Week_3_Intro_to_probability

Upol Chowdhury

2024-09-12

#Introduction

```
#define parameters
prob.heads = 0.6
number.tosses = 5
#Stimulate the coin tosses
outcomes = sample(c(0,1), size = number.tosses, prob = c(1 - prob.heads, prob.heads), replace = TRUE)
#view the resutls
table(outcomes)
## outcomes
## 0 1
## 2 3
#Store the results in single number
total.heads = sum(outcomes)
total.heads
## [1] 3
#deifne parameter
prob.heads = 0.6
number.tosses = 5
number.replicates = 10000
#create empty vector to store outcomes
outcomes = vector("numeric", number.replicates)
#set the seed for a pseudo-random sample
set.seed(2018)
#simulate the coin tosses
for(k in 1:number.replicates){
  outcomes.replicate = sample(c(0, 1), size = number.tosses,
                      prob = c(1 - prob.heads, prob.heads), replace = TRUE)
```

```
outcomes[k] = sum(outcomes.replicate)
}
#view the results
outcomes
```

```
[1] 5 3 1 2 4 4 4 4 2 3 3 3 4 3 4 4 4 3 3 2 4 4 2 3 3 3 2 2 3 3 2 4 3 3 1 3 2
##
      [37] 2 5 0 2 3 3 3 4 3 4 4 3 5 4 4 2 5 2 2 4 2 4 3 2 3 3 3 4 3 3 3 2 3 3 3 4
##
##
      [73] 2 3 3 3 3 2 3 2 1 3 3 4 2 4 1 3 3 2 4 2 3 3 2 2 5 2 2 1 1 3 3 4 3 3 2 3
     [109] 3 3 3 4 3 3 2 3 4 4 3 1 3 4 3 4 2 3 4 1 1 3 4 3 4 3 3 3 3 3 3 2 3 2 4 3 3
##
##
     [145] 5 4 3 3 3 4 3 5 5 3 3 2 3 3 2 4 5 3 4 3 1 3 5 3 4 4 1 4 3 3 1 4 3 3 3 4
     [181] 2 3 2 2 4 2 3 4 4 3 3 2 5 3 4 1 5 2 5 2 3 2 2 3 3 3 4 2 2 0 4 3 2 4 2 2
##
##
     [217] 4 3 2 2 5 5 3 3 4 5 2 4 4 2 3 2 2 5 5 2 3 5 3 2 4 3 4 4 2 2 3 3 2 4 5 3
##
     [253] 5 2 3 3 4 4 4 4 4 2 2 4 4 3 3 4 2 2 5 2 4 2 3 4 3 3 5 2 2 4 0 3 2 4 4 4
##
     [289] 3 3 5 2 3 3 2 2 1 3 2 4 2 3 1 4 4 3 3 3 4 5 3 2 2 2 4 5 3 3 4 4 3 2 5 3
##
     [325] 1 4 3 2 2 5 4 5 2 3 2 4 0 2 2 2 4 3 2 2 4 3 3 1 3 4 2 3 2 4 3 5 5 3 2 3
     [361] 4 3 4 1 3 4 3 4 4 2 2 2 3 2 2 4 4 4 2 2 2 3 1 2 3 1 2 5 3 2 2 2 3 2 4 5
##
##
     [397] 2 2 3 0 3 4 2 4 3 4 3 5 2 3 3 4 2 2 4 5 3 3 3 4 3 4 3 4 1 5 4 3 3 3 3 2
##
     [433] 4 4 3 3 4 1 4 1 4 5 4 4 2 4 1 3 3 2 3 4 3 3 3 2 2 4 3 5 4 5 1 2 3 2 3 3
##
     [469] 3 4 4 4 4 3 2 3 3 2 3 4 3 3 5 3 3 5 3 2 2 2 3 2 3 3 4 1 4 4 4 4 3 3 5
##
     [505] 4 4 4 3 4 1 2 3 4 4 3 3 4 2 2 3 2 4 4 1 4 3 1 3 2 3 4 2 1 3 2 3 1 3 3 3
##
     [541] 4 3 4 3 4 3 4 3 4 3 2 2 1 3 2 5 3 1 4 3 5 2 3 4 5 1 3 3 5 3 4 1 2 2 3 3
##
     [577] 3 3 1 2 2 4 3 3 3 3 4 2 4 1 4 3 4 4 3 2 5 4 5 2 4 2 3 2 3 3 3 4 2 4 3 1
     [613] 3 5 3 3 5 4 3 1 2 3 3 2 3 2 3 5 3 2 3 4 4 3 4 4 4 3 3 3 2 5 4 3 2 4 3 3
##
##
     [649] 3 3 3 3 3 2 3 4 2 4 2 3 2 3 3 3 4 3 3 4 3 3 4 3 2 1 2 4 1 3 2 2 2 4 5 3 2
##
     [685] 4 2 4 2 4 2 2 3 1 4 2 2 4 4 3 5 3 1 3 3 2 4 4 4 2 3 3 4 5 3 5 4 4 4 3 3
##
     [721] 4 4 4 2 3 4 4 4 4 3 3 4 4 4 2 2 3 2 3 4 4 3 2 5 4 3 3 3 4 3 3 3 4 3 4 2
##
     [757] 4 1 5 3 4 2 2 1 4 2 2 3 3 5 3 5 3 5 3 1 4 3 4 1 3 2 2 2 4 2 3 4 3 4 4 3
     [793] 2 4 2 3 3 4 3 4 4 4 2 3 5 2 4 3 4 1 2 4 5 4 3 3 4 4 1 3 2 2 3 4 4 2 4 3
##
##
     [829] 2 5 4 3 4 1 5 5 4 3 3 3 4 2 2 3 3 3 3 5 2 1 3 4 5 2 3 4 1 3 4 2 3 3 4
     [865] 5 3 3 2 2 5 2 4 4 3 2 2 2 2 3 4 4 5 2 1 3 4 1 2 4 3 4 1 5 2 2 2 4 2 3 4
##
##
     ##
     [937] 4 5 4 1 3 4 3 3 1 4 2 2 2 2 2 5 5 0 3 1 2 3 3 2 3 4 4 4 3 5 4 4 3 4 4 4
     [973] 4 3 2 3 5 2 2 3 3 3 4 2 5 1 5 4 3 4 2 4 3 4 5 4 1 3 3 3 4 4 4 4 1 2 4 2
##
##
    [1009] 4 4 5 2 2 2 3 5 3 4 1 4 5 2 3 5 4 2 1 5 4 5 2 4 2 3 4 3 4 3 5 3 1 2 3 3
##
    [1045] 3 3 2 4 5 4 1 1 1 3 3 4 3 2 3 5 2 4 5 4 3 2 4 3 3 4 0 4 3 3 2 3 3 4 4 4
    [1081] 3 1 3 2 3 5 3 3 3 4 3 4 2 2 4 2 2 0 1 3 3 3 4 2 2 2 4 3 3 4 3 4 4 3 2 4
##
    [1117] 2 4 2 2 3 1 3 2 2 5 3 2 4 3 4 3 4 2 1 4 1 0 2 3 2 2 4 3 4 4 4 3 3 3 3 3 4
    [1153] 4 3 4 3 3 4 3 4 2 4 3 3 3 2 3 2 3 2 5 2 3 1 4 3 4 4 3 3 5 4 5 4 3 2 4 2
##
    [1189] 1 4 4 3 4 2 3 4 4 4 4 2 5 3 3 0 2 2 2 3 5 3 3 4 3 5 4 3 4 2 3 2 3 2 2 2
##
##
    [1225] 3 5 2 4 4 5 4 5 3 3 1 4 4 1 1 2 2 2 2 4 4 3 3 3 5 4 3 4 3 2 3 3 4 3 2 2
    [1261] 4 3 2 3 1 1 4 4 3 4 3 3 3 4 4 3 2 2 2 0 2 2 3 4 3 5 4 4 4 4 3 5 3 4
##
    [1297] 3 4 2 3 3 4 3 4 3 3 5 4 2 4 4 4 1 3 3 2 4 2 4 2 2 4 5 3 5 4 3 3 3 1 3 4
##
    [1333] 3 2 3 2 1 2 4 3 2 4 5 4 2 3 3 3 4 3 5 3 5 3 3 3 3 3 4 2 4 2 2 4 3 1 2 5
##
##
    [1369] 2 3 4 3 4 5 3 3 3 2 3 4 4 5 5 2 5 4 2 3 3 3 1 4 3 5 1 3 4 3 4 2 2 2 4 1
##
    [1405] 1 4 2 1 1 3 1 5 4 4 4 5 4 3 2 4 3 2 3 5 1 4 3 5 2 5 2 2 4 3 2 3 5 1 2 3
    [1441] 1 2 1 4 3 2 4 2 3 1 3 3 5 2 5 4 3 2 3 4 1 3 4 3 4 1 3 4 0 2 4 5 3 2 4 4
##
##
    [1477] 4 2 5 1 5 2 5 3 1 2 4 4 3 4 2 3 3 5 3 2 2 3 0 4 3 3 3 4 4 5 5 1 2 2 2 4
    [1513] 3 3 4 2 3 4 2 5 2 3 4 3 3 4 4 2 4 4 3 2 3 3 2 2 4 2 2 3 0 2 2 2 3 4 4 3
##
##
    [1549] 3 4 5 1 3 2 1 4 4 5 2 2 2 1 3 3 3 1 3 1 1 4 3 2 4 1 2 5 4 3 4 4 2 4 2 5
     \begin{smallmatrix} 1585 \end{smallmatrix} ] \ 3 \ 4 \ 1 \ 3 \ 5 \ 4 \ 1 \ 4 \ 1 \ 2 \ 1 \ 2 \ 2 \ 3 \ 4 \ 3 \ 3 \ 4 \ 3 \ 5 \ 4 \ 2 \ 3 \ 3 \ 3 \ 3 \ 4 \ 3 \ 5 \ 1 \ 3 \ 1 \ 3 \ 2 \ 4 \ 2 
##
##
    [1621] 1 1 3 3 4 2 3 3 2 3 4 2 2 5 2 3 4 4 4 4 4 3 3 3 3 3 4 5 2 3 4 5 1 4 1 1 4
    [1657] \ 2\ 3\ 4\ 3\ 3\ 4\ 3\ 4\ 3\ 4\ 1\ 2\ 2\ 4\ 3\ 4\ 3\ 2\ 4\ 3\ 2\ 1\ 4\ 4\ 3\ 3\ 3\ 2\ 3\ 2\ 2\ 2\ 2\ 5\ 2\ 3\ 4
```

```
[9469] 4 3 3 4 4 4 4 5 4 5 2 2 1 5 3 2 2 4 5 1 3 5 4 2 4 4 2 2 4 3 2 2 3 2 4 2
##
   [9505] 2 3 2 5 2 3 2 3 3 4 2 3 1 3 2 4 2 2 3 3 2 2 2 3 3 3 3 2 3 2 2 2 3 3 3 3 3
## [9577] 3 3 4 5 3 4 2 4 4 4 4 4 5 2 3 2 2 4 3 1 1 3 3 4 5 4 3 2 1 3 4 3 4 4 2 1
   [9613] 4 1 4 3 2 3 3 5 1 4 2 3 2 3 4 2 5 1 0 5 3 2 3 2 3 3 3 3 3 2 1 4 2 4 4 3
## [9649] 2 2 2 1 3 4 4 2 4 5 3 4 3 3 5 3 2 3 3 2 5 3 4 5 3 4 4 5 3 3 2 4 2 4 3 2
## [9685] 4 2 3 2 2 3 3 5 4 5 2 1 5 1 3 5 4 2 2 3 2 4 5 3 4 4 3 3 2 3 3 2 4 4 4 1
## [9721] 2 3 3 2 2 4 3 2 2 4 5 2 3 3 3 2 2 3 4 4 4 3 4 3 2 2 3 3 3 4 2 3 3 2 3 3
   [9757] 4 4 3 3 3 2 2 2 4 4 2 3 1 2 3 3 4 2 2 4 0 1 3 3 4 3 2 1 1 4 2 4 2 3 3 4
## [9793] 3 3 3 5 3 4 3 4 3 1 3 5 3 3 3 3 2 3 1 3 2 1 4 2 3 2 3 3 3 3 3 3 4 0 3 4 1
## [9829] 3 4 3 3 3 4 3 2 3 1 3 2 3 5 5 4 1 2 2 4 2 3 2 4 2 3 3 3 3 2 3 2 1 2 4 2
## [9865] 5 4 4 2 2 4 3 3 2 2 3 3 4 3 5 4 5 2 3 3 5 3 4 1 4 1 3 0 3 2 3 1 4 3 1 3
## [9901] 4 3 1 3 4 4 4 5 2 2 1 3 3 2 1 3 3 3 2 3 1 3 1 3 3 5 4 4 4 4 4 3 4 1 4 3
## [9937] 4 2 4 4 3 1 2 3 3 2 3 5 1 4 3 2 5 4 3 3 3 2 3 4 3 0 1 4 3 3 4 2 4 2 2 5
## [9973] 5 3 4 2 4 3 4 2 2 4 3 1 2 5 3 4 5 3 3 2 0 3 2 2 4 3 2 1
addmargins(table(outcomes))
## outcomes
##
            1
                  2
                       3
                             4
                                   5
                                       Sum
##
    105
          724 2295 3521 2561
                                 794 10000
heads.3 = (outcomes == 3)
table(heads.3)
## heads.3
## FALSE TRUE
## 6479 3521
outcomes[4]
## [1] 2
prob.heads= 0.6
10*prob.heads^3*(1-prob.heads)^2
## [1] 0.3456
prob.false.positive = 0.012
prob.true.negative = 1 - prob.false.positive
1-prob.true.negative<sup>150</sup>
## [1] 0.836491
prob.false.positive = 0.012
number.employees = 150
#set the seed for a pseudo-random sample
set.seed(2018)
```

```
#simulate the tests
results = sample(c(0,1), size = number.employees,
prob = c(1 - prob.false.positive, prob.false.positive),
replace = TRUE)
#view the results
table(results)
## results
## 0 1
## 148
sum(results)
## [1] 2
#define parameters
prob.false.positive = 0.012
number.employees = 150
number.replicates = 100000
#create empty vector to store results
results = vector("numeric", number.replicates)
#set the seed for a pseudo-random sample
set.seed(2018)
#simulate the tests
for(k in 1:number.replicates){
results.replicate = sample(c(0,1), size = number.employees,
prob = c(1 - prob.false.positive, prob.false.positive),
replace = TRUE)
results[k] = sum(results.replicate)
#view the results
table(results)
## results
                         3
                               4
                                                                  10
## 16282 29847 27015 16201 7183 2524
                                       720
                                               169
at.least.1.pos = (results >= 1)
table(at.least.1.pos)
## at.least.1.pos
## FALSE TRUE
## 16282 83718
#define parameters
specificity = 0.95
number.women = 50
number.replicates = 100000
#create empty vector to store results
results = vector("numeric", number.replicates)
```

```
#set the seed for a pseudo-random sample
#set.seed(2018)
#simulate the tests
for(k in 1:number.replicates){
results.replicate = sample(c(0,1), size = number.women,
prob = c(specificity, 1 - specificity),
replace = TRUE)
results[k] = sum(results.replicate)
#view the results
table(results)
## results
##
      0
                   2
                         3
                                     5
                                                 7
                                                                  10
                                                                        11
             1
                                           6
## 7607 20265 26113 22180 13566 6497 2612
                                               842
                                                     256
                                                            50
at.most.1.pos = (results <= 1)
table(at.most.1.pos)
## at.most.1.pos
## FALSE TRUE
## 72128 27872
#define parameters
specificity = 0.99
number.women = 50
number.replicates = 100000
#create empty vector to store results
results = vector("numeric", number.replicates)
#set the seed for a pseudo-random sample
#set.seed(2018)
#simulate the tests
for(k in 1:number.replicates){
results.replicate = sample(c(0,1), size = number.women,
prob = c(specificity, 1 - specificity),
replace = TRUE)
results[k] = sum(results.replicate)
}
#view the results
table(results)
## results
                                     5
## 60495 30338 7720 1286
                            148
                                    13
at.most.1.pos = (results <= 1)
table(at.most.1.pos)
## at.most.1.pos
## FALSE TRUE
## 9167 90833
#Answer:
```

The probability of no more than one woman will test positive if none have breast cancer increased to 91%. The probability for the previous specificity of 95% was 28%. Thus, The test with 0.99 specificity is better because it results in far fewer false positives when we are doing for large number of replicates. This means fewer women without breast cancer will be incorrectly told they might have it.