Week_4_Lab_4_Probability

Upol Chowdhury

2024-09-19

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

1. What is the probability of drawing a white ball on the first pick and a red on the second?

Run the following code to simulate the results for 10 sets of two draws from the bag, where red and white balls are represented by R and W, respectively.

```
#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 1 0 0 1 0 0 1 0
```

```
table(successes)
```

```
## successes
## 0 1
## 7 3
```

1 a) The command is a generic way to replicate elements of a vector `x` a certain number of times. Describe what the vector `balls` contains, and explain how the code to create the vector could be modified for a bag that contains 5 red balls and 2 white balls.

```
#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
  }

}
```

```
table(successes)
```

```
## successes
##    0    1
## 7691 2309
```

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

```
## [1] 0.2309
```

2 b) Using simulation, estimate the probability of drawing exactly one red ball

```
#define parameters
balls = rep(c("R", "W"), c(3,3))
number.draws = 2
replicate = 10000
#create empty vector to store results
success = vector("numeric", replicate)
#set the seed for a pseudo-random sample
#set.seed(2018)
#simulate the draws
```

```
for(k in 1:replicate){
draw = sample(balls, size = number.draws, replace = FALSE)
if( (draw[1] == "W" & draw[2] == "R") | (draw[1] == "R" & draw[2] == "W") ){
  success[k] = 1
}
}
#view the results
table(success)
```

```
## success
##    0    1
## 4015 5985
```

```
#estimate the probability
sum(success)/replicate
```

```
## [1] 0.5985
```

3. In the United States population, approximately 20% of men and 3% of women are taller than 6 feet (72 inches). Let F be the event that a person is female and T be the event that a person is taller than 6 feet. Assume that the ratio of males to females in the population is 1:1.

```
#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)
#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    4057   964  5021
##   1    4814   165  4979
##   Sum  8871  1129 10000
```

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

```
## [1] 0.0165
```

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

```
## [1] 0.1129
```

4.Suppose a disease is caused by a single gene, with alleles A and a; the alleles have frequency

0.90 and 0.10 ### b) Suppose the disease is not fully penetrant, so that the probability of developing the disease is 0.8 for genotype AA, 0.4 for genotype Aa, and 0.1 for genotype aa. Simulate a population of 10,000 individuals, recording their genotype and disease status

```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

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   101     9   110
##       Aa  1065   705  1770
##       AA  1592  6528  8120
##       Sum 2758  7242 10000
```

What is the prevalence (overall probability) of disease in the population?

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

```
## [1] 0.7242
```

Given that an individual is known to have the disease, what is the probability they are genotype AA?

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

```
## [1] 0.9014085
```

Positive Predictive Value (Bayes' Theorem)

```
#define parameters
population.size = 100000
prevalence = 1/800
sensitivity = 0.980
specificity = 0.995
#create empty vectors to store results
disease.status = vector("numeric", population.size)
test.result = vector("numeric", population.size)
#set the seed for a pseudo-random sample
#set.seed(2018)
#assign disease status (part a)
disease.status = sample(c(0,1), size = population.size,
prob = c(1 - prevalence, prevalence),
replace = TRUE)
#assign test result (part b)
```

```
for(k in 1:population.size){
if(disease.status[k] == 0){
test.result[k] = sample(c(0,1), size = 1,
prob = c(specificity, 1 - specificity))
}
if(disease.status[k] == 1){
test.result[k] = sample(c(0,1), size = 1,
prob = c(1 - sensitivity, sensitivity))
}
}
#create matrix of disease status and test result (part c)
disease.status.and.test.result = cbind(disease.status, test.result)
#create a table of test result by disease status
addmargins(table(test.result, disease.status))
```

```
##             disease.status
## test.result      0      1    Sum
##         0    99403      3  99406
##         1      469    125    594
##         Sum  99872    128 100000
```

```
#calculate ppv (part d)
ppv = sum(test.result[disease.status == 1])/sum(test.result)
ppv
```

```
## [1] 0.2104377
```

```
#calculate npv (part e)
npv = sum(test.result == 0 & disease.status == 0) / sum(test.result == 0)
npv
```

```
## [1] 0.9999698
```

4.The strongest risk factor for breast cancer is age; as a woman gets older, her risk of developing breast cancer increases. The following table shows the average percentage of American women in each age group who develop breast cancer, according to statistics from the National Cancer Institute. For example, approximately 3.56% of women in their 60's get breast cancer.

```
#define parameters
population.size = 100000
prevalence = 0.0044
sensitivity = 0.85
specificity = 0.95
#create empty vectors to store results
disease.status = vector("numeric", population.size)
test.result = vector("numeric", population.size)
#set the seed for a pseudo-random sample
#set.seed(2018)
```

```
#assign disease status
disease.status = sample(c(0,1), size = population.size,
prob = c(1 - prevalence, prevalence),
replace = TRUE)
#assign test result
for(k in 1:population.size){
if(disease.status[k] == 0){
test.result[k] = sample(c(0,1), size = 1,
prob = c(specificity, 1 - specificity))
}
if(disease.status[k] == 1){
test.result[k] = sample(c(0,1), size = 1,
prob = c(1 - sensitivity, sensitivity))
}
}
#calculate expected number of positive tests
sum(test.result)
```

```
## [1] 5314
```

```
#calculate ppv
ppv = sum(test.result[disease.status == 1])/sum(test.result)
ppv
```

```
## [1] 0.0769665
```

5 a) Calculate the missing PPV and NPV values, using any method.

```
prevalence = c(0.001, 0.020, 0.060, 0.100)
sensitivity = rep(0.20, 4)
specificity = rep(0.94, 4)
ppv.numerator = prevalence*sensitivity
ppv.denominator = ppv.numerator + (1 - prevalence)*(1 - specificity)
ppv = ppv.numerator/ppv.denominator
ppv
```

```
## [1] 0.003325574 0.063694268 0.175438596 0.270270270
```

```
npv.numerator = (1 - prevalence)*specificity
npv.denominator = npv.numerator + (prevalence)*(1 - sensitivity)
npv = npv.numerator/npv.denominator
npv
```

```
## [1] 0.9991488 0.9829279 0.9484757 0.9136069
```