Assignment 3, Social Science Inquiry II (SOSC13200-W23-2)

Tiffanie Huang

Monday 1/23/22 at 5pm

1.

Consider the random process of flipping a fair coin three times.

(1a)

Write an R object, `Omega`, that is a vector whose elements describe the sample space in terms of heads and tails. E.g., three heads in a row could be described as 'HHH'.

total number of possible outcomes in *Omega*: `2^3`

```
Omega <- c("HHH","HHT","HTH","THH","TTH","THT","HTT","TTT")
```

(1b)

The random variable X that we're interested in is the number of heads that we get from our random process. Write a data.frame object with two columns. One column, `X`, describes all of the possible number of heads we could get. The second column, `probs`, describes the probability each of these events occurs.

Print your data.frame so that it shows in your report.

Hint: the coin is fair, so each of the outcomes in the sample space above occurs with equal probability. Note how many heads we get in each outcome. Then look at the proportion of times we get no heads, one head, etc. These proportions are equal to the probability.

```
#get number of heads for each element in Omega
x <- numeric(length(Omega))
split <- unlist(strsplit(Omega,""))
for(i in 1:length(split)){
  if(i%%3==0){
    x[i/3] <- sum(split[(i-2):i]=="H")
  }
  probs <- numeric(length(unique(x)))
  for(j in 1:length(probs)){
    probs[j] <- mean(unique(x)[j]==x)
  }
}

#data.frame for X heads and respective probabilities
X<- unique(x)
```

```r
h <- data.frame(X,probs)
print(h)
```

```
##   X probs
## 1 3 0.125
## 2 2 0.375
## 3 1 0.375
## 4 0 0.125
```

(1c) Calculate the mean of X.

```r
mean(x) #1.5
```

```
## [1] 1.5
```

(1d)

Write out code to simulate this random process, where the output is a single realization of the random variable (i.e., a number that represents the number of heads in your coin flips).

Note: I set a random seed here, so that every time you recompile your assignment, you'll get the same number. For analyses that involve sampling or random processes, it is really important to set a random seed so that you can get reproducible results. Feel free to change the seed number to anything you want. In general you should only set your random seed ONCE per script.

```r
set.seed(67676)

#simulation of random process
rand <- function(n){
  sample(x,n,replace=TRUE)
}

#output: single realization -- 3
rand(1)
```

```
## [1] 3
```

(1e)

Now run your random process so you sample from it 10,000 times [PLEASE DON'T OUTPUT ALL 10,000 OBSERVATIONS IN YOUR HOMEWORK, just save it to an R object]. What is the average number of heads across these 10k observations? This is the sample mean for a given sample.

```r
#output: 10,000 observations sampled from random process
n10k <- rand(10000)

#avg # of heads = 1.5126 (pretty close to true mean)
mean(n10k)
```

```
## [1] 1.5126
```

(1f)

Write your own function called `mymean()` to calculate the sample mean from a vector. Apply your function to your size 10k sample that you saved in the last problem.

(Don't use `mean()` inside your function, and don't call the specific object you created in the last question inside your function. Your `mymean()` function should work when applied to any vector.)

```
#function to manually calculate mean
mymean <- function(m){
  return(sum(m)/length(m))
}

#apply to n=10k sample = 1.5126 (same as before)
mymean(n10k)
```

```
## [1] 1.5126
```

(1g)

Re-run the code from 1f to get another length 10k sample from the same random process. [DON'T PRINT THIS WHOLE OBJECT.] Apply your `my_mean()` function to it.

```
n10k2 <- rand(10000) #new 10k-long sample
mymean(n10k2) #new mean = 1.5227
```

```
## [1] 1.5227
```

2.

Using the same random process of flipping three fair coins, code the random variable Y as 1 if we get three heads, and 0 otherwise.

(2a)

Write a data.frame object with two columns. One column, `Y`, describes all of the possible values of Y we could get. The second column, `probs`, describes the probability each of these events occurs.

Print your data.frame so that it shows in your report.

```
#set Y to 1 if we get three heads
y <- x==3
Y <- 1*unique(y)
probs <- c(mean(y==TRUE),mean(y==FALSE))

#data.frame for possible values of Y and respective probabilities
h3 <- data.frame(Y,probs)
print(h3)
```

```
##   Y probs
## 1 1 0.125
## 2 0 0.875
```

(2b)

Write a new data.frame object that has three columns. Two columns, `X` and `Y`, jointly describe the values that `X` and `Y` can take on together. The third column, `probs`, describes the probability each of these pairs of events occurs jointly.

Print your data.frame so that it shows in your report.

```
#redefine Y for values of X
Y <- 1*(X==3)

#probs -- p[a]*p[b|a] = p[a]*1, a for x, b for y
probs <- h$probs

#data.frame
joint <- data.frame(X,Y,probs)
print(joint)
```

```
##   X Y probs
## 1 3 1 0.125
## 2 2 0 0.375
## 3 1 0 0.375
## 4 0 0 0.125
```

(2c)

Report the conditional mean of X given that Y equals 0.

Recall that conditional probability can be written as:

$$\mathrm{P}[A|B] = \frac{\mathrm{P}[AB]}{\mathrm{P}[B]}$$

```
#P[mean(X)|Y=0] = P[mean(X) and Y=0]/P[Y=0]

XcondY <- mean(x[which(y==FALSE)])
print(XcondY)
```

```
## [1] 1.285714
```

(2d)

Are the events that $X = 3$ and that $Y = 1$ are independent?

```
#P[AB] ?= P[A]*P[B] #independence test?

#P[X=3] -- event A
a <- joint$probs[which(joint$X==3)]
print(a)
```

```
## [1] 0.125
```

```r
#P[Y=1] -- event B
b <- joint$probs[which(joint$Y==1)]
print(b)
```

```
## [1] 0.125
```

```r
#P[A]*P[B] -- both (if independent)
a*b #0.015625
```

```
## [1] 0.015625
```

```r
#P[AB] -- draw from both
ab <- joint$probs[which(joint$X==3 & joint$Y==1)]
print(ab) #vs 0.125
```

```
## [1] 0.125
```

```r
#Independence test:
print(ab ==a*b) #false -- not independent
```

```
## [1] FALSE
```

```r
#These events are not independent, since the joint probability of both events does not equal the produc
```