Stat 110: R Section

Credit to Joe Blitzstein, Kenneth Baclawski, & Matt DosSantos DiSorbo

Justin Zhu

Monty Hall

You are on a game show, where Mr. Monty Hall is the host. There are three closed doors. A car has been placed behind one of the doors at random; there are goats behind the other two doors (assume that you prefer the car to the goat). You pick a door; say, for the purpose of this example, you pick Door 1. Monty, who knows where the car is, then opens up Door 2 to reveal a goat (importantly, Monty will always open a door with a goat, and if you pick the door with the car so that the other two doors have goats behind them, he will select one of the goat doors to open with equal probabilities). He then offers you the option to stay with Door 1 or switch to Door 3. Should you switch doors?

```
#simulate
set.seed(110)
# Declare variables
sims = 1000

#door that monty opens
monty = rep(NA, sims)

#keep track of where the car is
car = rep(NA, sims)

#run the loop
for(i in 1:sims){

  #generate the doors randomly; 1 is the car, 0 is a goat
  doors = sample(c(1, 0, 0))

  #mark where the car is
  car[i] = which(doors == 1)

  #if we picked the car, Monty opens another door at random
  if(car[i] == 1){
    monty[i] = sample(c(2, 3), 1)
  }

  #if we picked a goat, monty opens the other goat door
```

```
  if(car[i] != 1){

    #label the picked door; always pick door 1
    doors[1] = 1

    #monty picks the other goat
    monty[i] = which(doors == 0)
  }
}

#probability that the car is behind door 1 or door 3

length(car[car == 1 & monty == 2])/length(car[monty == 2])
```

```
## [1] 0.326087
```

```
length(car[car == 3 & monty == 2])/length(car[monty == 2])
```

```
## [1] 0.673913
```

Gambler's Ruin

How do we track the amount that A has throughout the game?

```
# We create our variable N for total money
# In this case 10 dollars
N <- 10; N
```

```
## [1] 10
```

```
# We create our probability p,
# In this case it's 1/2,
# 50-50 chance of winning for A and B
p <- 1/2; p
```

```
## [1] 0.5
```

```
time <- sample(50:100, 1); time
```

```
## [1] 50
```

```
A_values <- rep(0,time); A_values
```

```
##  [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [36] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```r
# Initially, let us allocate 5 dollars for each gambler
A_values[1] = 5; A_values[1]
```

```
## [1] 5
```

```r
# Alternatively, we can also randomly allocate any value between 0 and N dollars
# A_values[1] = sample(1:N-1, 1); A_values[1]

# We now simulate subsequent values of Player A's Markov chain
for (i in 2:time){
  if (A_values[i-1]==0 || A_values[i-1]==N){
    A_values[i] <- A_values[i-1]
  }
  else{
    A_values[i] <- A_values[i-1] + sample(c(1,-1),1, prob=c(p,1-p))
  }
}

A_values[time]; A_values
```

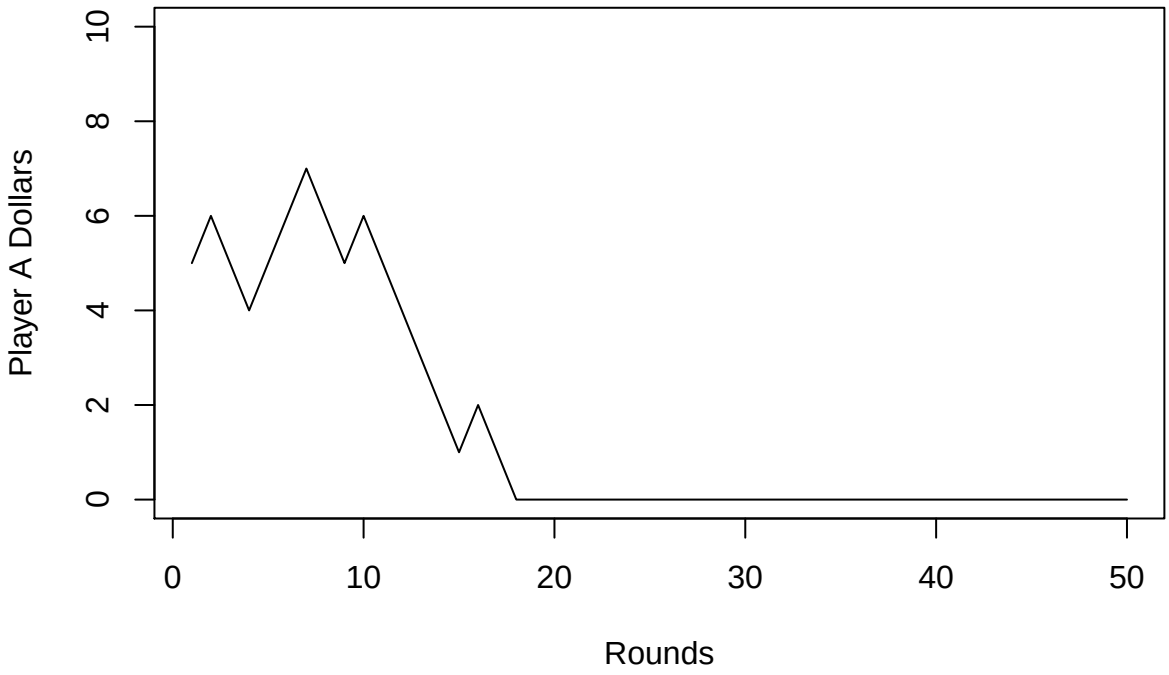
```
## [1] 0
```

```
##  [1] 5 6 5 4 5 6 7 6 5 6 5 4 3 2 1 2 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [36] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

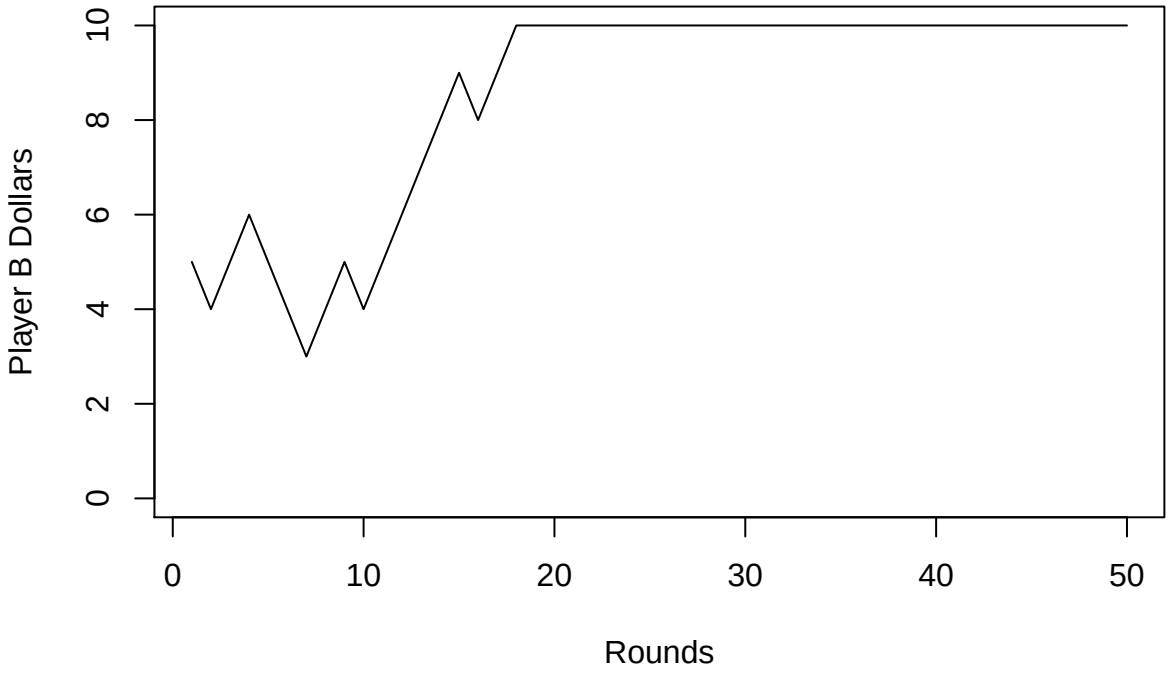
```r
B_values <- N-A_values; B_values
```

```
##  [1]  5  4  5  6  5  4  3  4  5  4  5  6  7  8  9  8  9 10 10 10 10 10 10
## [24] 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10
## [47] 10 10 10 10
```

```r
plot(A_values, type='l',ylim=c(0,N),xlab="Rounds",ylab="Player A Dollars")
```




```
plot(B_values, type='l',ylim=c(0,N),xlab="Rounds",ylab="Player B Dollars")
```



Run these tests again by changing each of the parameters. What do you notice?

1. N
2. p
3. time
4. how the funds are initally allocated