Harsh Patel

NetId: hkp49

Assignment 1:

```
monte.hall<-function(N,n=3){ # This is a function name Monte and arguments N=no. of games or trials and n = no. of doors

 winswitch<-NULL    # A variable named winswitch with initial value = NULL. Stores the outcomes when we switch

 winstay<-NULL      # A variable named winstay with initial value = NULL. Stores the outcomes when we stay

 for(i in 1:N){     # a for loop running N times. Hence we are playing the game N times and inside this loop is the code for the game


 true.door<-sample(n,n)  # true.door is a variable that stores sample(n,n) i.e. choosing from n values and choosing exactly n values. #For ex. Sample(10,2) selecting 2 values between 1 to 10. 10 specifies upper bound.

 choice.door<-sample(n,1)     # choice.door is a variable that stores sample(n,1) i.e. choosing from n values and choosing exactly 1 values.

                    #For ex. Sample(10,1) selecting 1 value1 between 1 to 10. 10 specifies upper bound.

 Mdoor<-sample(true.door,1)   # Monte hall opens a door and we assign it to Mdoor

 if(choice.door==true.door[1]){     # here if our choice == prize door

  winswitch<-c(winswitch,0)        # then add 0 to winswitch vector

  winstay<-c(winstay,1)            # and add 1 to winstay vector

 }


 else {

 winswitch<-c(winswitch,1/(n-2))  # Else add 1 to winswitch vector

 winstay<-c(winstay,0)       # add 0 to winstay vector

 }

 }

 m1<-cbind(winswitch,winstay)   # combining the two vectors winstay and winswitch

  apply(m1,2,sum)/N       # here we are finding the probabilities of winning.

                    #Hence, dividing the individual wins of winstay and winswitch vectors by N and getting   the win percentage for each case
```

}

Results:

```
> monte.hall(100,3)
winswitch   winstay
     0.64       0.36
> monte.hall(10000,3)
winswitch   winstay
   0.6543     0.3457
>
```