

```
Assign1
lcm <- function(N, x0, a, c, m){
  x <- rep(0,N)
  x[1] <- x0
  for (i in 2:N) {
    x[i] <- (a*x[i-1]+c)%%m
  }
  return(x)
}
```

```
sub1 <- lcm(N = 750, x0 = 1, a = 13, c = 0, m = 101)
sub2 <- lcm(N = 750, x0 = 2, a = 13, c = 5, m = 101)
sub3 <- lcm(N = 750, x0 = 3, a = 13, c = 6, m = 101)
sub4 <- lcm(N = 750, x0 = 4, a = 13, c = 4, m = 101)
sub5 <- lcm(N = 750, x0 = 5, a = 13, c = 8, m = 101)
sub6 <- lcm(N = 750, x0 = 6, a = 13, c = 2, m = 101)
```

```
sub1_mean <- mean(sub1)
print(sub1_mean)
```

```
sub1_median <- median(sub1)
print(sub1_median)
```

```
sd_sub1 <- sd(sub1)
print(sd_sub1)
```

```
hist(sub1)
hist(sub2)
hist(sub3)
hist(sub4)
hist(sub5)
hist(sub6)
```

```
sub.sum <- c(sum(sub1),sum(sub2),sum(sub3),sum(sub4),sum(sub5),sum(sub6))
print(sub.sum)
mean(sub.sum)
median(sub.sum)
sd(sub.sum)
hist(sub.sum)
```

```
outlier_sub1 <- boxplot(sub1)$out
outlier_sub1
outlier_sub2 <- boxplot(sub2)$out
outlier_sub2
outlier_sub3 <- boxplot(sub3)$out
outlier_sub3
outlier_sub4 <- boxplot(sub4)$out
outlier_sub4
outlier_sub5 <- boxplot(sub5)$out
outlier_sub5
outlier_sub6 <- boxplot(sub6)$out
outlier_sub6
```

Assign 3
#Builtin random number generator functions

```
#Skewnwss and Kurtosis
```

```
sub1<-runif(750,min=0,max=100)
sub2<-sample(0:100,750,replace=TRUE)
sub3<-rnorm(750,mean=50,sd=20)
sub4<-randu[2:101,1]*100
```

```
hist(sub1)
hist(sub2)
hist(sub3)
hist(sub4)
```

```
#Mode function
```

```
Mode <- function(x) {
  ux <- unique(x)
  ux[which.max(tabulate(match(x, ux)))]
}
```

```
Mode(sub1)
```

```
#Mean Deviation
```

```
meandev<-function(x)
{
  m=sum(abs(x-mean(x)))/length(x)
  return (m)
}
meandev(sub1)
```

```
#Skewness and Kurtosis
```

```
library(moments)
skewness(sub1)
kurtosis(sub1)
```

```
Assign 4
```

```
# q 1 :-  
s=1  
n=100  
p1=numeric(n)  
for(i in 1:n)  
{  
  s=s*(365-(i-1))/365  
  p=1-s  
  p1[i]=p  
  cat("No.of People=", i-1 , "prob.=", p, "\n")  
}
```

```
plot(p1,main="Probability that at least 2 of them have birthday on same day",xlab="No.of people",ylab="Probability ",col="blue")
```

```
i=1;
c=1;
while(c>0.5){
  c=c*(365-(i))/365;
  i=i+1;
}
cat("Smallest value for n such that probability is greater than .5. =", i,"\\n")
```

```

#q2 :-
# First Part :-
a<-vector(length=11)
x <- c(0:10)
payoff<- (x^2) - (7*x)
probOfHead<-choose(10,x)*(0.6^x)*(0.4^(10-x))
print(probOfHead)
print(payoff)
plot(payoff,main="Payoff",xlab="No.of throws",ylab="Payoff",col="blue")

```

```

# Second Part :-
sum<-0
for(i in 1:11){
  sum = sum + (probOfHead[i] * payoff[i])
}
print(sum)
if(sum>=0){
  print("Good Bet")
}else
{
  print("Bad Bet")
}

```

```

#Third Part :-
probOfhead = .6
a = 10
ntrials = 1000
trials = rbinom(ntrials, a, probOfHead)
payoff = trials^2 - 7*trials
print(mean(payoff))
-----
```

```

#ASSIGNMENT_5
#q1 &q2
n=1e4
exp=replicate(n,rbinom(n=250,c(0,1),prob=0.5))
k<-0

for(i in 1:n){
  y=rle(exp[,i])
  if(length(which(y$lengths==16))>=1)
    k<-k+1
}
p<-k/n
print(p)
#part2
exp1=replicate(10000,rbinom(n=8,c(0,1),prob=0.5))
k1<-0

for(i in 1:n){
  y1=rle(exp[,i])
  if(length(which(y1$lengths>=2))>=0)
    k1<-k1+1
}
```

```
}
```

```
p1<-k1/n
```

```
print(p1)
```

```
#ASSIGNMENT_5
```

```
#ANS 1
```

```
x<- sample(x = c(0,1), size = 250, replace = TRUE,prob=c(0.5,0.5))
```

```
x
```

```
y=dbinom(x,250,0.5)
```

```
y
```

```
rle(y)
```

```
max(rle(y)$lengths)
```

```
z=sum(rle(y)$lengths == 16)
```

```
cat('Probability of same type of items 16 times in a row',z)
```

```
#PART-2
```

```
rounds = 10000
```

```
res_prob_1=c()
```

```
for(i in 1:rounds)
```

```
{
```

```
  x<- sample(x = c(0,1), size = 250, replace = TRUE,prob=c(0.5,0.5))
```

```
  x
```

```
  y=dbinom(x,250,0.5)
```

```
  y
```

```
  rle(y)
```

```
  max(rle(y)$lengths)
```

```
z=sum(rle(y)$lengths == 16)
```

```
#print(paste('Probability of same type of items 16 times in a row',z))
```

```
  res_prob_1[i]=z
```

```
}
```

```
#print(sum(res_prob))
```

```
cat("Estimated Probaility is",sum(res_prob_1)/rounds)
```

```
#ASSIGNMENT_6
```

```
#ANS 3
```

```
n = 10000 ;
```

```
count = 0 ;
```

```
for( i in 1:n ){
```

```
  Animals=c('A','B','C','D','E','F') ;
```

```
  X = sample(Animals,6,replace=FALSE) ;
```

```
  flag = 1 ;
```

```
  for( j in 1:6 ){
```

```
    if( Animals[j] != X[j] ){
```

```
      flag = 0 ;
```

```
      break ;
```

```
    }
```

```
  }
```

```
  if( flag ){
```

```

print(X);
print(Animals);
count = count + 1;
}
}
print( count / n );
# estimated
print(1/factorial(6))

#ANS 4
n = 10000 ;
count = 0 ;
for( i in 1:n ){
  Animals=c('D','D','D','H','H','H') ;
  X = sample(Animals,6,replace=FALSE) ;
  flag = 1 ;
  for( j in 1:6 ){
    if( Animals[j] != X[j] ){
      flag = 0 ;
      break ;
    }
  }
  if( flag ){
    #print(X) ;
    #print(Animals) ;
    count = count + 1 ;
  }
}
print( count / n );
# estimated
print(1/(factorial(6)/factorial(3)))
-----
#Assign 7
r1<-rnorm(1000,mean=1.7,sd=0.1)
r2<-rnorm(10000,mean=1.7,sd=0.1)
plot(density(r1),col='Green')
lines(density(r2),col='Red')

#qnorm(0.95)
#error1<-qnorm(.95)*sd(r1)/sqrt(1000)
#error1<-mean(r1)-error1
#error1<-mean(r1)+error1
#error2<-qnorm(.95)*0.1/sqrt(1000)
#error2<-1.7-error2
#error2<-1.7+error2

left<-qnorm(0.05,mean=1.7,sd=0.1)
right<-qnorm(0.95,mean=1.7,sd=0.1)

percentiles<-seq(0.01,.99,0.01)
qnorm(percentiles)

```

```

z<-seq(0,1,0.05)
pnorm(p)

####Question2
data<-read.csv('auto.csv')
head(data)
correlation<-cor(data$Price,data$MPG)
?cor

numeric_columns<-data[,3:12]
str(numeric_columns)

correlation_matrix<-cor(numeric_columns)
symnum(correlation_matrix)

install.packages('corrgram')
library('corrgram')
corrgram(numeric_columns)
?corrgram

corrgram(numeric_columns,lower.panel = panel.cor,upper.panel = panel.pts)
-----

#Assignment 8
data_reg<-read.csv('regressionDataSet.csv')
str(data_reg)
train<-data_reg[1:13000,]
test<-data_reg[13001:16382,]

relation<-lm(formula=Area~Energy,data = train)
print(relation)

#Energy<-data.frame(test$Energy)
#predicted<-predict(relation,data.frame(Energy=test$Energy))
predicted2<-predict(relation,test)

plot(data_reg$Area,data_reg$Energy,col = "red",main = "Regression",
      abline(relation),cex = 1.3,pch = 16,xlab = "Independent",ylab = "Dependent")
cor(predicted,test$Area)

RMSE = function(m, o){
  sqrt(mean((m - o)^2))
}

Accuracy<-RMSE(predicted,test$Area)

####Question2

x<-rpois(100, 50)
y<-rpois(100, 100)
z<-rpois(100, 150)
df<-data.frame(x,y,z)
lm<-lm(z~x+y,data=df)
summary(lm)

df<-cbind(df,x2)

```

```
model1<-lm(z~x,df)
print(model1)

x2<-x^2
model2<-lm(z~x+x2,df)
print(model2)

model3<-lm(log(z)~x,df)
print(model3)
```