STT 810

Homework 1

Aditya Jain

2022-09-16

Contents

Question 1

This is a simpler version of a more rigorous experiment we will do later in the semester.

a. Use the sample function to define a variable with 100 simulations of rolling a single 6-sided die.

```
a1 <- sample(x = seq(1,6),size = 100, replace = TRUE)

a1
```

```
##   [1] 5 5 5 6 3 1 3 2 3 4 5 6 3 3 6 1 2 1 4 3 5 2 6 1 5 3 2 1 3 6 3 6 4 1 1 3 3
##  [38] 1 2 1 3 6 2 6 4 6 6 1 2 3 1 1 2 1 2 1 3 2 2 4 5 3 5 6 3 3 6 6 5 1 1 1 4 4
##  [75] 3 4 1 2 2 6 5 2 2 6 5 3 3 5 5 3 3 3 2 1 2 5 1 6 2 2
```

b. Find the average value of this simulation. You can use the mean() function to find the mean.

```
paste(mean(a1), "is the average mean of all the simulations")
```

```
## [1] "3.25 is the average mean of all the simulations"
```

c. Next, duplicate (1) and (2) with

- 1,000 simulations
- 10,000 simulations
- 100,000 simulations
- 1,000,000 simulations

Is the mean approaching the true value?

```
c1 <- lapply(c(1000,10000,100000,1000000,10000000,100000000),
             function(x) sample(x = seq(1,6),size = x, replace = TRUE))

lapply(c1 ,function(x) return(paste("The mean is",mean(x))))
```

```
## [[1]]
## [1] "The mean is 3.445"
##
## [[2]]
## [1] "The mean is 3.5042"
##
## [[3]]
## [1] "The mean is 3.49452"
##
## [[4]]
## [1] "The mean is 3.49855"
##
## [[5]]
## [1] "The mean is 3.4998249"
##
## [[6]]
## [1] "The mean is 3.5001451"
```

The mean is approaching the true value.

d. Next, repeat the experiment with flipping a fair coin, with a 1 being heads and 0 being tails. Same convergence?

```
d1 <- lapply(c(100,1000,10000,100000,1000000,10000000,100000000),
             function(x) sample(x = c(0,1),size = x, replace = TRUE))

lapply(d1 ,function(x) return(paste("The mean is",mean(x))))
```

```
## [[1]]
## [1] "The mean is 0.5"
##
## [[2]]
## [1] "The mean is 0.484"
##
## [[3]]
## [1] "The mean is 0.4962"
##
## [[4]]
## [1] "The mean is 0.49759"
##
## [[5]]
## [1] "The mean is 0.499561"
##
## [[6]]
## [1] "The mean is 0.5001331"
##
## [[7]]
## [1] "The mean is 0.50002368"
```

Same Conclusion, mean is approaching its true value

e. Finally, repeat the experiment with an unfair coin which comes up heads 2/3 of the time.

```
e1 <- lapply(c(100,1000,10000,100000,1000000,10000000,100000000),
             function(x) sample(x = c(0,1),size = x, replace = TRUE,prob = c(1/3,2/3)))

lapply(e1 ,function(x) return(paste("The mean is",mean(x))))
```

```
## [[1]]
## [1] "The mean is 0.69"
##
## [[2]]
## [1] "The mean is 0.656"
##
## [[3]]
## [1] "The mean is 0.6593"
##
## [[4]]
## [1] "The mean is 0.66581"
##
```

```
## [[5]]
## [1] "The mean is 0.666628"
##
## [[6]]
## [1] "The mean is 0.6670697"
##
## [[7]]
## [1] "The mean is 0.66662258"
```

The mean is approaching the input probability of the sample size

Question 2

A 2 6-sided dice are rolled (each with values 1, 2, 3, 4, 5, 6). The outcome of the roll is found by the difference between the larger and smaller numbers (so that if a 3 and 5 are rolled, the result is a 2, if a 5 and a 1 is rolled, the results is a 4, if the results is a 3 and a 3, the results is a 0, etc.)

a. Find the sample space.

```r
d1 <- sample(x = seq(1,6),size = 1000, replace = TRUE)
d2 <- sample(x = seq(1,6),size = 1000, replace = TRUE)
sample_size <- sort(unique(abs(d2-d1)))
sample_size
```

```
## [1] 0 1 2 3 4 5
```

b. Find the probability that the result is a 1.

Ans.

P(x-y=1) = probability of getting difference 1
Total number of possible outcomes = 36
Total number of outcomes where difference is 1 = 10
So, P(x-y=1) = 10/36 = 0.27 = 27.7 %

c. Create a simulation of 10,000 rolls of the 2 dice. Calculate the difference and find the proportion of rolls for which the result is a 1. Does your result approximately agree with what you got in (b)?

```r
# creating a roll function to run one simulation
roll <- function(nd) return(sample(1:6,nd,replace = TRUE))

# creating a total probability calculation function for K as the difference between rolls
probtotk <- function(d,k,nreps){
  diff <- replicate (nreps , abs(diff(roll(d))))
  return(mean(diff == k)*100)
}

p_1 <- probtotk(2,1,10000)
paste(p_1, "% is the probability to get 1 as a difference.")
```

```
## [1] "28.28 % is the probability to get 1 as a difference."
```

Question 3

In a certain game, 1 6-sided die is rolled and 2 coins are flipped. A person will win if the die rolls exactly the same value as the number of heads flipped.

a. What is the probability of winning the game?

Ans.

> probabilty of one head = 2/4
> probabilty of Two head = 1/4
> probability of getting one on dice = 1/6
> probability of getting two on dice = 1/6
>
> probability of win = probability of one head*probability of getting one on dice + probabilty of Two head*probability of getting two on dice
>
> P(win) = 3/24 = 0.125 = 12.5%

b. Create a simulation of 10,000 plays of the game. Does the number of wins approximately agree with (a)?

```r
# creating roll and toss function to run one simulation
roll <- function(nd) return(sample(1:6,nd,replace = TRUE))
toss <- function(d) return(sample(0:1,d,replace = TRUE))

# creating total probability function
probtotk <- function(nd,d,nreps){
  dice <- replicate(nreps , roll(nd))
  coin <- replicate(nreps , sum(toss(d)))
  return(mean(dice==coin))
}
p_1 <- probtotk(1,2,10000)
paste(p_1*100, "% is the winning probability")
```

```
## [1] "12.37 % is the winning probability"
```

> **The number of wins does seems to approximate our answer in question a**

Question 4

A survey of a group's viewing habits over the last year revealed the following information:

a. 28% watched gymnastics
b. 29% watched baseball
c. 19% watched soccer
d. 14% watched gymnastics and baseball
e. 12% watched baseball and soccer
f. 10% watched gymnastics and soccer
g. 8% watched all three sports.

Calculate the percentage of the group that watched none of the three sports during the last year.

Ans.

> G = gymnastics, B = baseball, S = soccer
> Using the below formula of set theory
> n(G or B or S) = n(G) + n(B) + n(S) - n(G & B) - n(G & S) - n(B & S) + n(G & B & S)
> n(G) + n(B) + n(S) - n(G & B) - n(G & S) - n(B & S) + n(G & B & S) = 48 %
> Percentage of people watching any one sport atleast - 48 %
> Percentage of people not waching any sports - 100- 48 = 52 %
>
> **venn diagram in the image uploaded**

Question 6

Consider the Preferential Attachment Graph model, Section 1.10.1.

a. Find P(N3 = 1 | N4 = 1).
b. Find P(N4 = 3)

Ans.

> P(N3=1|N4=1) = (P(N3=1) * P(N4=1|N3=1))/ ((P(N3=1) * P(N4=1|N3=1)) + (P(N3=2) * P(N4=1|N3=2)))
> equals = (05 * 0.5)/(0.5 * 0.5 + 0.5 * 0.25) = 0.66
>
> P(N4 = 3) = 1/4 (Total number of connections = 4 and connections on node 3 = 1)

Question 7

Modify the simulation code in the broken-rod example, Section 2.6, so that the number of pieces will be random, taking on the values 2, 3 and 4 with probabilities 0.3, 0.3 and 0.4.

```
# creating a function to simulate one observation and obtain minimum length of broken piece
minpiece <- function ( k ) {
  breakpts <- sort (runif(sample(2:k,size = 1,prob = c(0.3,0.3,0.4)) -1))
  lengths <- diff ( c (0 , breakpts ,1))
  min ( lengths )
}

# Calculating mean to get probability for nreps simulation
bkrod <- function ( nreps ,k , q ) {
  minpieces <- replicate ( nreps , minpiece ( k ))
  mean ( minpieces < q )
}

satisfy <- bkrod (10000 ,4 ,0.02)

paste(satisfy*100, "% of the pieces are of length < 0.02")
```

```
## [1] "13.15 % of the pieces are of length < 0.02"
```

Question 8

Alter the code in Section 2.4 to find the probability that the bus will be empty when arriving to at least one stop among the first 10.

```
# defining variables
nreps <- 10000
nstops <- 10
zcount <- 0
for ( i in 1: nreps ) {
  passengers <- 0 # starting with passenger zero and reaching the first stop
  newpass <- sample (0:2 ,1 , prob = c (0.5 ,0.4 ,0.1)) # simulating new passenger at stop 1
  passengers <- passengers + newpass # adding passengers
  for ( j in 2: nstops ) { # starting loop from stop 2
    if (passengers == 0){ # testing if passengers are zero
      count <- 1 # assigning value to count
    }
    if (passengers > 0){ # running simulation of alight passengers
      for ( k in 1: passengers )
        if ( runif (1) < 0.2){
          passengers <- passengers - 1
        }
    }

    newpass <- sample (0:2 ,1 , prob = c (0.5 ,0.4 ,0.1)) # for new passengers at each stop
    passengers <- passengers + newpass

  }
  if (count > 0) {
    zcount <- zcount + 1 # calculating total count
    count <- 0
  }
}

print(paste("Probability that the bus will be empty atleast one stop is", 100*(zcount/nreps),"%"))
```

```
## [1] "Probability that the bus will be empty atleast one stop is 63.64 %"
```

Alternate and faster solution

```
nreps <- 10000
nstops <- 10
count <- 0
for ( i in 1: nreps ) {
  passengers <- 0
  newpass <- sample (0:2 ,1 , prob = c (0.5 ,0.4 ,0.1))
  passengers <- passengers + newpass
  for ( j in 2: nstops ) {
    if (passengers == 0){
      count <- count + 1 # counting within loop by breaking simulation when passengers are zero
      break
    }
    if (passengers > 0){
```

```
      for ( k in 1: passengers )
        if ( runif (1) < 0.2){
          passengers <- passengers - 1
        }
    }

    newpass <- sample (0:2 ,1 , prob = c (0.5 ,0.4 ,0.1))
    passengers <- passengers + newpass

  }
}

print(paste("Probability that the bus will be empty atleast one stop is", 100*(count/nreps),"%"))
```

```
## [1] "Probability that the bus will be empty atleast one stop is 64.33 %"
```

Thank You!