R Scripting

Lab for unit 1 - R basics and data structures (I)

Marcus Wurzer

20. September 2022

Please solve the following problems!

1. In the game of European roulette, a wheel is spun in one direction, then a ball is spun in the opposite direction. The ball eventually falls into one of 37 colored and numbered pockets on the wheel. Numbers 1–36 are classified into groups of numbers in three different ways:

   1. Low (1-18) vs. high (19-36)
   2. Odd (1, 3, ..., 35) vs. even (2, 4, ..., 36)
   3. Red (1, 3, 5, 7, 9, 12, 14, 16, 18, 19, 21, 23, 25, 27, 30, 32, 34, 36) vs. black (2, 4, 6, 8, 10, 11, 13, 15, 17, 20, 22, 24, 26, 28, 29, 31, 33, 35)

   Players may choose to place bets on one of these groups (e.g., put their money on low, red, uneven, etc.) and get double their money back if they win. Number 37 is the green zero.

   a. Create a vector that can be used to simulate a fair roulette wheel.
   b. "Spin" the wheel $n = 500$ times and record the results.
   c. Determine the indices for all the (green) zero results.
   d. Extract all the even numbers and assign them to a new object. Do the same for the odd and the (green) zero results, respectively.
   e. Identify the lengths of the three vectors created in d. above.
   f. Compute the *simulated* occurrence probabilities for even, odd and zero and save them in a vector of length 3 with appropriately named elements. What are the *expected* occurrence probabilities for these three outcomes? To what extent do the probabilities differ?

   Hints:

   - R command `%%` gives you the remainder of a division, e.g.:

     ```
     8 %% 2
     ```

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

     ```
     9 %% 2
     ```

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

   - Use the logical operator `&` to separate 0 from the even numbers.

```
# a.
roulette <- 0:36

# b.
set.seed(1)
res <- sample(roulette, 500, replace = TRUE)
# each number has the same probability of being chosen -> fair
```

```r
# c.
(ind.0 <- which(res == 0))
```

```
##  [1]   2  42  72  85 105 136 148 181 221 261 299 315 324 355 393 429 457 480
```

```r
# d.
(res.0 <- res[ind.0]) # Green zeroes
```

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

```r
ind.odd <- res %% 2 != 0
(res.odd <- res[ind.odd]) # Odd numbers
```

```
##   [1]  3 33 13 17  9 33 19  5  9 19 27 19  5  5  5 23 31 13  1 17 21 13  5  5 35
##  [26] 19 21 27 25 29 31 33  9 25 23 11 23 27 21 25 13 21  5 27 21 19 23 15 35  1
##  [51] 25 33  1 33 15 27 23  9 31 11 13 15  1 19  7 35 19 23 21 15 29  1  7 13 35
##  [76] 25  7 35 35  5  3 27 27 19 27 13 11 15  9 33 25  3 21 23 13 17 31  7 19 33
## [101]  1 27 27 13 25 25  3 15 15 33  9  5 25 25 33  3 23 29 21 31  3 35 19 21 21
## [126] 17 35 31 23 13 25 29  5 33 25  1  7 19 19  9 19  1 31  3  7 27  7 19  1 11
## [151] 25 13 33 29 25  7 21  5 29  9 29  5 25  1 31 35 19 27 27 11  3 25 33  3 25
## [176] 15 23 25 35 21  3 35 21  5 35  5 19 17  7  5  1 31 21  1  5 27  1  1 13  5
## [201] 19 19 33 35 19 23 33 19 19 33 29 15 35 29 27 17  3 35 33 25 35 21 31 27 11
## [226] 29 25
```

```r
ind.even <- res %% 2 == 0 & res != 0
(res.even <- res[ind.even]) # Even numbers
```

```
##   [1] 22 32 20 20  6  8 14 20 36 24 36 36 24 14 32 34 22 24 22 10 16 12 24 24 22
##  [26] 28 12 28 32 20 30 16  8 22 18 28 14 28 32 18 34 28 36  2 34  6 34 26  6 18
##  [51] 12 18 16 10 30 34 34 18 10  8 14 36 36 30 36 28 14  2  8 16  4 26 30 28 20
##  [76] 28 36  6  8 26 14 34 34 12  6 24 30 36 28 30 28 32 24 36 14 16 32  6 26  8
## [101] 36 28 10  2 14  4 36 16 32 18 22 26 28 34 32 22 32 34 10 12 36 24 16  8  8
## [126] 10  2 18 32 12  4 34 22 28 34  8 34 34 18 28 26 30 16 36 24  8 20 20 14 36
## [151]  6 26 14 20 16 12 24 30  4 14 28  4 24 36 12 24  4 12 18 10 22  4  4  2 12
## [176] 30  6 26  8 26 16 12  4 22 34 12  2 14 20 20 36 20  2 22  2 36 36 20 26 10
## [201] 10 32 10 10 20 20 16  8 30 22 10 36 24 34 10 12 14  8  4 26 28 36 30 24 20
## [226] 22 20  4 24 16 32 18 16 22 30  4 26 32 28 20  2 14 12 28 12  6  4 32 18 16
## [251] 12  6  8 32 34
```

```r
# e.
length(res.0)
```

```
## [1] 18
```

```r
length(res.odd)
```

```
## [1] 227
```

```r
length(res.even)
```

```
## [1] 255
```

```r
# f.
(res.counts <- c(zero = length(res.0), odd = length(res.odd), even = length(res.even)))
```

```
## zero  odd even
##   18  227  255
```

```r
sim.prob <- res.counts / sum(res.counts) # simulated probabilities
exp.prob <- c(1/37, 18/37, 18/37) # expected probabilities
sim.prob - exp.prob # differences
```

```
##         zero          odd         even
##  0.008972973 -0.032486486  0.023513514
```