Homework 6

Jushira Thelakkat (jt39634)

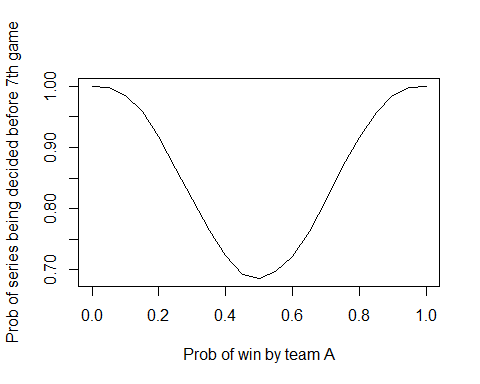
April 18, 2018

### Question 1

Pwin = seq(0, 1, 0.05)  
  
# To find probability that series will be decided before 7th game;  
# let that probability be Pwin7  
# Initialize Pwin7  
  
Pwin7 = c()  
for (i in Pwin) {  
 sim = rbinom(100000,6,i)  
 Pwin7 = c(Pwin7, mean(sim >= 4 | sim <= 2))  
}  
Pwin7

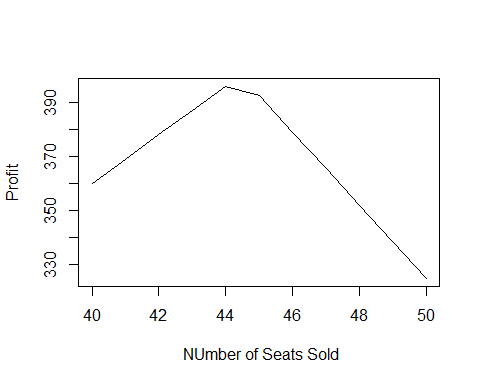
## [1] 1.00000 0.99801 0.98547 0.95901 0.91793 0.86615 0.81477 0.76479  
## [9] 0.72277 0.69311 0.68607 0.69779 0.72246 0.76304 0.81464 0.86835  
## [17] 0.91700 0.95721 0.98541 0.99797 1.00000

# Plotting probabilities for various p values  
  
plot(Pwin, Pwin7, type="l", xlab="Prob of win by team A", ylab="Prob of series being decided before 7th game")



### Question 2

# The minimum seats booked would be 40; let's take 40-50 as the range with increments of 1  
n = seq(40, 50, 1)  
p = c()  
for (i in n) {  
 nShow = mean(rbinom(100000,i,0.9))  
 Revenue = nShow \* 10  
 nExcess = nShow - 40  
 if (nExcess > 0) {  
 Cost = nExcess \* 25  
 Profit = Revenue - Cost  
 }   
 else {  
 Profit = Revenue #Cost is zero here  
 }  
 p = c(p,Profit)  
}  
  
#Plotting the Number os sold seats vs the Profit for each  
  
plot(n, p, type="l", xlab=" NUmber of Seats Sold", ylab="Profit")



max(p)

## [1] 395.9771

# At what number of seats is the Profit maximum  
n[which.max(p)]

## [1] 44

Therefore, 44 tickets must be sold!

### Question 3

library(triangle)

library(lpSolveAPI)  
  
time = c()  
num\_k = 0  
num\_l = 0  
  
for (i in (1:10000)) {  
 l = matrix(0,16,16)  
 l[1,2] = rltriangle(a=8, b=16, c=9) #A  
 l[2,3] = l[2,4] = rltriangle(a=4, b=12, c=5) #B  
 l[3,5] = l[3,7] = rltriangle(a=5, b=7, c=6) #C  
 l[4,6] = l[4,7] = rltriangle(a=4, b=16, c=13) #D  
 l[5,6] = rltriangle(a=3, b=5, c=4) #E  
 l[6,8] = l[6,9] = rltriangle(a=2, b=4, c=3) #F  
 l[7,10] = rltriangle(a=4, b=8, c=6) #G  
 l[8,12] = rltriangle(a=10, b=18, c=11) #H  
 l[9,11] = rltriangle(a=3, b=3, c=3) #I  
 l[10,13] = rltriangle(a=12, b=16, c=14) #J  
 l[11,12] = rltriangle(a=3, b=5, c=4) #K  
 l[12,14] = l[12,13] = rltriangle(a=2, b=4, c=3) #L  
 l[13,15] = rltriangle(a=8, b=8, c=8) #M  
 l[14,15] = rltriangle(a=6, b=22, c=11) #N  
 l[15,16] = rltriangle(a=3, b=6, c=4) #O  
   
 lanProj <- make.lp(0,16\*16)  
   
 #set objective coefficients  
 set.objfn(lanProj, as.vector(t(l)))  
   
 #set objective direction  
 lp.control(lanProj,sense='max')  
   
 nodes = c(1:16)  
   
 rhs = c(1,rep(0,14),-1)  
 for (n in 1:16){  
 coef = c(l[n,1:16]/l[n,1:16],-l[1:16,n]/l[1:16,n])  
 ind = c((n-1)\*16+c(1:16),(c(1:16)-1)\*16+n)  
 nz = is.finite(coef)  
 add.constraint(lanProj,coef[nz], "=",rhs[n],ind[nz])   
 }  
   
 ColNames = c()  
 RowNames = c()  
 for(i in 1:16){  
 for(j in 1:16){  
 ColNames = cbind(ColNames,paste("x",i,",",j, sep=""))  
 }  
 RowNames=cbind(RowNames,paste("node",i))  
 }  
   
 dimnames(lanProj) <- list(RowNames, ColNames)  
 set.type(lanProj, c(1:256), "binary")  
   
 #write to text file  
 write.lp(lanProj,'lanProj.lp',type='lp')  
   
   
 #solve the model, if this return 0 an optimal solution is found  
 status = solve(lanProj)  
   
 #this return the proposed solution  
 time = c(time, get.objective(lanProj))  
 x = get.variables(lanProj)  
 if (sum(matrix(x,16,16)[,11]) > 0) {  
 num\_k = num\_k + 1  
 }  
 if (sum(matrix(x,16,16)[,12]) > 0) {  
 num\_l = num\_l + 1  
 }  
}  
  
mean(time)

## [1] 62.81269

We ran 10,000 simulations; the expected (mean) completion time is 62.9 secs.

num\_k/10000

## [1] 0

K falls on the longest path 0% of the time

num\_l/10000

## [1] 0.8117

L falls on the longest path 81.17% of the time.

### Question 4

car=c()  
pick=c()  
  
# Using sw for switch option  
Sw =c()   
SwitchWin = 0  
StickWin = 0  
t<-proc.time()  
for (s in 1:10000){  
 # 33 doors  
 car[s]=sample(33,1)  
 pick[s]=sample(33,1)  
   
 x=setdiff(c(1:33),union(pick[s],car[s]))  
 host=x[sample(length(x),5)]  
   
 StickWin = StickWin + (car[s] == pick[s])  
   
 Sw[s]=sample(setdiff(c(1:33),union(pick[s],host)),1)  
 SwitchWin = SwitchWin + (car[s] == Sw[s])  
}

SwitchWin/10000

## [1] 0.0341

This is the probabilty of winning if Switch is made

StickWin/10000

## [1] 0.0299

This is the probability of winning if switch is not made

### Question 5

Here we need the ten probablity values

max = rep(0,10)  
  
for (i in (1:10000)) {  
 Rem = 100 # A portion of this remaining amount will be allotted to next person and so on  
 All = c() #Initiallizing the allocations  
   
 p <- c(runif(9),1)  
 for (j in p) {  
 newAll = Rem \* j  
 All = c(All, newAll)  
 Rem = Rem - newAll  
 }  
 max[which.max(All)] = max[which.max(All)] + 1  
}  
  
max/10000

## [1] 0.6264 0.2443 0.0916 0.0278 0.0073 0.0018 0.0005 0.0002 0.0000 0.0001

The ten probabilities are thus shown above