02 Conditional Probability

Harsh Patel

2024-09-17

```
#define parameters
balls = rep(c("R", "W"), c(3,3))
number.draws = 2
replicates = 10

#create empty vector to store results
successes = vector("numeric", replicates)

#set the seed for a pseudo-random sample
#set.seed(5011)

#simulate the draws
for(k in 1:replicates){

draw = sample(balls, size = number.draws, replace = FALSE)

  if(draw[1] == "W" & draw[2] == "R"){
successes[k] = 1
  }

}
#view the results
successes

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

table(successes)

## successes
## 0 1
## 8 2
```

#a) The vector balls includes elements R and W, each repeated three times, representing the color and count of balls in the bag. To update the vector to reflect a bag with 5 red balls and 2 white balls, adjust c(3, 3) to c(5, 2).

#b) The vector draw() is created by randomly sampling from the balls vector without replacement. The sample size is specified by number.draws, which is set to 2. Since the prob vector is not defined, sample() defaults to equal probability for each element.

#c) A success is recorded when the first element of draw is W (white) and the second is R (red). Thus, a success occurs when the first draw is a white ball and the second is a red ball.

#d) Out of the 10 trials, successes were observed on the second, fifth, seventh, and eighth draws.

```
which(successes == 1)
```

```
## [1] 6 7
```

#e) The estimated probability of drawing a white ball first and a red ball second is approximately 0.307.

```
#define parameters
balls = rep(c("R", "W"), c(3,3))
number.draws = 2

replicates = 10000

#create empty vector to store results
successes = vector("numeric", replicates)

#set the seed for a pseudo-random sample
#set.seed(2018)

#simulate the draws
for(k in 1:replicates){

  draw = sample(balls, size = number.draws, replace = FALSE)
  if(draw[1] == "W" & draw[2] == "R"){

successes[k] = 1
  }

}

#view the results
table(successes)
```

```
## successes
##    0    1
## 6945 3055
```

```
#estimate the probability
sum(successes)/replicates
```

```
## [1] 0.3055
```

#f) Using the multiplication rule for conditional probabilities, P(A and B) = P(B|A)P(A), where W1 is the event of drawing a white ball first and R2 is drawing a red ball second, the probability is P(W1 and R2) = P(R2|W1)P(W1) = (3/5) * (1/2) = 3/10. This result is close to the 0.307 estimated in part e.

```r
#define parameters
balls = rep(c("R", "W"), c(5,2))
number.draws = 2
replicates = 10000

#create empty vector to store results
successes = vector("numeric", replicates)

#set the seed for a pseudo-random sample
#set.seed(5011)

#simulate the draws
for(k in 1:replicates){

draw = sample(balls, size = number.draws, replace = FALSE)

  if(draw[1] == "W" & draw[2] == "R"){
successes[k] = 1
  }

}
#view the results
head(successes,10)

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

table(successes)

## successes
##    0    1
## 7650 2350
```

#2.

#a) To draw exactly one red ball in two picks, there are two possible scenarios: drawing a red ball first followed by a white ball, or a white ball first followed by a red ball. The total probability of drawing exactly one red ball is the sum of the probabilities of these two mutually exclusive events. As calculated previously, the probability of red second is known. Now, calculate the probability of drawing a red ball first and a white ball second: P(R1 and W2) = P(W2|R1)P(R1) = (3/5) * (1/2) = 3/10. Thus, P(exactly one R) = P(W1 and R2) + P(R1 and W2) = (3/5) * (1/2) + (3/5) * (1/2) = 3/5. The probabilities of red-first-white-second and white-first-red-second are equal, given the equal number of red and white balls.

#b) The estimated probability of drawing exactly one red ball is 0.60.

```r
#define parameters
balls = rep(c("R", "W"), c(3,3))
number.draws = 2
replicates = 10000
```

```r
#create empty vector to store results
successes = vector("numeric", replicates)

#set the seed for a pseudo-random sample
#set.seed(2018)

#simulate the draws

for(k in 1:replicates){

  draw = sample(balls, size = number.draws, replace = FALSE)

if( (draw[1] == "W" & draw[2] == "R") | (draw[1] == "R" & draw[2] == "W") ){

  successes[k] = 1
  }
}

#view the results
table(successes)
```

```
## successes
##    0    1
## 4024 5976
```

```r
#estimate the probability
sum(successes)/replicates
```

```
## [1] 0.5976
```

#c) To express the condition, use sum(draw == “R”) == 1.

```r
#define parameters
p.female = 0.50
p.tall.if.female = 0.03
p.tall.if.male = 0.20
population.size = 10000

#create empty vectors to store results
sex = vector("numeric", population.size)
tall = vector("numeric", population.size)

#set the seed for a pseudo-random sample
#set.seed(2018)

#assign sex
sex = sample(c(0,1), size = population.size, prob = c(1 - p.female,
p.female),
replace = TRUE)
```

```r
#assign tall or not
for (k in 1:population.size){

  if (sex[k] == 0) {
tall[k] = sample(c(0,1), prob = c(1 - p.tall.if.male, p.tall.if.male),
                 size = 1, replace = TRUE)
  }

if (sex[k] == 1) {
tall[k] = sample(c(0,1), prob = c(1 - p.tall.if.female, p.tall.if.female),
                 size = 1, replace = TRUE)
  }

}

#view results
addmargins(table(sex, tall))
```

```
##      tall
## sex       0     1   Sum
##   0    4021  1023  5044
##   1    4795   161  4956
##   Sum  8816  1184 10000
```

#3.

#a) Assuming a 1:1 ratio of males to females, P(F) = 0.50. The probability of being tall if female is 0.03, and if male, it's 0.20

#b) Use the sample function to assign values of 0 or 1 based on the probabilities of not being female (1 – P(F)) and being female (P(F)). Ensure every individual in the population is assigned a value by setting size = population.size.

#c) The probability of being tall is contingent on sex, as it varies between males and females. Use if statements to correctly assign height status based on the sex-specific probabilities.

#d) Use the sample() function to determine height status based on sex, with probabilities 1 – P(T|M) and P(T|M) for males, and 1 – P(T|F) and P(T|F) for females. Since the for loop processes one individual at a time, set size = 1.

#e) Calculate the desired probabilities using the table data or R. The estimated probability of being a female taller than 6 feet is 0.017, and the overall probability of being tall is 0.119.

```r
#probability of female and tall
sum(tall == 1 & sex == 1)/population.size
```

```
## [1] 0.0161
```

```r
#probability of tall
sum(tall)/population.size
```

```
## [1] 0.1184
```

#f) To find P(F and T), apply the multiplication rule: P(A and B) = P(A|B)P(B). Hence, P(F and T) = P(T|F)P(F) = (0.03) * (0.50) = 0.015. Since tall individuals can be either female or male, P(T) is P(F and T) + P($\bar{F}$ and T). Thus, P(T) = P(T|F)P(F) + P(T|$\bar{F}$)P($\bar{F}$) = (0.03) * (0.50) + (0.20) * (0.50) = 0.115.

#4.

#a) Given the assumptions, use the multiplication rule to compute genotype frequencies. Let p denote the allele frequency of A, and q denote the frequency of a. – The frequency of AA is $p^2 = (0.90)^2 = 0.81$. – The frequency of Aa is 2pq = (2)(0.90)(0.10) = 0.18. An individual can inherit an A allele from one parent and an a from the other, or vice versa, which are mutually exclusive events. – The frequency of aa is $q^2 = (0.10)^2 = 0.01$.

```r
#define parameters
p.disease.AA = 0.8
p.disease.Aa = 0.4
p.disease.aa = 0.1

p.AA = 0.81
p.Aa = 0.18
p.aa = 0.01

population.size = 10000

#create empty vectors to store results
genotype = vector("numeric", population.size)
disease = vector("numeric", population.size)

#set the seed for a pseudo-random sample
#set.seed(2018)

#assign genotype
genotype = sample(c("AA", "Aa", "aa"), size = population.size,
prob = c(p.AA, p.Aa, p.aa), replace = TRUE)

#assign disease status
for(k in 1:population.size){

  if(genotype[k] == "AA"){
disease[k] = sample(c(0, 1), size = 1,
                    prob = c(1 - p.disease.AA, p.disease.AA),
                    replace = TRUE)
  }

  if(genotype[k] == "Aa"){
```

```
disease[k] = sample(c(0, 1), size = 1,
                    prob = c(1 - p.disease.Aa, p.disease.Aa),
                    replace = TRUE)
  }

  if(genotype[k] == "aa"){
disease[k] = sample(c(0, 1), size = 1,
                    prob = c(1 - p.disease.aa, p.disease.aa),
                    replace = TRUE)
  }

}
#view results
addmargins(table(genotype, disease))
```

```
##          disease
## genotype     0     1   Sum
##       aa     99    13   112
##       Aa   1063   717  1780
##       AA   1638  6470  8108
##       Sum  2800  7200 10000
```

#b)

#i. The estimated prevalence of disease, P(D), is 0.718.

```
sum(disease)/population.size
```

```
## [1] 0.72
```

#ii. The estimated probability of having genotype AA given the disease is 0.900. This is calculated by dividing the number of individuals with disease and genotype AA by the total number of individuals with the disease: 6464/7182 = 0.900.

```
sum(genotype == "AA" & disease == 1)/sum(disease)
```

```
## [1] 0.8986111
```

#iii. To find the probability of developing the disease, P(D), consider the three possible genotypes: AA, Aa, and aa. Use the formula P(D) = P(D ∩ AA) + P(D ∩ Aa) + P(D ∩ aa). Given conditional probabilities P(D|AA), P(D|Aa), and P(D|aa), and using the definition of conditional probability, compute: P(D) = P(D|AA)P(AA) + P(D|Aa)P(Aa) + P(D|aa)P(aa) = (0.8)(0.81) + (0.4)(0.18) + (0.1)(0.01) = 0.721. The probability of being genotype AA given the disease is: P(AA|D) = P(D|AA)P(AA) / P(D) = (0.8 * 0.81) / 0.721 = 0.899.