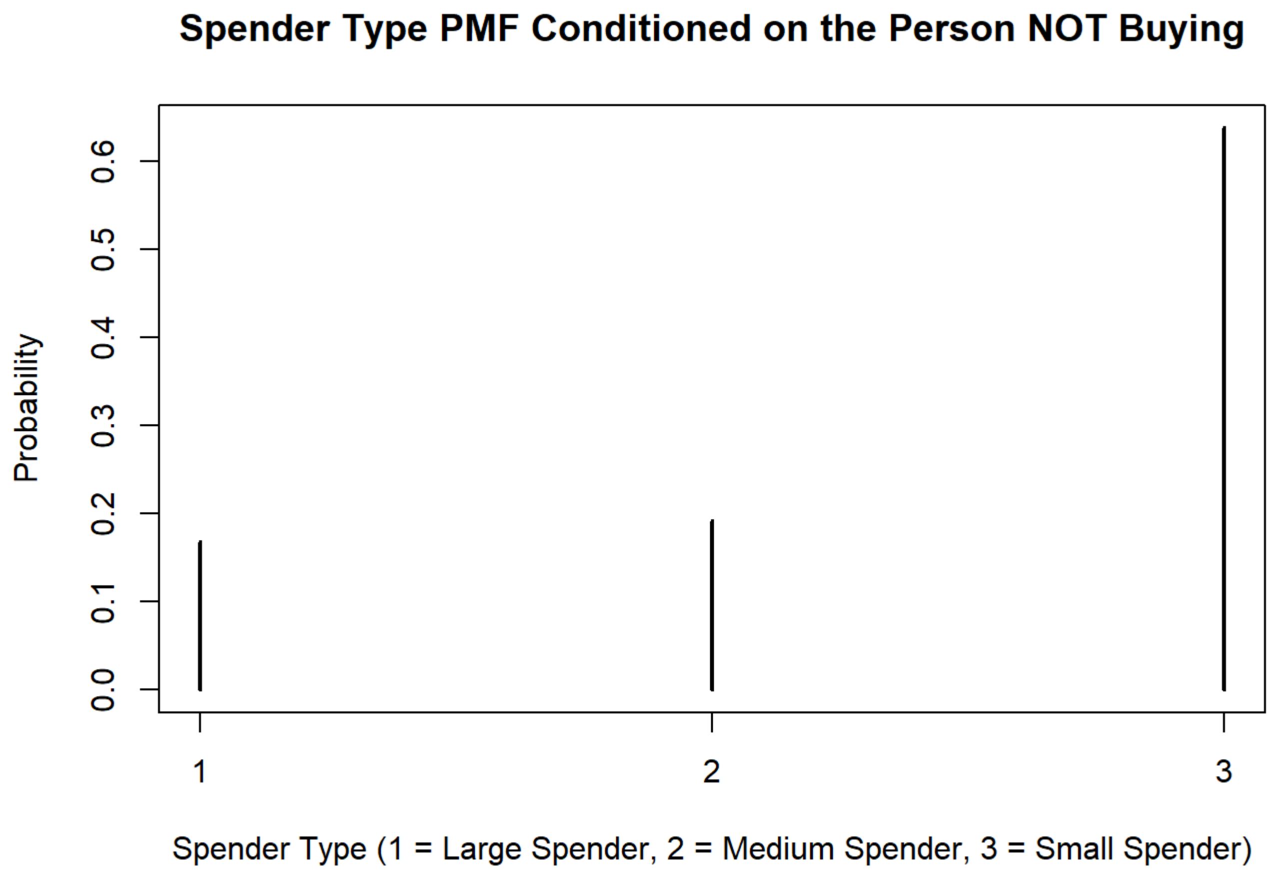
```
print("The proportion table for age type conditioned on bought is shown below")
```

```
## [1] "The proportion table for age type conditioned on bought is shown below"
```

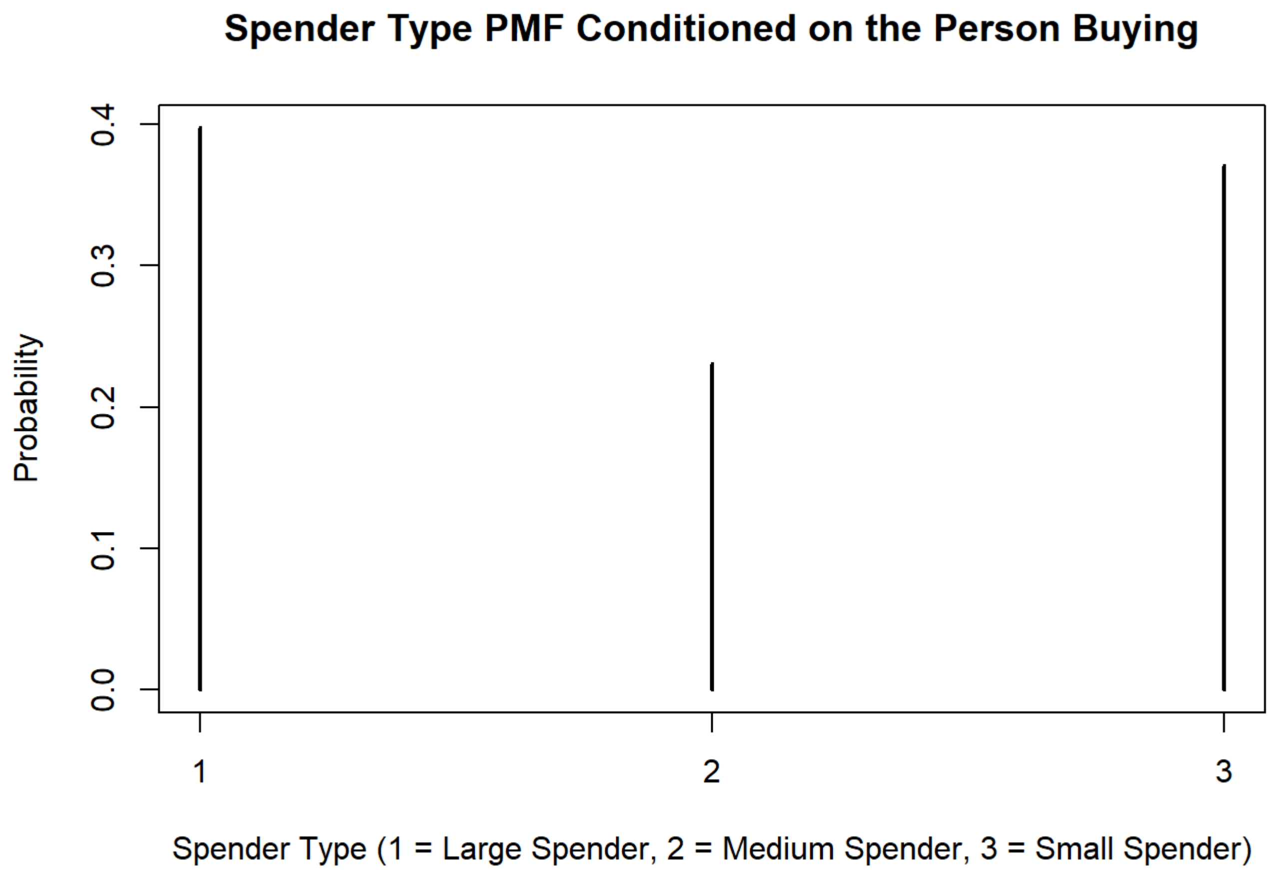
```
B1condAge
```

```
##
##          15          16          17          18          19          20
## 0.002923977 0.029239766 0.011695906 0.014619883 0.023391813 0.011695906
##          21          22          23          24          26          27
## 0.005847953 0.002923977 0.005847953 0.008771930 0.002923977 0.002923977
##          28          29          30          31          32          33
## 0.005847953 0.011695906 0.011695906 0.020467836 0.017543860 0.035087719
##          34          35          36          37          38          39
## 0.032163743 0.011695906 0.017543860 0.046783626 0.023391813 0.049707602
##          40          41          42          43          44          45
## 0.023391813 0.023391813 0.035087719 0.029239766 0.029239766 0.035087719
##          46          47          48          49          50          51
## 0.023391813 0.029239766 0.029239766 0.020467836 0.032163743 0.032163743
##          52          53          54          55          56          57
## 0.005847953 0.017543860 0.017543860 0.017543860 0.008771930 0.023391813
##          58          59          60          61          62          63
## 0.005847953 0.011695906 0.017543860 0.005847953 0.005847953 0.020467836
##          64          65          66          67          68          69
## 0.011695906 0.014619883 0.005847953 0.005847953 0.002923977 0.008771930
##          70          71          73          75          77          78
## 0.005847953 0.005847953 0.011695906 0.005847953 0.005847953 0.005847953
##          95
## 0.002923977
```

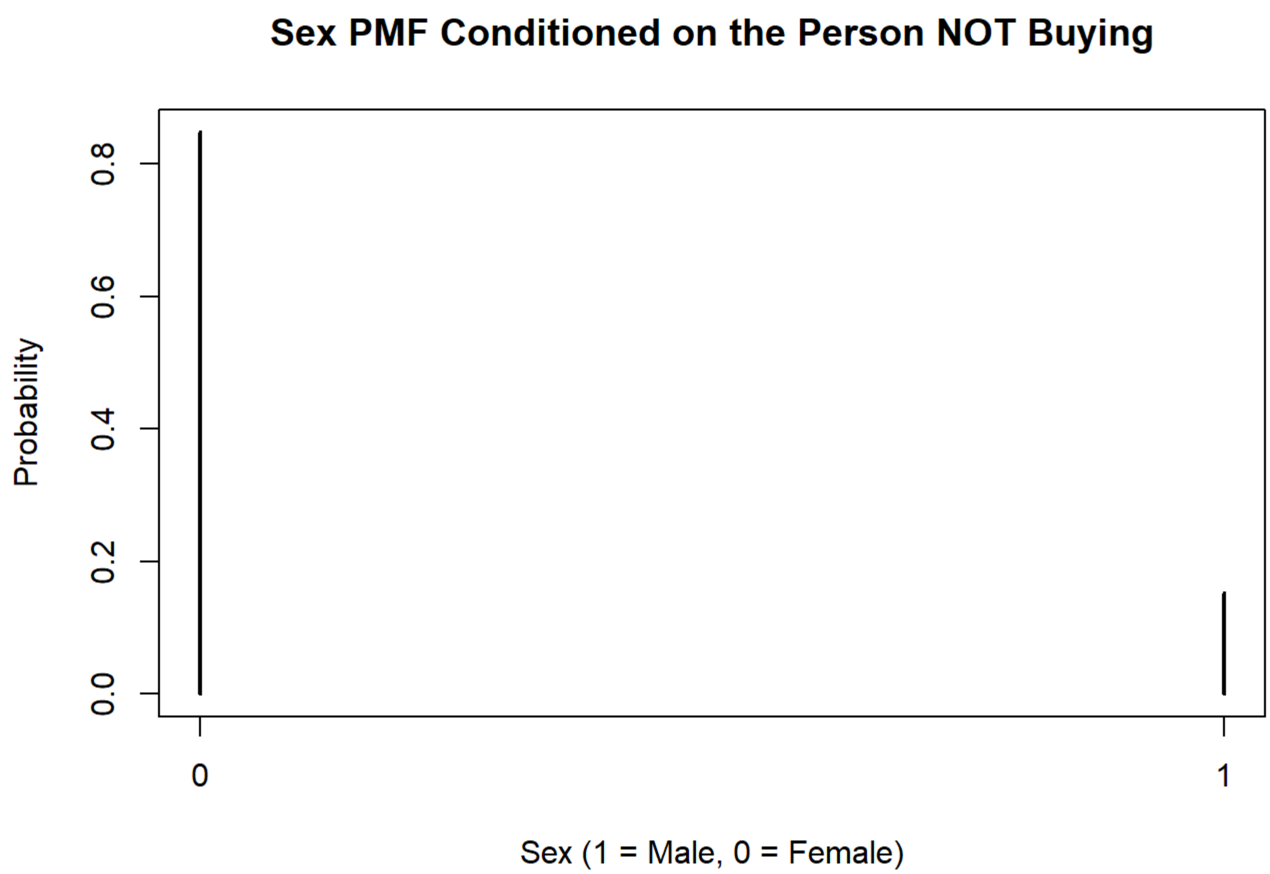
```
plot(B0condSpenderType, main = "Spender Type PMF Conditioned on the Person NOT Buying", ylab= "P
robability",xlab="Spender Type (1 = Large Spender, 2 = Medium Spender, 3 = Small Spender)")
```



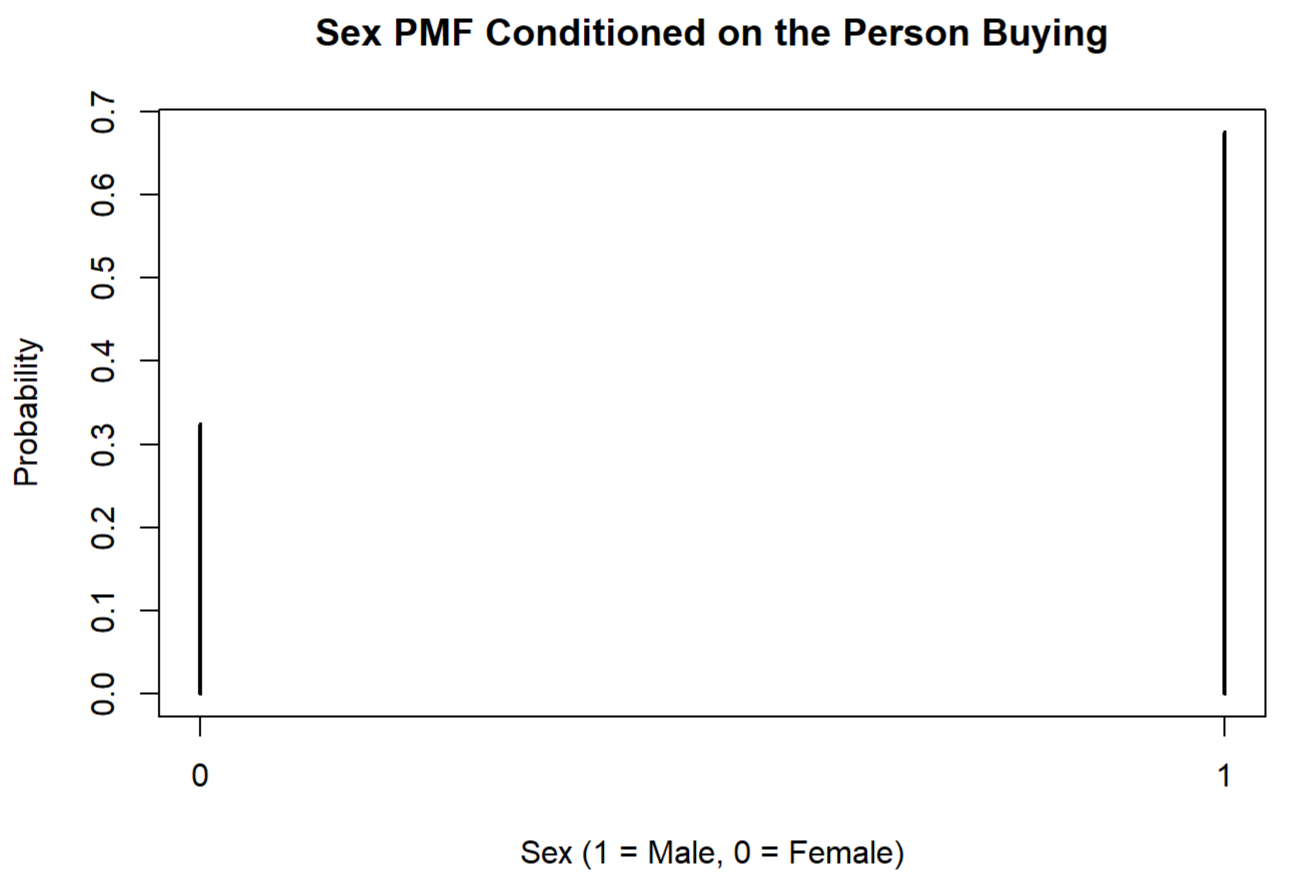

```
plot(B1condSpenderType, main = "Spender Type PMF Conditioned on the Person Buying", ylab= "Proba
bility",xlab="Spender Type (1 = Large Spender, 2 = Medium Spender, 3 = Small Spender)")
```



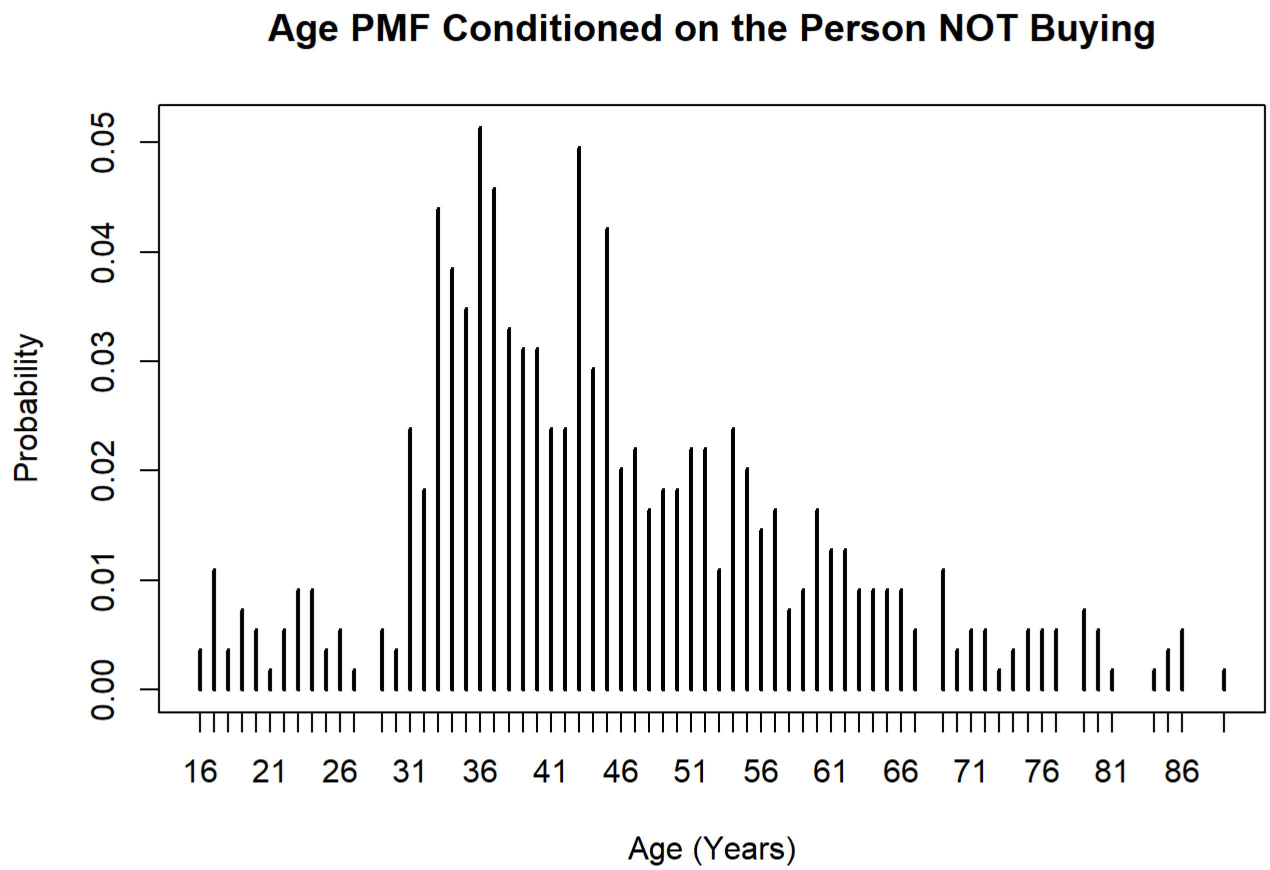

```
plot(B0condSex, main = "Sex PMF Conditioned on the Person NOT Buying", ylab= "Probability",xlab
="Sex (1 = Male, 0 = Female)")
```



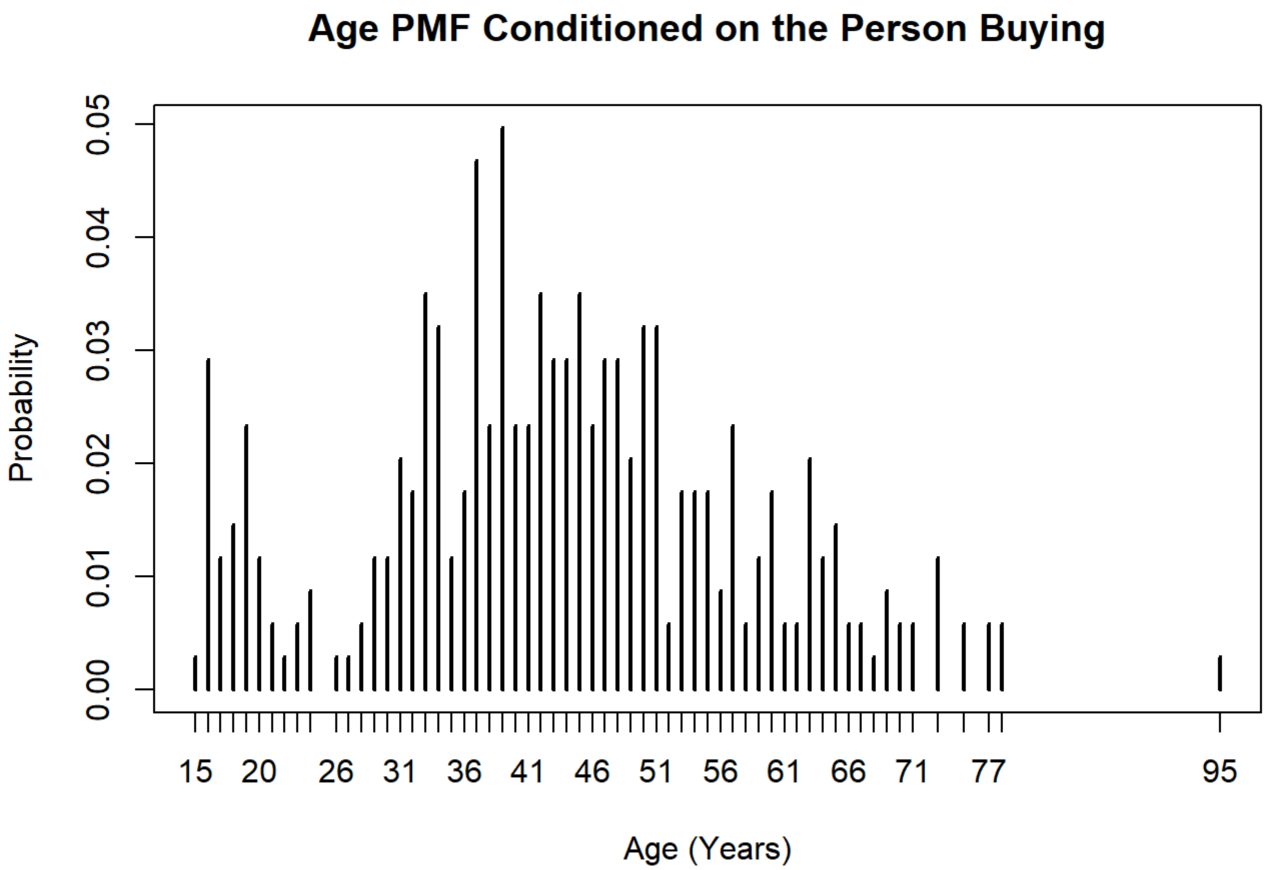

```
plot(B1condSex, main = "Sex PMF Conditioned on the Person Buying", ylab= "Probability",xlab="Sex (1 = Male, 0 = Female)")
```




```
plot(B0condAge, main = "Age PMF Conditioned on the Person NOT Buying", ylab= "Probability",xlab
="Age (Years)")
```




```
plot(B1condAge, main = "Age PMF Conditioned on the Person Buying", ylab= "Probability",xlab="Age
(Years)")
```



c

```
#From tables above we can paste in the values
pB0 <-0.6144307 # proportion of users that did not buy
pT1B0 <-0.1688073 # proportion of users that are type 1 spenders conditioned on them not buying
pS0B0 <-0.8477064 # proportion of users that are female conditioned on them not buying


Age55 <- userData[userData$Age <=55,]
B0Age55 <- Age55[Age55$Bought == 0,]
tally(B0Age55)
```

```
##     n
## 1 432
```

```
tally(B0)
```

```
##     n
## 1 545
```

```
pB0A55 <- 432/545 # proportion of users that are below 55

pIntersectB0 = pB0*pT1B0*pS0B0*pB0A55
print(paste("P(B = 0, T = 1, S = 0, A ≤ 55) = ", pIntersectB0) )
```

```
## [1] "P(B = 0, T = 1, S = 0, A ≤ 55) =  0.0696942320745183"
```

```
pB1 <-0.3855693# proportion of users that bought
pT1B1 <- 0.3976608# proportion of users that are type 1 spenders conditioned on them buying
pS0B1 <- 0.3245614# proportion of users that are female conditioned on them buying
  Age55 <- userData[userData$Age <=55,]
    B1Age55 <- Age55[Age55$Bought == 1,]
  tally(B1Age55)
```

```
##     n
## 1 276
```

```
  tally(B1)
```

```
##     n
## 1 342
```

```
pB1A55 <- 276/342 # proportion of users that are below 55 conditioned on them buying

pIntersectB1 = pB1*pT1B1*pS0B1*pB1A55
print(paste("P(B = 1, T = 1, S = 0, A ≤ 55) = ", pIntersectB1) )
```

```
## [1] "P(B = 1, T = 1, S = 0, A ≤ 55) =  0.0401601265728286"
```

d

```
pT1<-0.2570462
pS0<-0.6459977

Age55 <- userData[userData$Age <=55,]
tally(Age55)
```

```
##     n
## 1 708
```

```
tally(userData)
```

```
##     n
## 1 887
```

4b $\boxed{\text{mean } X_i: \overset{[3,7]}{\frac{7+3}{2}} = 5}$ Uniform RV mean: $E[X] = \frac{a+b}{2}$

$$\boxed{VAR(X_i) = \frac{7^3 - 3^3}{3(7) - 3(3)} - 5^2 = \frac{4}{3}}$$

$[a,b]$

$$VAR(X) = \frac{b^3 - a^3}{3b - 3b} - \left(\frac{a+b}{2}\right)^2$$

$$Zn: \frac{\sum_{i=1}^{n} x_i}{n}$$

$E[Zn] = \frac{1}{n}\left[E[X_1 + X_2 ... + X_n]\right]$ through linearity of expectations and i.i.d.

$$\boxed{E[Z_n] = 5}$$

$VAR(Zn) = \frac{1}{n^2}\sum_{i=1}^{n} VAR(X_i) \rightarrow$ i.i.d. $\frac{1}{n} \cdot VAR(X) = \frac{4}{3(n)}$

independent

$$\boxed{VAR(Z_n) = \frac{4}{3n}}$$

4d $E[X] = \sum_{i}^{10} P(X_i) \cdot X$

$$\left(\frac{2}{3} \cdot \frac{1}{5}\right) 1$$

$$\frac{2 \cdot (1+3+5+7+9) + (2,4,6,8,10)}{15}$$

$$\frac{50+20}{15} = \boxed{\frac{16}{3}}$$

$VAR(X) = E[X^2] - m_X^2$

$E[X^2] = \sum P(X_i) \cdot X^2$

$$\frac{2(1+9+25+49+81) + (4+16+36+64+100)}{15} = \frac{110}{3}$$

$$\boxed{\frac{110}{3} - \frac{256}{9} = \frac{74}{9} = VAR(X)}$$

$E[Z_n] = E[X_1 + X_2 + X ...] = \frac{nE[X_1]}{n}$ ← because i.i.d. $= \boxed{E[X_1] = E[Z_n] = \frac{16}{3}}$

$$Z_n = \frac{X_1 + X_2 + \cdots X_n}{n}$$

$VAR(Z_n) = \frac{1}{n^2}\sum_{i=1}^{n} VAR(X_i)$

since i.i.d $\boxed{\frac{1}{n} \cdot VAR(X_i) = \frac{74}{9n}}$ → Therefore $\boxed{\sigma = \sqrt{\frac{74}{9n}} \text{ or } \sqrt{\frac{8.22}{n}}}$

```
pA55<-708/887


B0conditioned <-pIntersectB0/(pT1*pS0*pA55)
print(paste("P(B = 0|T = 1, S = 0, A ≤ 55) = ", B0conditioned) )
```

```
## [1] "P(B = 0|T = 1, S = 0, A ≤ 55) =  0.525829611971687"
```

```
B1conditioned <-pIntersectB1/(pT1*pS0*pA55)
print(paste("P(B = 1|T = 1, S = 0, A ≤ 55) = ", B1conditioned) )
```

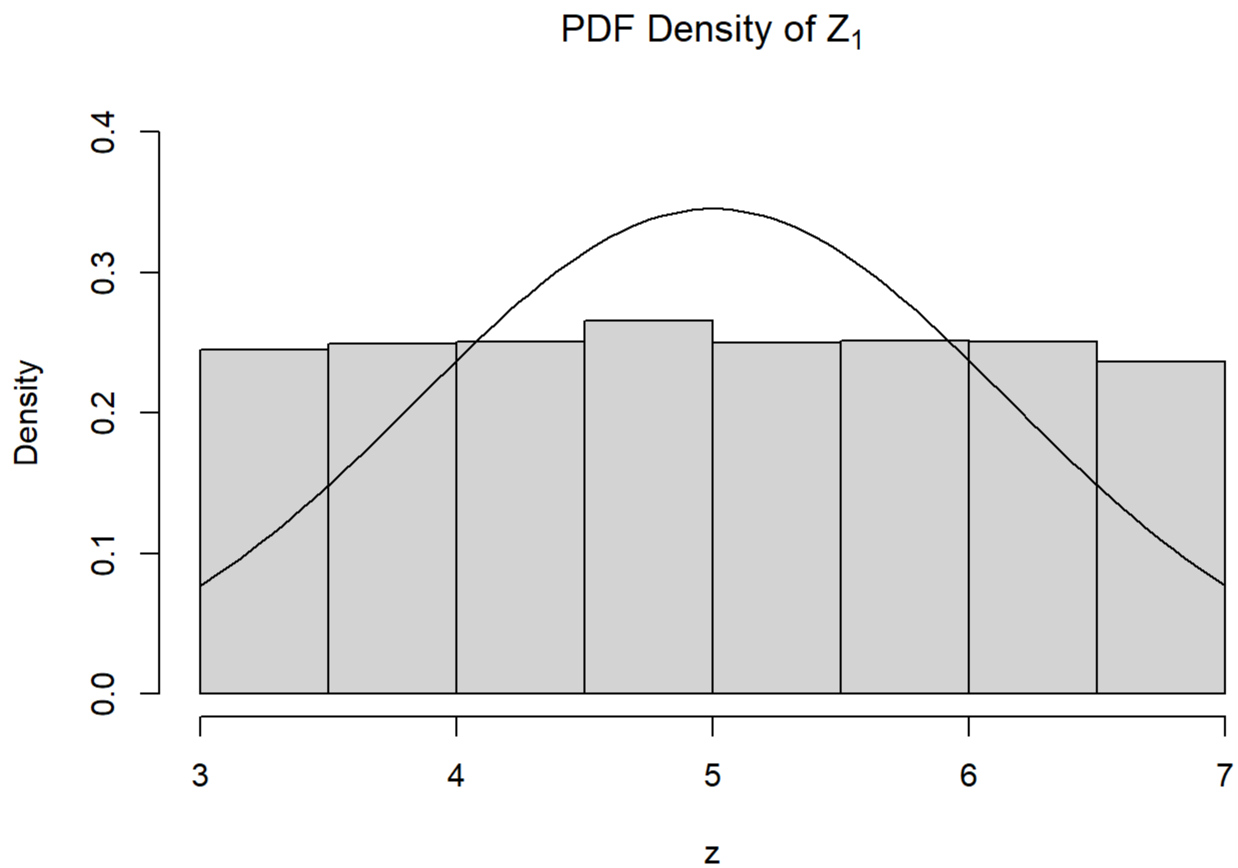
```
## [1] "P(B = 1|T = 1, S = 0, A ≤ 55) =  0.303000451313463"
```

Based on the probabilities listed above we know that the probability of a female, whos age is below 55 and who is a large spender will buy this product is about to equal to 0.303 while the probability that she will not buy the product is 0.523. Therefore we can predict that she will not buy this product.

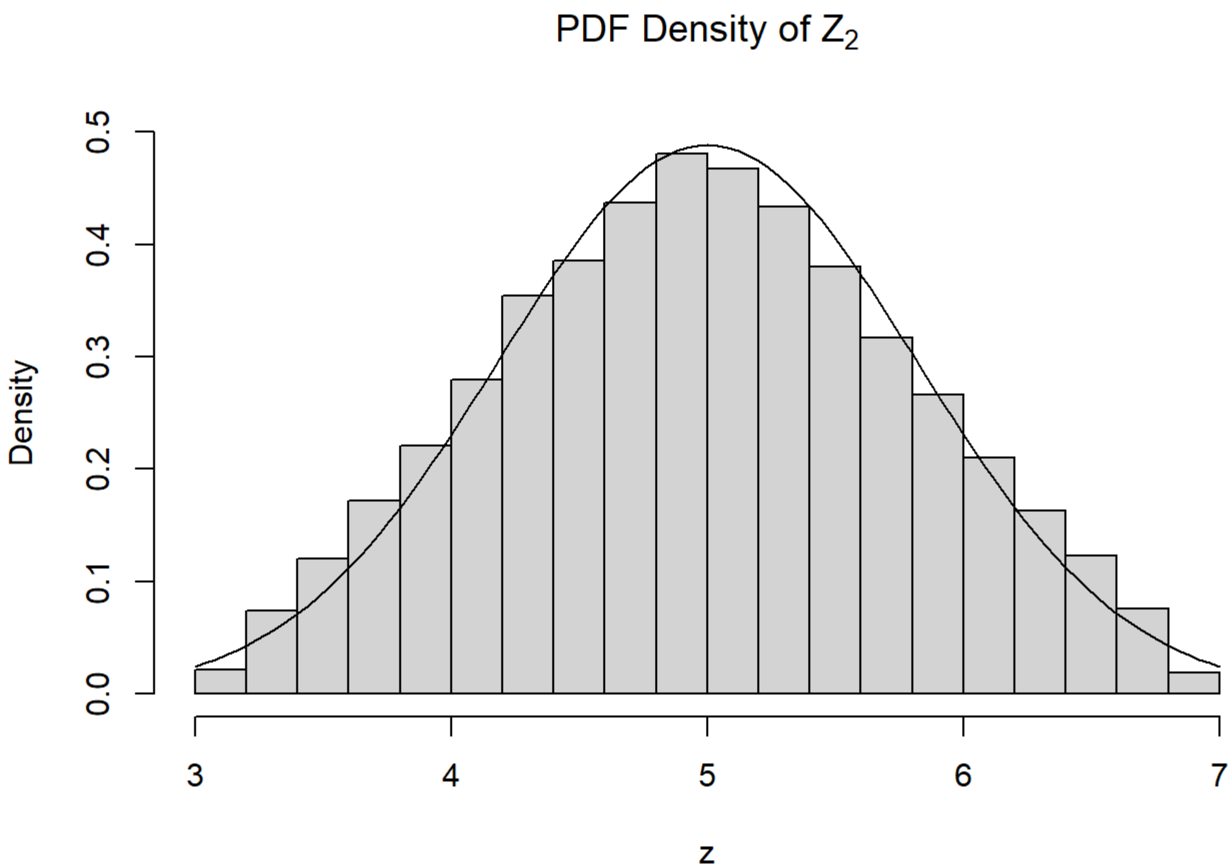
Question 4

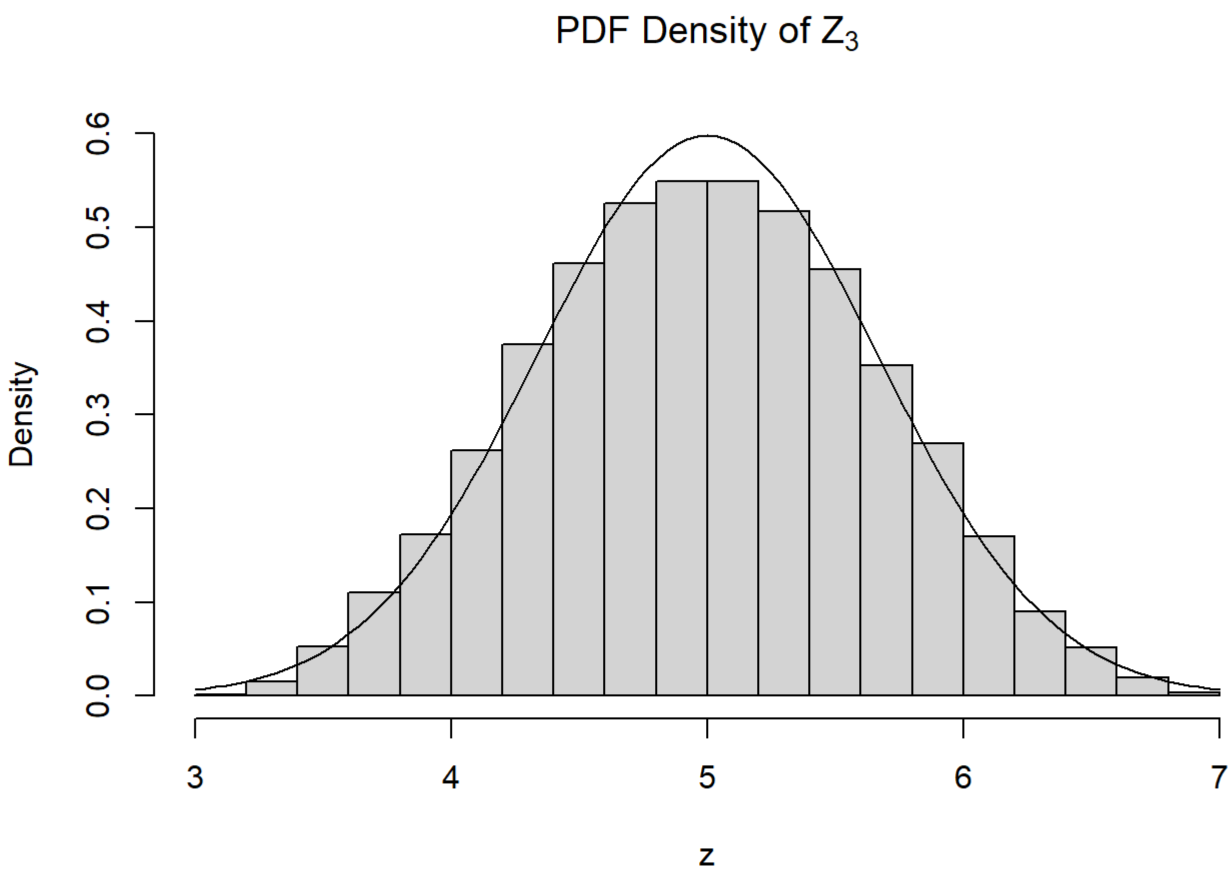
a and c

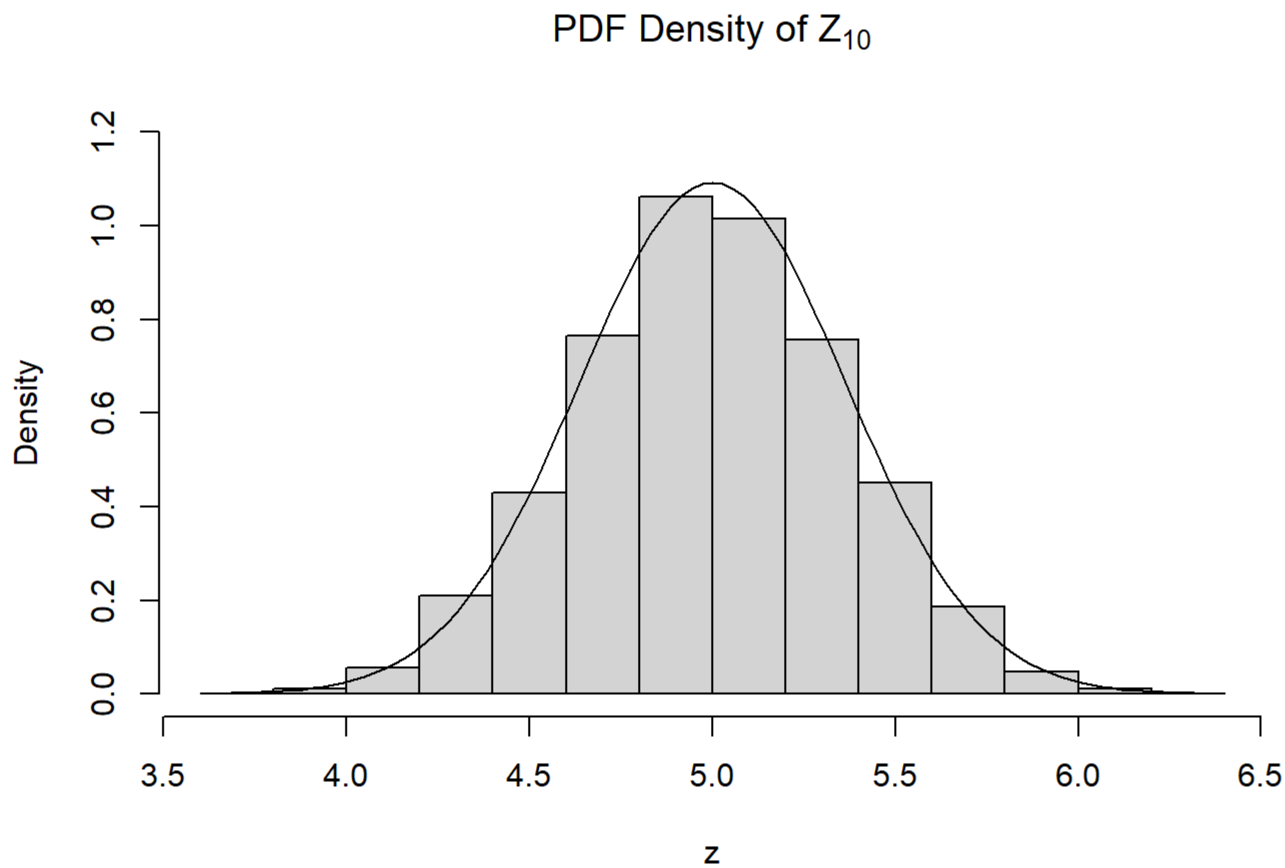
```
set.seed(123)
samples = 10000
x.samples <- replicate(100,runif(samples,min=3,max=7)) # creates the uniform samples of x, 100 o
f them


sn <-c(2,3,10,30,100) # this is for the value of n
sn1 <-c("2","3","10","30","100") # this is for naming the graphs

mu <- 5 # E[X]

Zn1<- x.samples[,1] # n=1
hist(Zn1,xlab="z", ylab="Density" ,main=substitute(paste('PDF Density of ',Z[1])), prob=TRUE, yl
im =c(0,0.4),breaks = seq(from=3, to=7, by=1/2) )
curve(dnorm(x, mu,sqrt(4/3)), add = TRUE)
```

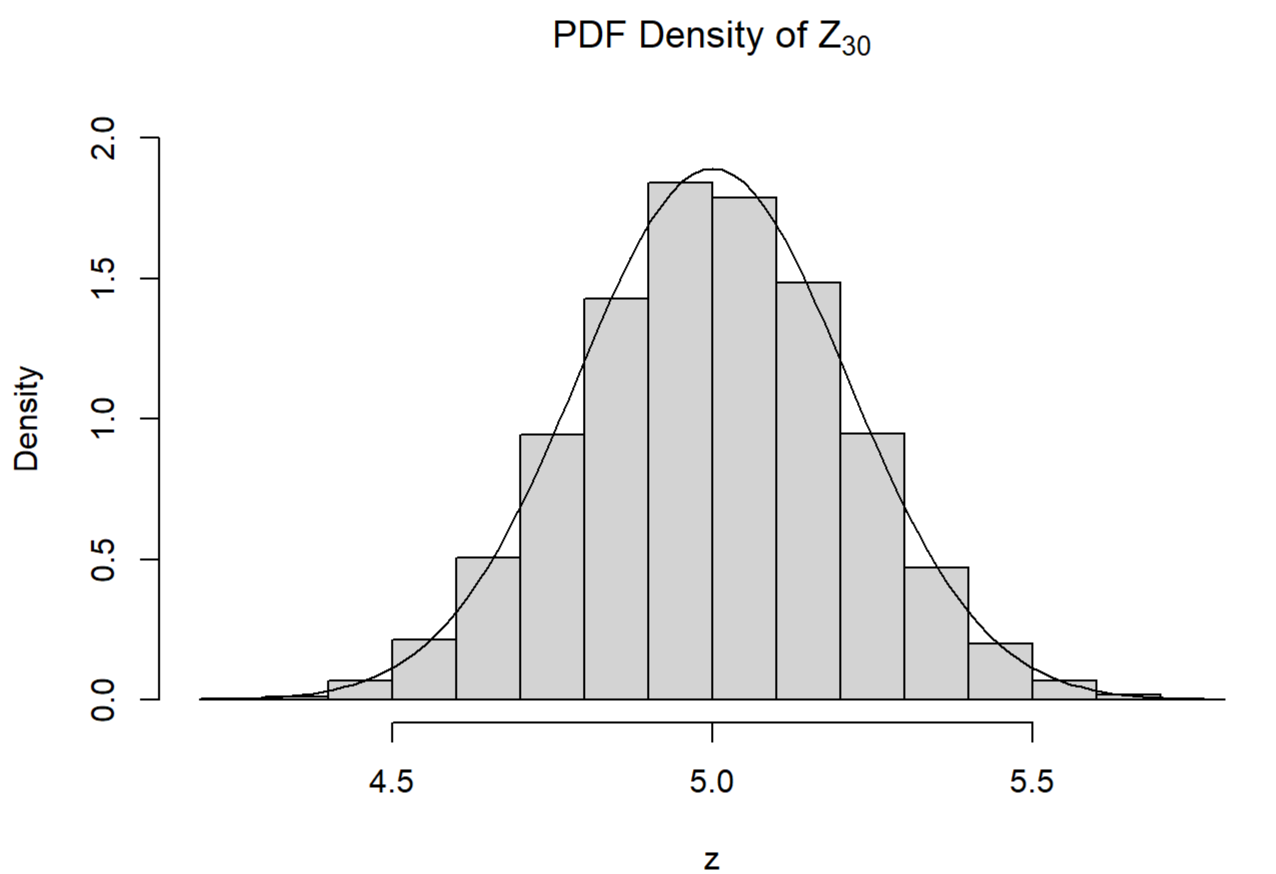

PDF Density of Z_1
Density
0.0
0.1
0.2
0.3
0.4
3
4
5
6
7
z

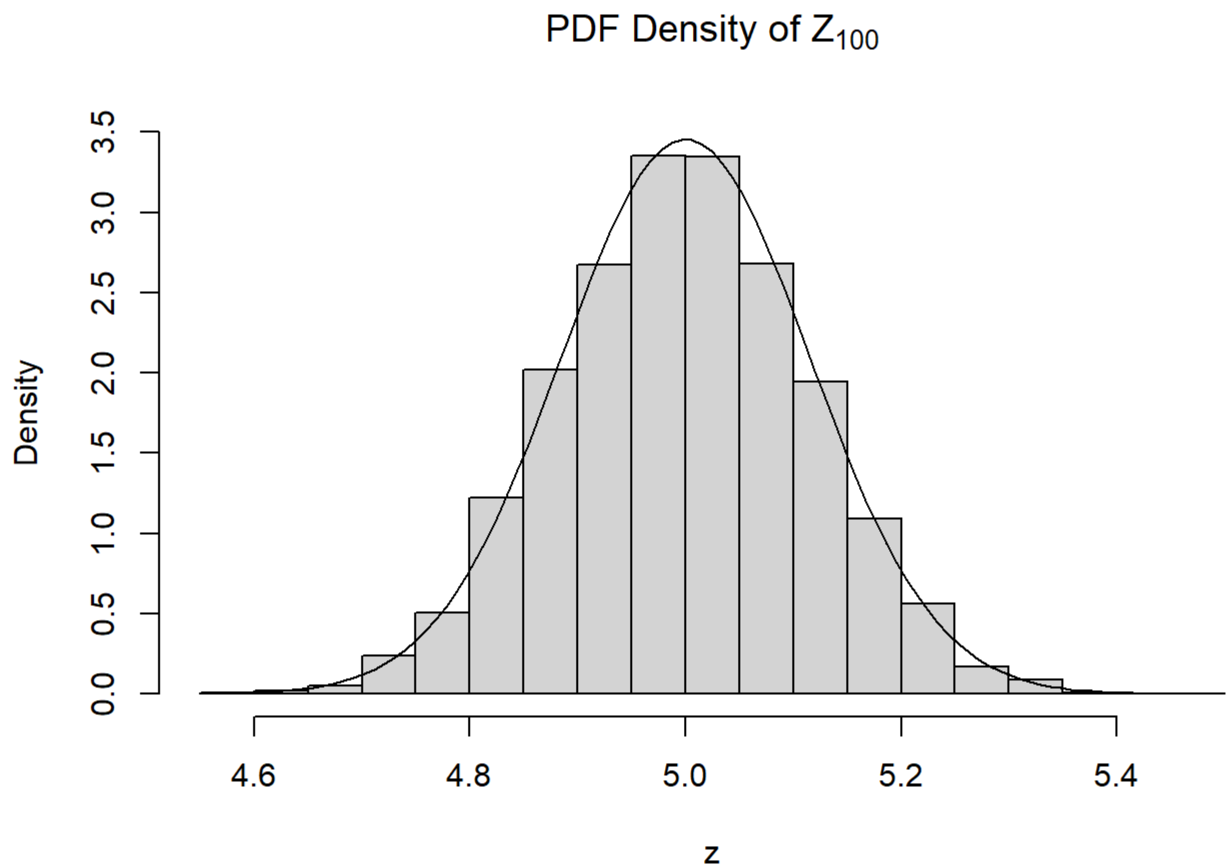
```
#Loop that counts up the rest of the conditions of n and names them appropriately
for(i in sn){
  Zn<-rowSums(x.samples[,1:i])/i # adds up the rows
  #determines what graph it will get
  if(i==100){
    hist(Zn,xlab="z", ylab="Density"  ,main=substitute(paste('PDF Density of ',Z[100])), prob=TR
UE, ylim = c(0,3.5))
    curve(dnorm(x, mu,sqrt(4/(3*100))), add = TRUE)
  }
  else if (i==30){
    hist(Zn,xlab="z", ylab="Density" , main=substitute(paste('PDF Density of ',Z[30])), prob=TRU
E, ylim = c(0,2) )
    curve(dnorm(x, mu,sqrt(4/(3*30))), add = TRUE)
  }
   else if (i==10){
    hist(Zn,xlab="z", ylab="Density" , main=substitute(paste('PDF Density of ',Z[10])), prob=TRU
E, ylim = c(0,1.2) )
    curve(dnorm(x, mu,sqrt(4/(3*10))), add = TRUE)
   }
   else if (i==3){
    hist(Zn,xlab="z", ylab="Density" , main=substitute(paste('PDF Density of ',Z[3])), prob=TRU
E, ylim = c(0,0.6) )
    curve(dnorm(x, mu,sqrt(4/(3*3))), add = TRUE)
   }
   else if (i==2){
    hist(Zn,xlab="z", ylab="Density" , main=substitute(paste('PDF Density of ',Z[2])), prob=TRU
E, ylim = c(0,0.5) )
    curve(dnorm(x, mu,sqrt(4/(3*2))), add = TRUE)
  }
}
```


PDF Density of Z_2
Density
0.0
0.1
0.2
0.3
0.4
0.5
3
4
5
6
7
z


PDF Density of Z_3
Density
0.0
0.1
0.2
0.3
0.4
0.5
0.6
3
4
5
6
7
z

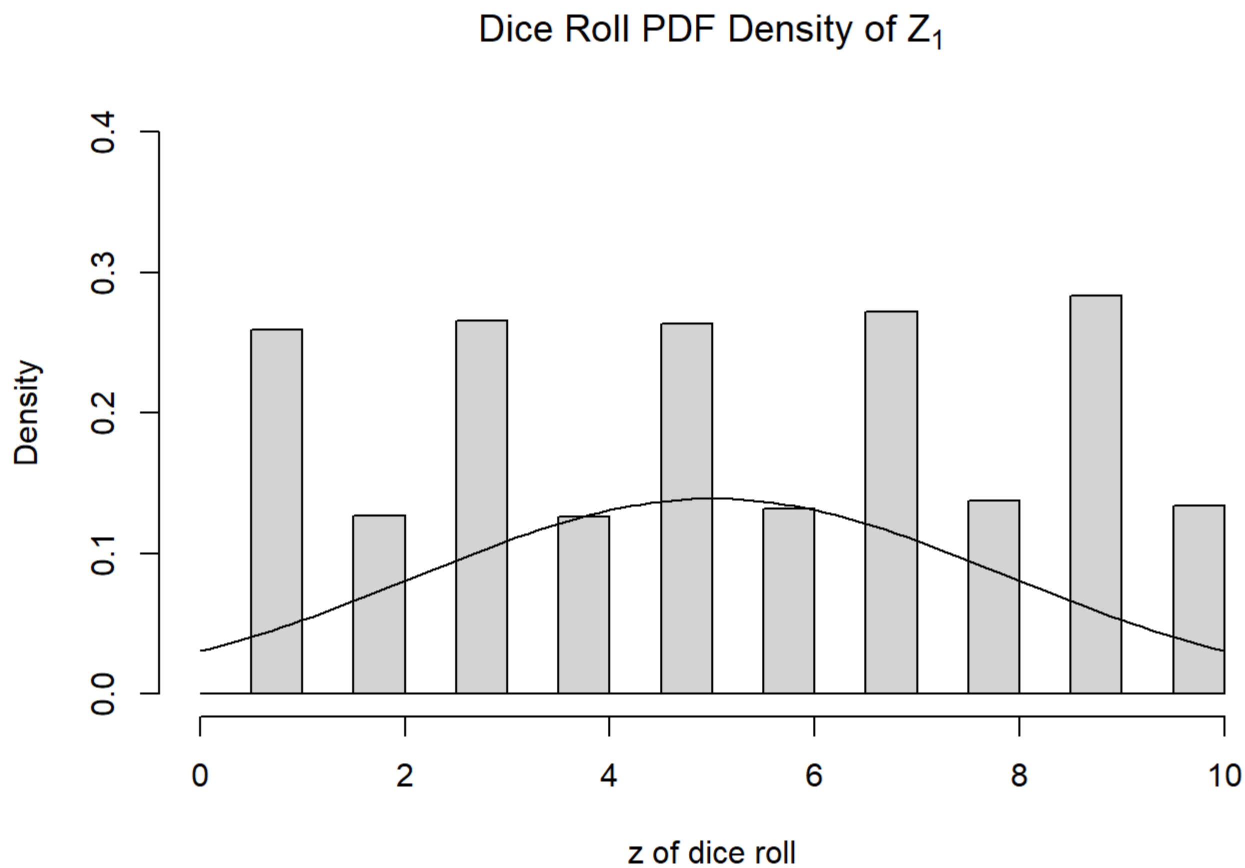

PDF Density of Z_{10}
Density
0.0
0.2
0.4
0.6
0.8
1.0
1.2
3.5
4.0
4.5
5.0
5.5
6.0
6.5
z


PDF Density of Z_{30}
Density
0.0
0.5
1.0
1.5
2.0
4.5
5.0
5.5
z

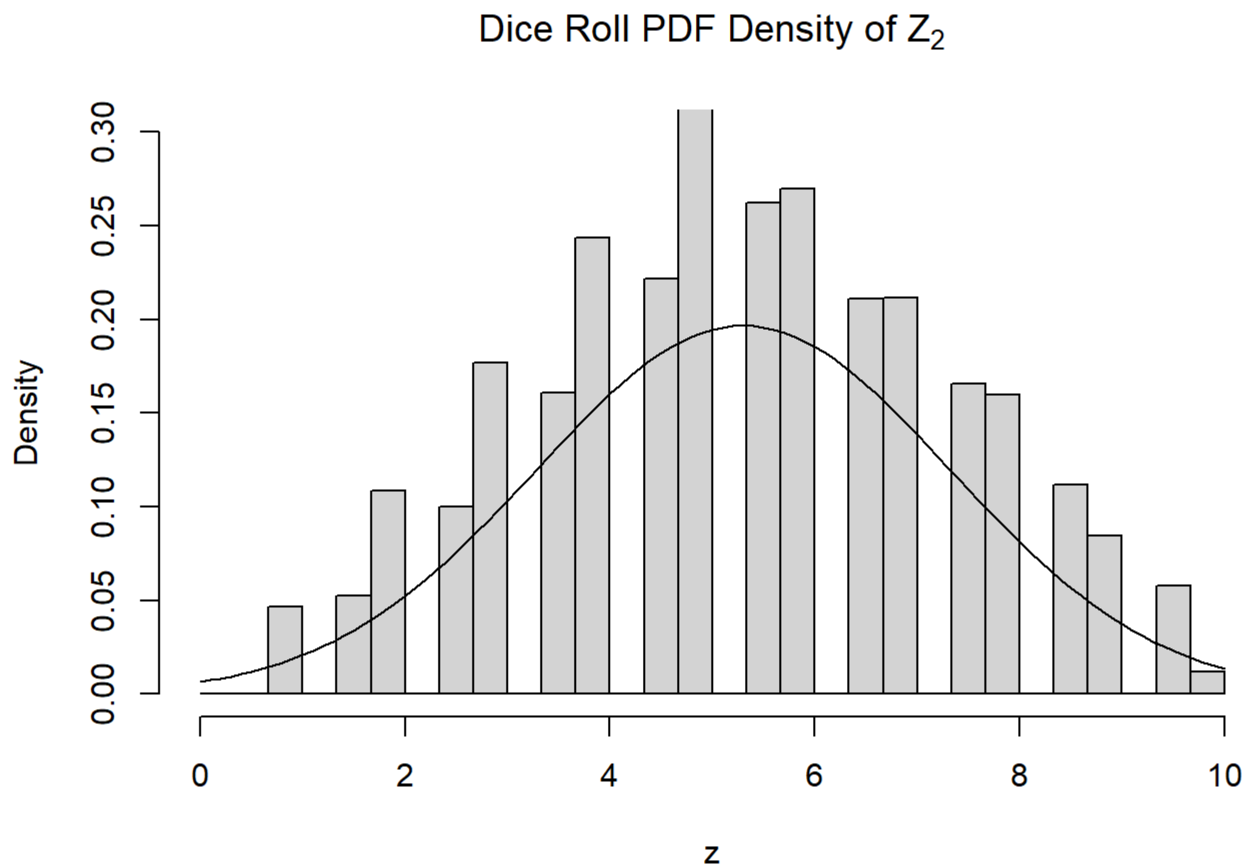


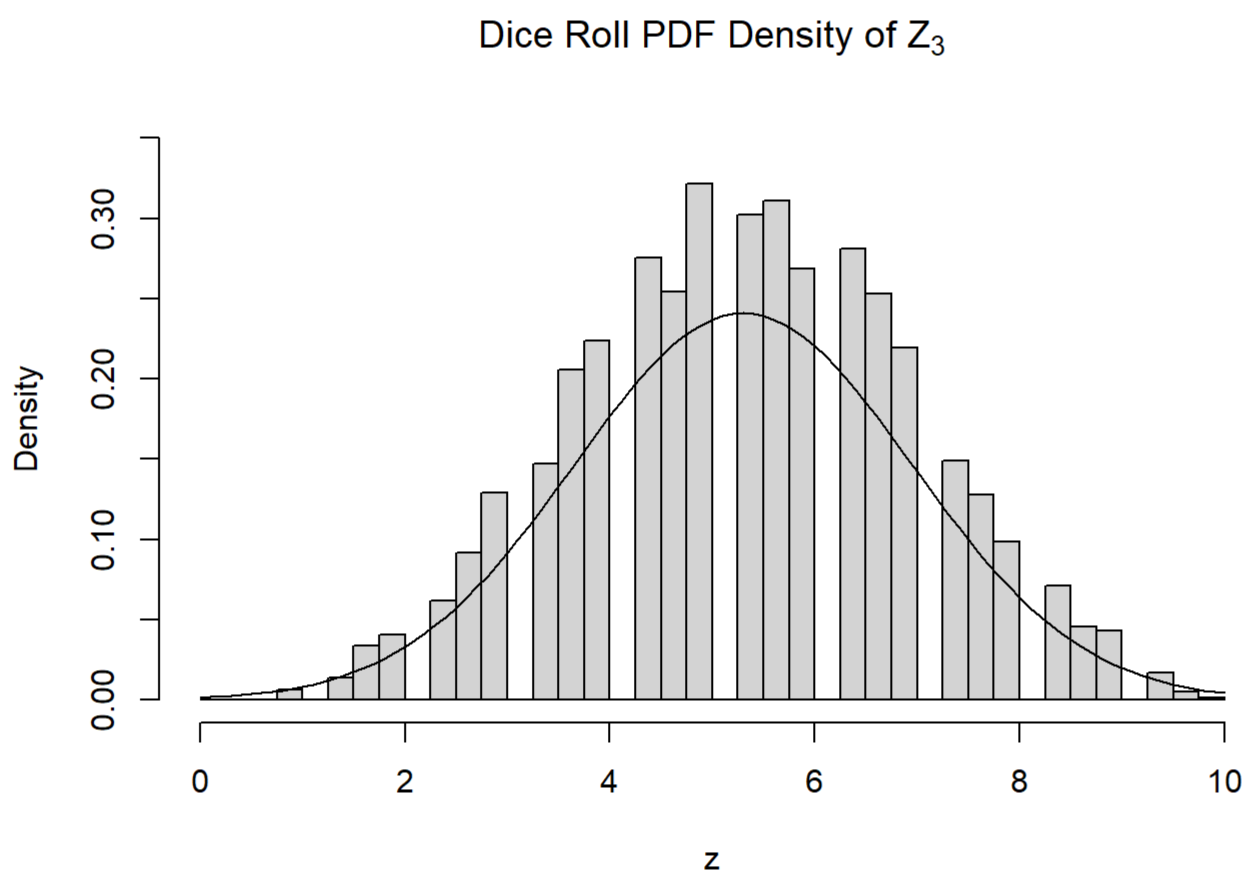
d

```
set.seed(123)
double <-c(2,1,2,1,2,1,2,1,2,1) #vector for weights of 1-10

#generating sample rolls with probability of double for odd numbers
x.dice <- replicate(100,replicate(10000, sample(1:10, 1, prob = double)))

sn <-c(2,3,10,30,100)
sn1 <-c("2","3","10","30","100")

muDice<-5.3

#figure out what binwidth is

Zn1Dice<- x.dice[,1]# macking the first graph
hist(Zn1Dice,xlab="z of dice roll", ylab="Density" ,main=substitute(paste('Dice Roll PDF Density
of ',Z[1])), prob=TRUE, ylim =c(0,0.4),breaks = seq(from=0, to=10, by=1/2))
curve(dnorm(x, mu,sqrt(74/9)), add = TRUE)
```

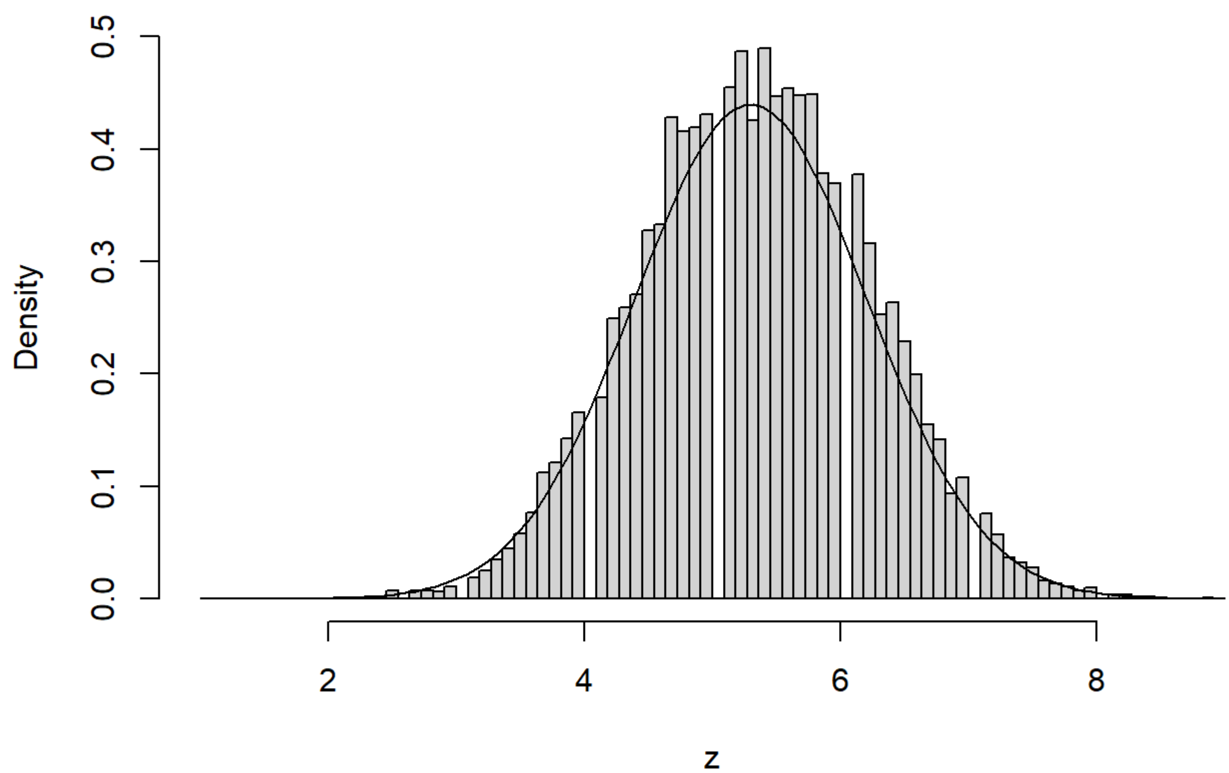

Dice Roll PDF Density of Z_1
Density
0.0
0.1
0.2
0.3
0.4
0
2
4
6
8
10
z of dice roll

```
for(i in sn){
  ZnDice<-rowSums(x.dice[,1:i])/i# adds up the rows
  #determines what graph it gets
  if(i==100){
    hist(ZnDice,xlab="z", ylab="Density"  ,main=substitute(paste('Dice Roll PDF Density of ',Z[1
00])), prob=TRUE, ylim = c(0,1.5),breaks = seq(from=4, to=7, by=1/101) )
    curve(dnorm(x, muDice,sqrt(74/(9*100))), add = TRUE)
  }
  else if (i==30){
    hist(ZnDice,xlab="z", ylab="Density" , main=substitute(paste('Dice RollPDF Density of ',Z[3
0])), prob=TRUE, ylim = c(0,1),breaks = seq(from=2, to=8, by=1/31) )
    curve(dnorm(x, muDice,sqrt(74/(9*30))), add = TRUE)
  }
   else if (i==10){
    hist(ZnDice,xlab="z", ylab="Density" , main=substitute(paste('Dice Roll PDF Density of ',Z[1
0])), prob=TRUE, ylim = c(0,0.5),breaks = seq(from=1, to=9, by=1/11) )
    curve(dnorm(x, muDice,sqrt(74/(9*10))), add = TRUE)
   }
   else if (i==3){
    hist(ZnDice,xlab="z", ylab="Density" , main=substitute(paste('Dice Roll PDF Density of ',Z
[3])), prob=TRUE, ylim = c(0,0.35),breaks = seq(from=0, to=10, by=1/4) )
    curve(dnorm(x, muDice,sqrt(74/(9*3))), add = TRUE)
   }
   else if (i==2){
    hist(ZnDice,xlab="z", ylab="Density" , main=substitute(paste('Dice Roll PDF Density of ',Z
[2])), prob=TRUE, ylim = c(0,0.3),breaks = seq(from=0, to=10, by=1/3) )
    curve(dnorm(x, muDice,sqrt(74/(9*2))), add = TRUE)
  }
}
```

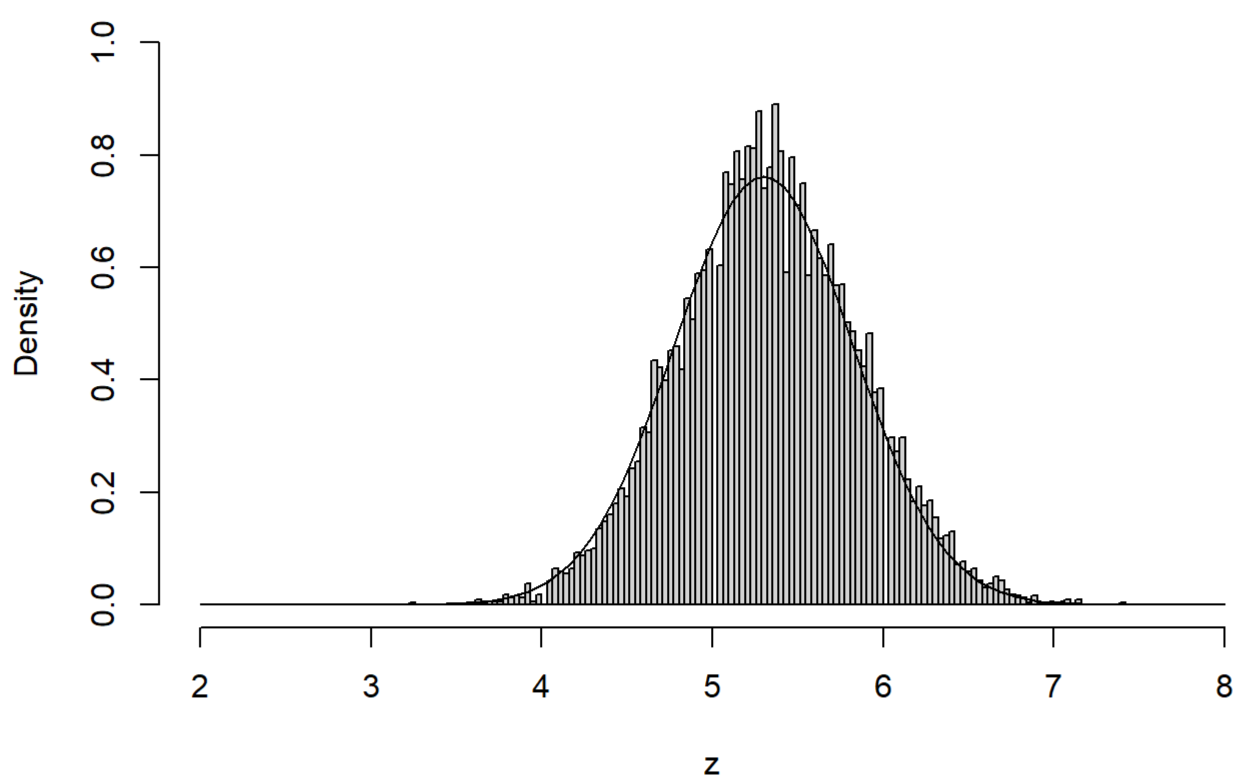

Dice Roll PDF Density of Z_2
Density
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0
2
4
6
8
10
z

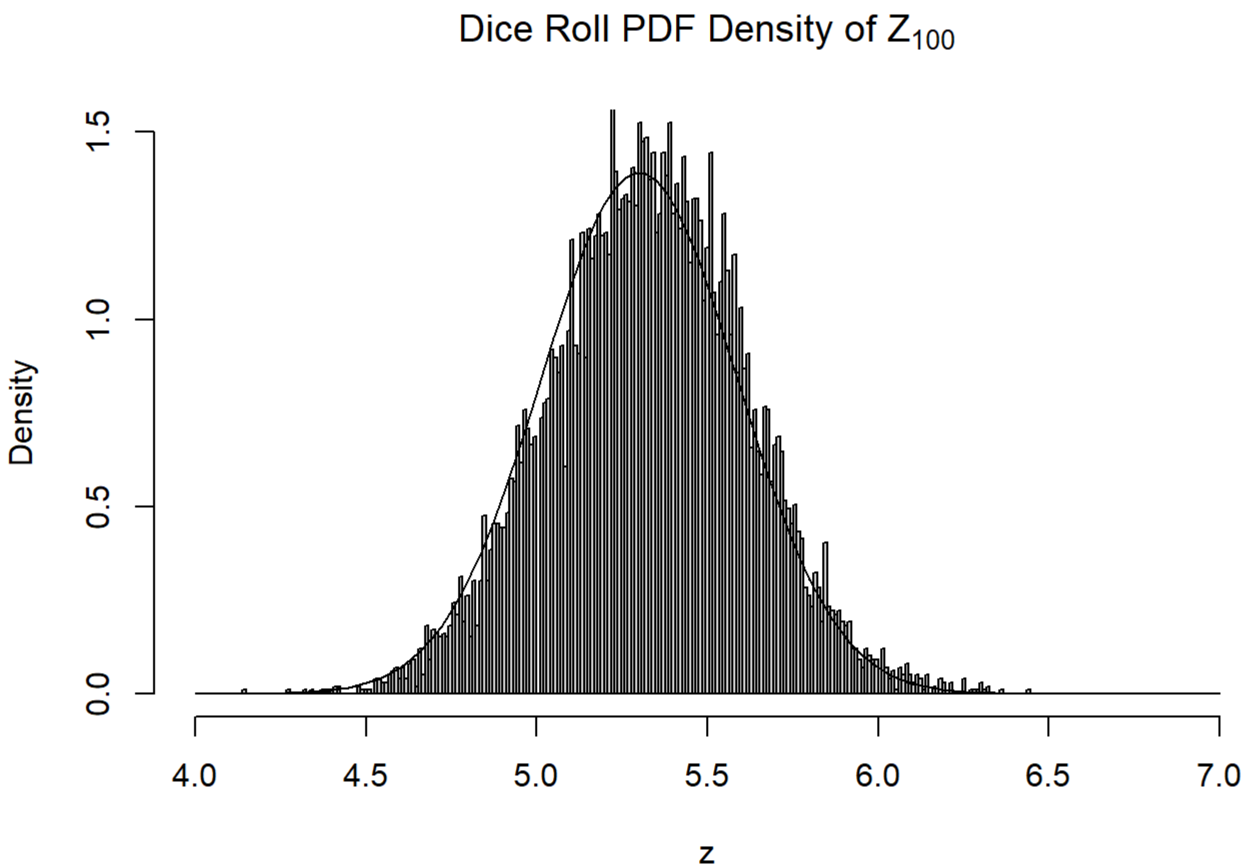

Dice Roll PDF Density of Z_3
Density
0.00
0.10
0.20
0.30
0
2
4
6
8
10
z

Dice Roll PDF Density of Z_{10}



Dice RollPDF Density of Z_{30}





Dice Roll PDF Density of Z_{100}
Density
z
0.0
0.5
1.0
1.5
4.0
4.5
5.0
5.5
6.0
6.5
7.0