kb42582_HW5

Khyathi Balusu

4/11/2020

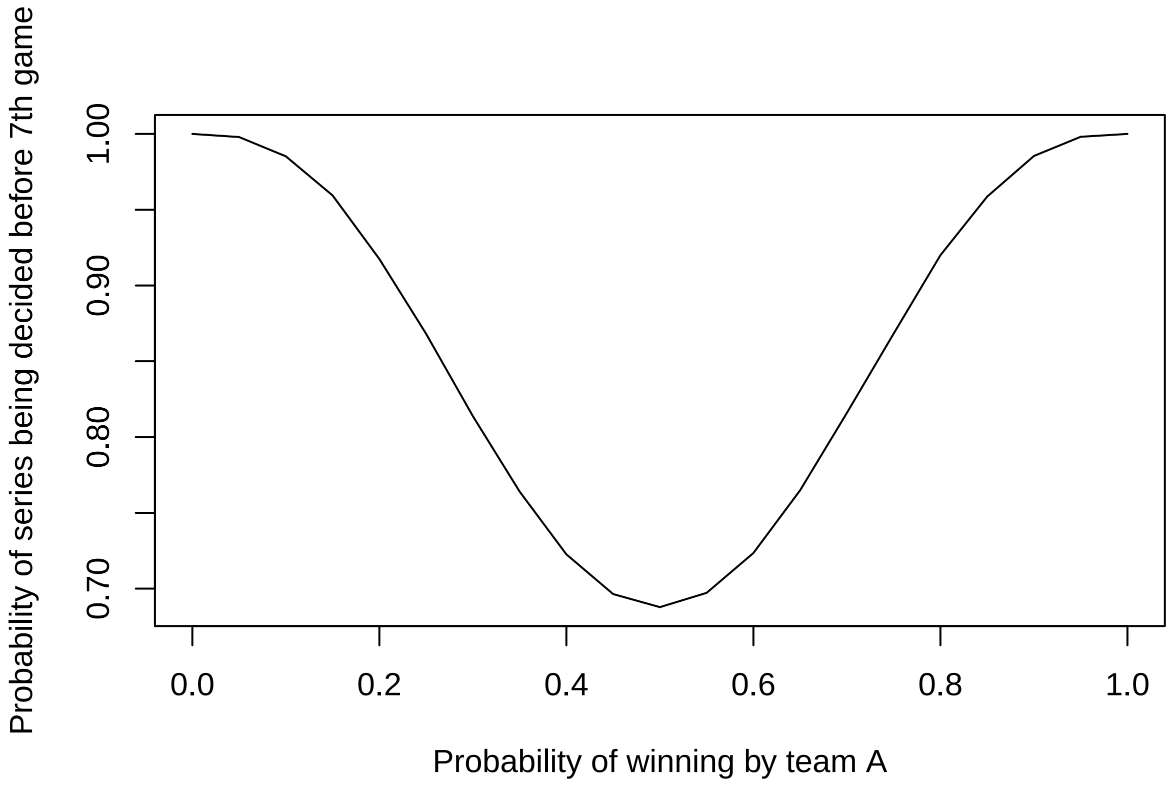
Q1

```
WinProb = seq(0, 1, 0.05)

WinProb7 = c()
for (i in WinProb) {
  sim = rbinom(100000,6,i)
  WinProb7 = c(WinProb7, mean(sim >= 4 | sim <= 2))
}
WinProb7
```

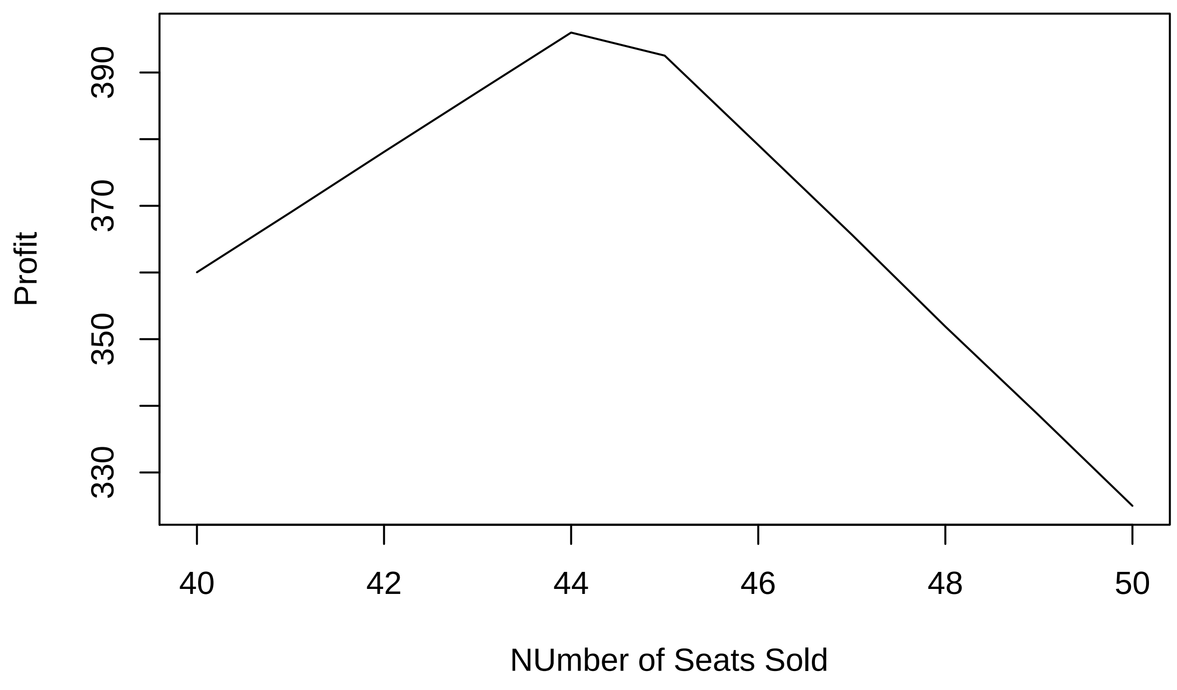
```
##  [1] 1.00000 0.99794 0.98524 0.95941 0.91748 0.86811 0.81378 0.76403
##  [9] 0.72260 0.69641 0.68778 0.69720 0.72348 0.76491 0.81595 0.86830
## [17] 0.91999 0.95865 0.98546 0.99813 1.00000
```

```
# Plotting probabilities for various p values
plot(WinProb, WinProb7, type="l", xlab="Probability of winning by team A", ylab="Probability of series 
```



Q2

```r
# 40 will be the minimum seats booked; 40-50 as  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 of sold seats vs  Profit for each
plot(n, p, type="l", xlab=" NUmber of Seats Sold", ylab="Profit")
```



```r
max(p)
```

```
## [1] 395.989
```

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

```
## [1] 44
```

Therefore, 44 tickets must be sold.

Q4

```r
car=c()
pick=c()
# Using sw for switch option
Sw =c()
WinSwitch = 0
WinStick = 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)]

  WinStick = WinStick + (car[s] == pick[s])

  Sw[s]=sample(setdiff(c(1:33),union(pick[s],host)),1)
  WinSwitch = WinSwitch + (car[s] == Sw[s])
}
```

```r
WinSwitch/10000
```

```
## [1] 0.0356
```

This is the probabilty of winning if Switch is made

```r
WinStick/10000
```

```
## [1] 0.0305
```

This is the probability of winning if switch is not made

Q5

Here we need the ten probablity values

```r
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] 0.6255 0.2457 0.0876 0.0292 0.0088 0.0025 0.0005 0.0002 0.0000 0.0000
```

Above given are the ten probabilities