Prashant Sharma

Grading:
20 marks- Complete submission of worksheet9
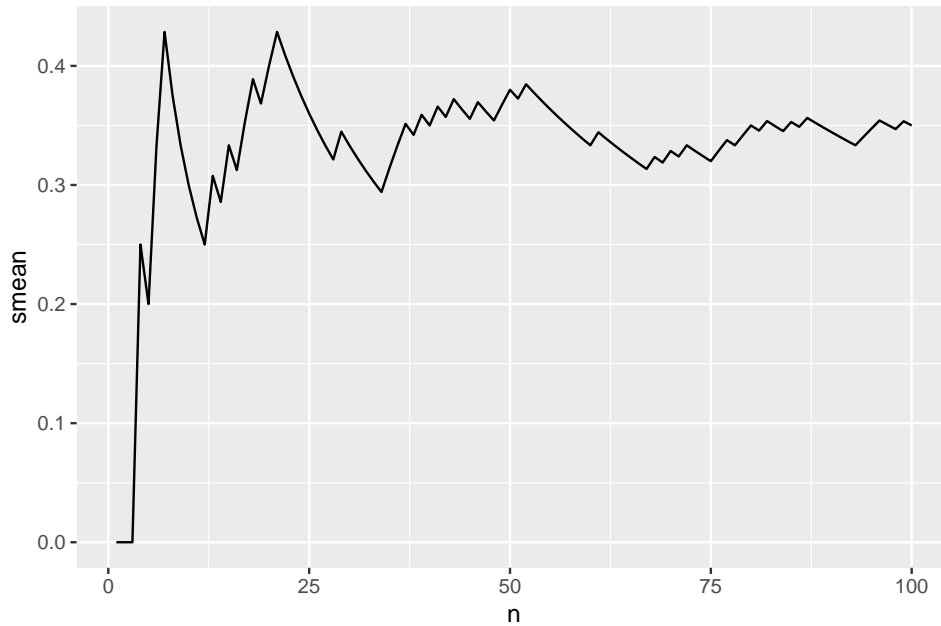40 marks- Problem 1 and 40 marks- Problem 2

Problem:1(a)

```
> #Given, X follows Bernoulli(p) and p = 1/3
> #Simulating for n=100, x1,x2,...,xn iid X
> set.seed(1)
> Xsim = rbinom(100,1,1/3)
> Xsim
```

```
 [1] 0 0 0 1 0 1 1 0 0 0 0 0 1 0 1 0 1 1 0 1 1 0 0 0 0 0 0 0 1 0 0 0 0 0 1 1 1
[38] 0 1 0 1 0 1 0 0 1 0 0 1 1 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 1 0 1 0 0
[75] 0 1 1 0 1 1 0 1 0 0 1 0 1 0 0 0 0 0 0 1 1 1 0 0 1 0
```

Problem:1(b)

Now,Demonstrating the Law of Large numbers by plotting the sample mean as a function of n.

```
> cum = cumsum(Xsim)
> n = 1:100
> smean = cum/n
> library(ggplot2)
> df_smean=data.frame(smean)
> plot_smean=ggplot(df_smean, aes(x=n, y=smean)) +geom_line()
```

As we know that Weak Law of Large Numbers says as the sample size becomes large the sample mean converges to true mean or population mean. Here the true mean is 1/3=0.333. Also, from the plot we can easily observe that as the sample size increases the sample means are close to 0.333. Hence we can demonstrate Weak Law of Large Numbers in this case.

Problem:1(c)

```
> Bernoulli_sim = function (p)
+ {
+ y = rbinom(100,1,p)
+ c = cumsum(y)
+ n = 1:100
+ c/n }
> Bernoulli_sim(1/3)
```

```
 [1] 0.0000000 0.0000000 0.0000000 0.2500000 0.2000000 0.1666667 0.1428571
 [8] 0.1250000 0.2222222 0.2000000 0.2727273 0.3333333 0.3076923 0.2857143
[15] 0.2666667 0.2500000 0.2941176 0.2777778 0.2631579 0.2500000 0.2857143
[22] 0.2727273 0.2608696 0.2500000 0.2800000 0.2692308 0.2592593 0.2500000
[29] 0.2413793 0.2333333 0.2258065 0.2187500 0.2121212 0.2058824 0.2285714
[36] 0.2222222 0.2162162 0.2105263 0.2307692 0.2250000 0.2439024 0.2380952
[43] 0.2325581 0.2272727 0.2444444 0.2391304 0.2340426 0.2500000 0.2448980
[50] 0.2600000 0.2549020 0.2500000 0.2452830 0.2407407 0.2363636 0.2321429
[57] 0.2280702 0.2241379 0.2203390 0.2166667 0.2131148 0.2258065 0.2222222
[64] 0.2343750 0.2461538 0.2424242 0.2388060 0.2352941 0.2463768 0.2428571
[71] 0.2394366 0.2500000 0.2602740 0.2567568 0.2533333 0.2631579 0.2597403
[78] 0.2692308 0.2658228 0.2750000 0.2716049 0.2682927 0.2771084 0.2738095
[85] 0.2823529 0.2790698 0.2873563 0.2954545 0.3033708 0.3000000 0.3076923
[92] 0.3043478 0.3010753 0.3085106 0.3052632 0.3020833 0.2989691 0.3061224
[99] 0.3030303 0.3100000
```
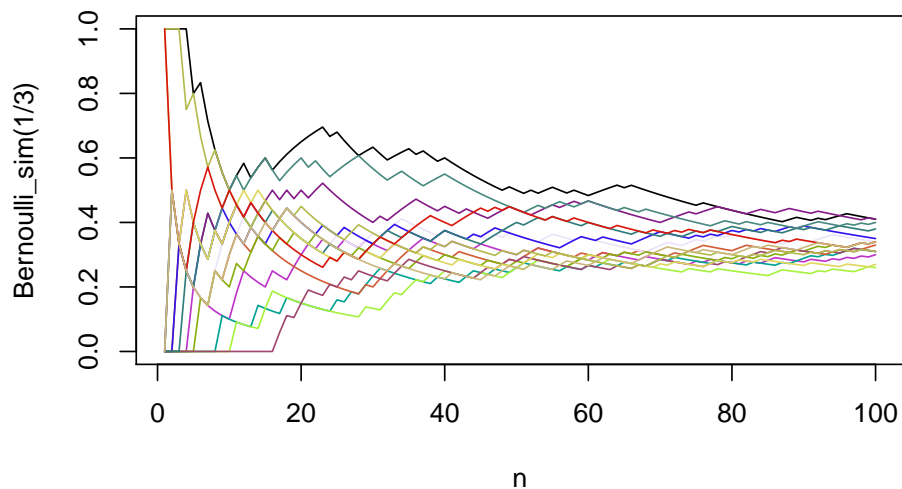
```
> n = 1:100
> plot(n,Bernoulli_sim(1/3) , type = "l",ylim = c(0,1))
> replicate_plot=replicate(15, lines(Bernoulli_sim(1/3)~n, type="l", col=rgb(runif(3),runif(3),runif(3)
```

```
> #library(ggthemes)
> #set.seed(100)
> #mydata <- diamonds[sample(nrow(diamonds), 100), ]
> #ggplot(data = mydata, aes(x = carat, y = price)) + geom_point(shape = 1) +theme_base()
```

```
> plot(n,Bernoulli_sim(1/3) , type = "l",ylim = c(0,1))
> replicate_plot=replicate(15, lines(Bernoulli_sim(1/3)~n, type="l", col=rgb(runif(3),runif(3),runif(3)
```


Thus, We see that as well as the sample sizes become large the sample means of each trial converge to true mean $1/3=0.333$
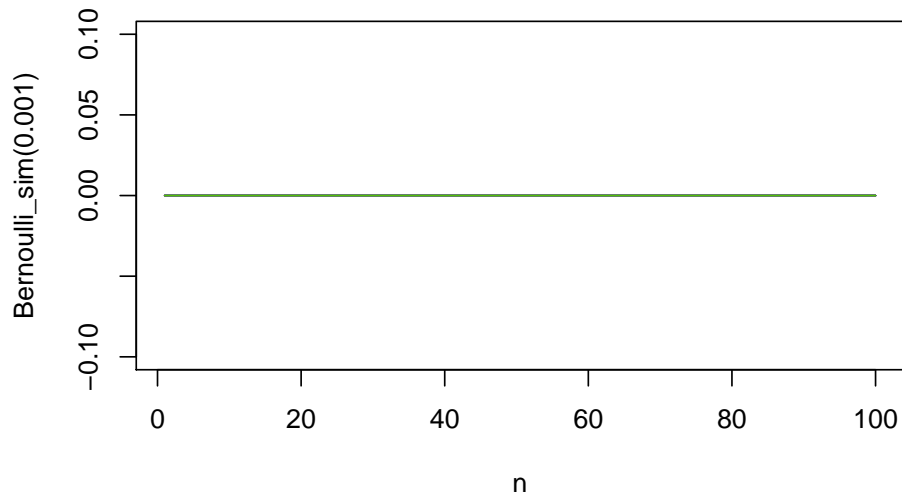
Problem:1(d)

for n=100 and p=0.001

```
> plot(n,Bernoulli_sim(0.001) , type = "l",ylim = c(-0.1,0.1))
> replicate_plot=replicate(15, lines(Bernoulli_sim(0.001)~n, type="l", col=rgb(runif(3),runif(3),runif(3
```
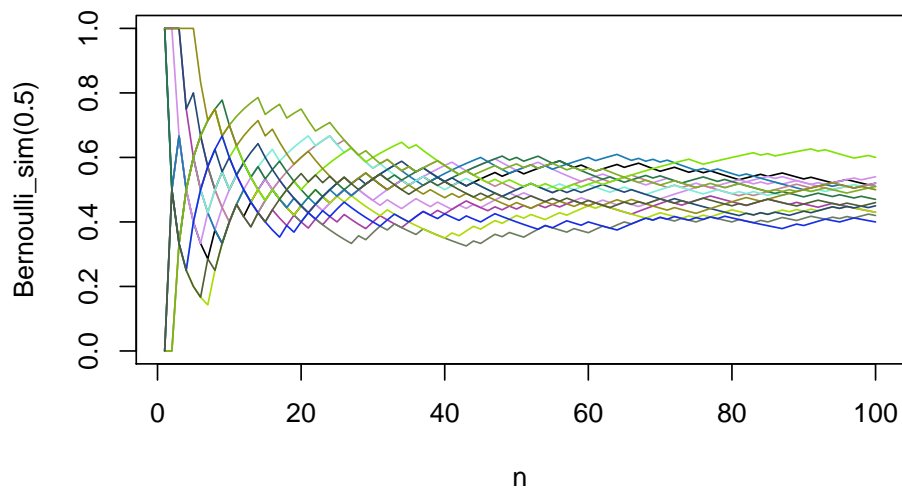
Thus, we can easily see that when p = 0.001, n = 100 the sample mean converges to the true mean of distribution for all 15 replications.

Now,for n=100 and p=0.5

```
> plot(n,Bernoulli_sim(0.5) , type = "l",ylim = c(0,1))
> replicate_plot=replicate(15, lines(Bernoulli_sim(0.5)~n, type="l", col=rgb(runif(3),runif(3),runif(3)
```


Thus, we can easily see that when p = 0.5, n = 100 the sample mean converges to the true mean of distribution for all 15 replications.

Now,for n=100 and p=0.99

```
> plot(n,Bernoulli_sim(0.99) , type = "l",ylim = c(0.5,1.2))
> replicate_plot=replicate(15, lines(Bernoulli_sim(0.99)~n, type="l", col=rgb(runif(3),runif(3),runif(3)
```
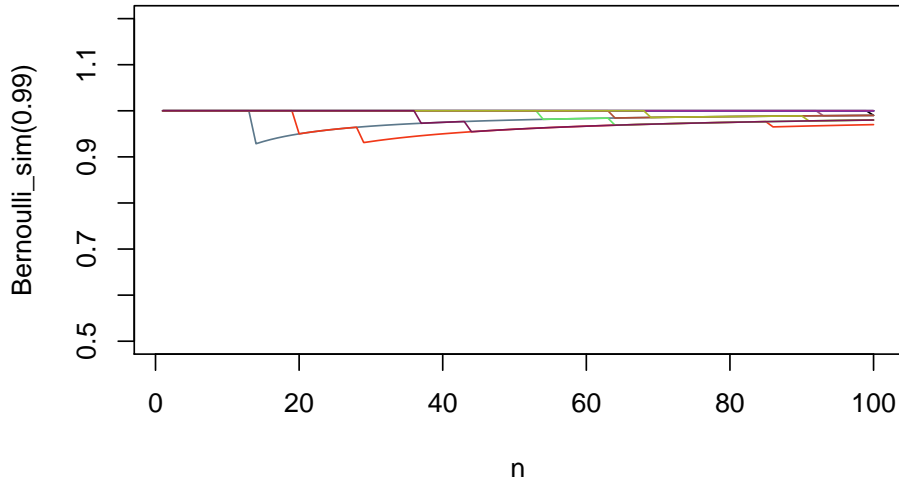

Thus, we can easily see that when p = 0.99, n = 100 the sample mean converges to the true mean of distribution for all 15 replications.

<div align="center">Problem:2(a)</div>

Here we have to compute

$$\int_a^b f(x)dx$$

using the Law of Large Number. We generate samples Generate samples of $X_1, X_2, \ldots X_n$ i.i.d. Uniform (a,b).. Therefore $X_i's$ are iid follows $Uniform(a,b)$. Thus, pdf of each X_i, i=1, 2, . . , n is, $P(x) = 1/(b-a)$, $a \le x \le b$ As $X_i's$ are random samples then $f(X_i)$ for $i = 1, 2, ..., n$ are also n random samples. Then the true mean or population mean of $f(X_i)'s$ is,

$$E(f(x)) = \int_a^b f(x)P(x)dx$$

$$= \frac{b}{1-a} \int_a^b f(x)dx$$

The sample mean of $f(x)$ is

$$S = \frac{f(X_1) + f(X_2) + ... + f(X_n)}{n} = \frac{1}{n}\sum_{i=1}^n f(x_i)$$

The Weak Law of Large Numbers says that as sample size increases sample mean converge to true mean or population mean Let $\epsilon > 0$ then,

<div align="center">5</div>

$$\lim_{n \to \infty} P[|S - E(f(x))| > \epsilon] = 0$$

Hence,for large n

$$S \approx E(f(x))$$

,

$$\frac{1}{n} \sum_{i=1}^{n} f(X_i) \approx \frac{1}{b-a} \int_{a}^{b} f(x)$$

$$\frac{b-a}{n} \sum_{i=1}^{n} f(X_i) \int_{a}^{b} f(x)$$

for large n Hence it is proved.

<div align="center">Problem:2(b)</div>

```
> integral_value = function(x)
+ {
+   func = (16+ sin(x))/(x^2+4)
+   return (func)
+ }
> sample = runif(400,0,7)
> fx = integral_value(sample)
> estimate = 7*(sum(fx)/400)
> estimate # estimate of the integration
```

```
[1] 10.76341
```

<div align="center">Problem:2(c)</div>

```
> c = 0
> for(i in 1:100){
+ sample = runif(400,0,7)
+ c = c + (7*sum(integral_value(sample)))/400
+ }
> # Taking mean to get the approximate value of integral
> mean_integral = c/100
> mean_integral
```

```
[1] 10.57474
```

<div align="center">Problem:2(d)</div>

```
> integrate(f=integral_value,0,7)$value #Actual value of integration
```

<div align="center">6</div>

[1] 10.58204

Thus, we observe that the estimated and actual value of the integration are very close. So the simulation procedure has approximated the value of Integral correctly.

Problem:3(a)

```
> Poisson_X = rep(NA,500)
> for(j in 1:500) {
+ lambda = 10
+ i = 0
+ U = runif(1)
+ cdf = exp(-lambda)
+ while(U >= cdf){
+ i = i + 1
+ cdf = cdf + exp(-lambda)*(lambda^i)/factorial(i)
+ }
+ X = i
+ Poisson_X[j] = X
+ }
> #Computing summary statistics
> summary(Poisson_X)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   2.00    8.00   10.00   10.18   12.00   22.00
```

Problem:3(b)

Here, $X \sim$ p.d.f f given by

$$f(x) = \begin{cases} x & 0 \le x \le \sqrt{2} \\ 0 & \text{otherwise.} \end{cases}$$

The CDF will be

$$F(x) = \int_{-\infty}^{x} x dx$$

$$F(x) = \begin{cases} \frac{x^2}{2} & 0 \le x \le \sqrt{2} \\ 0 & \text{otherwise.} \end{cases}$$

If

$$u = F(x)$$

$$u = \frac{x^2}{2}$$

$$x = \sqrt{2u}$$

as

$$x \ge 1$$

7

```
> u=runif(500,0,1)
> X=sqrt(2*u)
> summary(X)



   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.03233 0.70821 1.03163 0.96438 1.25431 1.41412
```

<div align="center">Problem:3(c)</div>

For generating Normal from uniform distribution we have to use Box-Muller Transformation given as -
Suppose U_1 and U_2 are independent samples chosen from the uniform distribution(0,1). Let us define,

$$X_1 = \sqrt{-2ln(U_1)}cos(2\pi U_2)$$

and

$$Y_1 = \sqrt{-2ln(U_1)}sin(2\pi U_2)$$

Then, X_1 and Y_1 are independent random variables with a standard normal distribution. To obtain Normal(3,4) from X_1, Y_1, we need to transformation as:

$$X_1 = \frac{X - 3}{2}$$

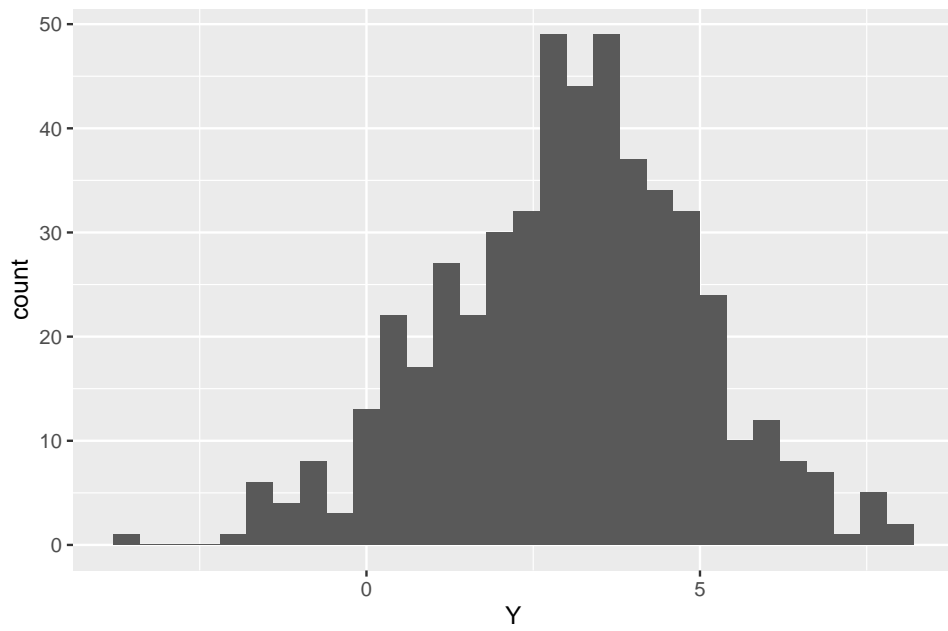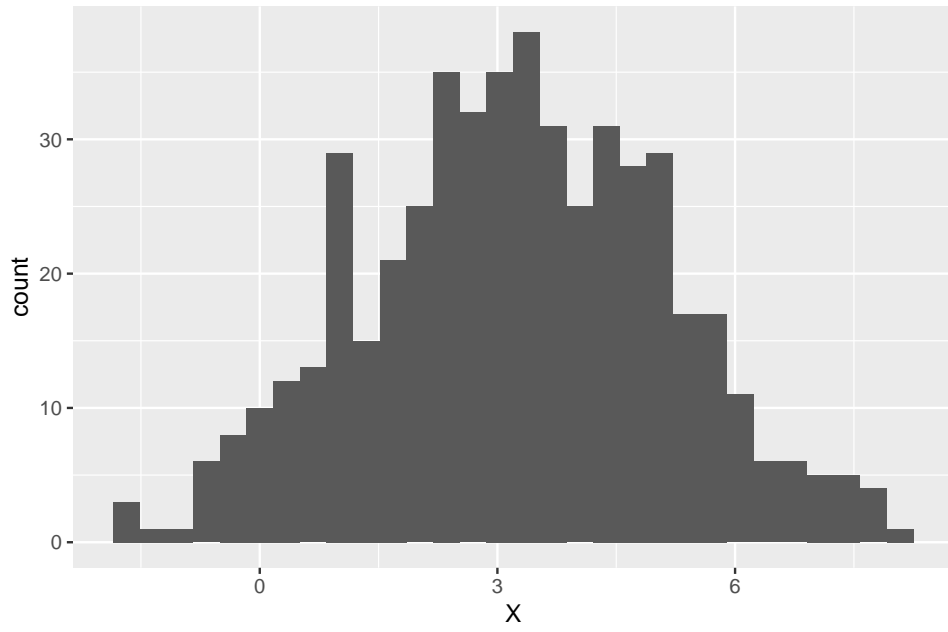Thus,

$$X = 2X_1 + 3$$

and

$$Y_1 = \frac{Y - 3}{2}$$
$$Y = 2Y_1 + 3$$

Thus, X , Y are iid Normal(3,4)

```
> X1=c()
> Y1=c()
> for (k in 1:500){
+ U1=runif(n=1,min=0,max=1)
+ U2=runif(n=1,min=0,max=1)
+ X1[k]=sqrt((-2)*log(U1))*cos(2*pi*U2)
+ Y1[k]=sqrt((-2)*log(U1))*sin(2*pi*U2)
+ }
> X=(X1*2)+3
> Y=(Y1*2)+3
> library(ggplot2)
> df_X=data.frame(X)
> hist_plotX= ggplot(df_X, aes(x=X)) + geom_histogram()
> df_Y=data.frame(Y)
> hist_plotY= ggplot(df_Y, aes(x=Y)) + geom_histogram()
```

Thus, by observing the shape of histograms we can make conclusion that it resembles Normal distribution.