################################################################################

################################################################################

####\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ***INPUT VARIABLES***\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ####

################################################################################

m=6 #Size of sealant box in terms of no. of service #Decision Variable

#N=4 # N Policy #Demand generated before starting the service #Decision Variable

S = 0 #Number of boxes to order. #Code Output #Setting Initial Value #Decision Variable

lambda = 1 # Arrival rate of demand

mu = 3 #Service rate

alpha <- (lambda\*exp(-lambda))

beta <- (mu\*exp(-mu))

#####\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* COSTS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

h = 10 # Waiting Cost

l = 10000 # Wastage Cost

orderingcost = 10000 # Ordering Cost

hdc = 10 # Holding Cost

######\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*COSTS END\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

P = 8 # Life in oPen State

L=100 # Life in seaLed State

ro = lambda/mu

# Number of States for m,N

noofstates = (m-N+1)\*N + (m-N+1)\*(m-N+2)/2 -1 + N

#################################################################################

###### ***Creating States Name Array*** ######

#################################################################################

# The bottom states in transition diagram before the start of the service

a <- matrix(noofstates)

j = 0

for(i in 1:N) {

a[i] = paste(i-1,",",j,sep="")

}

##################################################################################

#The states on the top rectangular block of the state space diagram

for(j in 1:N) {

for(i in 1:(m-N+1)) {

f = N-(j-1)+(i-1)

e = m-(j-1)

a[N+(j-1)\*(m-N+1)+i] = paste(f,",",e,sep="")

}

}

##### The states with a value as 0, the triangular state block.

k=1

for(j in 1:(m-N)) {

for(i in 1:(m-N+1-(j-1))) {

f = i-1

e = m-N-(j-1)

a[N+N\*(m-N+1)+k] = paste(f,",",e,sep="")

k=k+1

}

}

######### All states Defined #########

################################################################################

#### Creation of the TRANSITION PROBABILITY MATRIX ####

################################################################################

TPM <- matrix(0,nrow=noofstates, ncol=noofstates)

rownames(TPM) <- a

colnames(TPM) <- a

################################################################################

#### ***Filling up of the probabilities in the TRANSITION PROBABILITY MATRIX*** ####

################################################################################

for(i in 1:(noofstates)) {

for(j in 1:(noofstates)) {

vars <- scan(text=paste(a[i],a[j],sep=","), sep=",")

if (vars[1]+1 == vars[3] & vars[2] == vars[4] ) {

TPM[i,j] = alpha

} else {

if ((vars[1]-1) == vars[3] & (vars[2]-1) == vars[4] ) {

TPM[i,j] = beta

} else { TPM[i,j] = 0}

}

}

}

TPM[N,N+1] = alpha

################################################################################

k=0

for(i in 1:(noofstates)) {

for(j in 1:(noofstates)) {

k = k+ TPM[i,j]

}

TPM[i,i] = 1-k

k = 0

}

################################################################################

#### ***Power of the matrix to its life in open State*** ####

################################################################################

library(expm)

TPMP <- TPM %^% P

################################################################################

#### ***EXPECTED WASTAGE UNITS*** ####

################################################################################

ESL = 0 #Expected Number of Services Lost

for(i in 1:(noofstates)) {

vars <- scan(text=paste(a[N+1],a[i],sep=","), sep=",")

ESL = ESL + TPMP[N+1,i]\*vars[4]

}

################################################################################

#### ***EXPECTED WASTAGE COST*** ####

################################################################################

EWageC = l\*ESL

################################################################################

#### ***EXPECTED WAITING COST*** ####

################################################################################

EWC = h\*((N/lambda)+P)\*(((N-1)/(2\*mu))+(lambda/((1-ro)\*mu\*mu)))

################################################################################

#### ***EXPECTED Holding COST*** ####

################################################################################

EHC = hdc \* ((N/lambda)+P)

################################################################################

#@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

################################################################################

################################################################################

#### ***CODE FOR THE SECOND STAGE*** ####

################################################################################

################################################################################

# L is the life in cLosed (sealed) state. & . P is the life in oPen state.

# S is the number of boxes to be ordered.

#ESL is Expected Number of Services Lost

#Alternating Renewal Process #YServicePeriod #XHolidayPeriod

Y <- P

X <- (N/lambda)

# L >= [S(X+Y)]

# S <= L/(X+Y)

S <- L/(X+Y)

S = floor (S)

TEWC <- EWC\*S

TEHC <- EHC\*S + hdc \*((S-1)\*(S-2)/2)

TEWageC <- EWageC\*S

TotalCost = TEWC + TEWageC + orderingcost + TEHC

TotalServ = S\*(m-ceiling(ESL))

CostperDelivery = TotalCost/TotalServ

###############################################################################

EVal <- eigen(TPM)$values ## Eigen Values of TPM

eig2 = EVal[2]

T <- 1/log(1/eig2) ## Mixing Time

################################################################################

################################################################################

#lambda

#mu

#alpha

#beta

#L

#P

#m

#ESL

#N

#TEWC

#TEWageC

#S

CostperDelivery

#h

#l