Dataquest Guided Project: Mobile App for Lottery Addiction

Cindy Zhang

1/10/2021

Contents

Introduction

This is my solution to Dataquest's Guided Project from the third Probability and Statistics course, which involves building the logic behind an app for treating lottery addiction.

More details such as the RMD and csv files can be found in the repository in GitHub.

Setting up the Core Functions

factorial = calculates factorials combination = calculates combinations

combination function incorporates factorial function. Combination key for the rest of the project, used to calculate probability.

```
factorial <- function(n) {
    final_product <- 1
    for (i in 1:n) {
        final_product <- final_product * i
    }
    return(final_product)
```

```
}

combination <- function(n, k) {
    return(factorial(n)/(factorial(n - k) * factorial(k)))
}
```

Setting up One-ticket Probability Function

Background: in 6/49 lottery, six numbers drawn from a set of 49 numbers, ranged 1 to 49. Winning number must match: {13, 22, 24, 27, 42, 44}.

First version of app calculates probability of winning the big prize with various numbers for a single ticket. First version must include the following details: * Inside the app, the user inputs six different numbers from 1 to 49. * Under the hood, the six numbers will come as an R vector, which will serve as the single input to our function. * The engineering team wants the function to print the probability value in a friendly way — in a way that people without any probability training are able to understand.

```
one_ticket_probability <- function(x) {
  c_lotto <- combination(49, 6)
  probability <- 1/c_lotto
  actual_prob <- sprintf("%1.9f",probability)
  statement <- paste("You have a ", actual_prob, "% chance of winning the big prize.", sep = "" )
  return(statement)
}

vec <- c(13,22,24,27,42,44)
one_ticket_probability(vec)
```

```
## [1] "You have a 0.000000072% chance of winning the big prize."
```

Historical Data Check for Canada Lottery

Saved Canada Lottery data as a data frame, examined number of rows and columns in data set, printed first and last three rows.

```
can_data <- data.frame(read_csv("649.csv"))
nrow(can_data)
```

```
## [1] 3665
```

```
ncol(can_data)
```

```
## [1] 11
```

```
head(can_data, 3)
```

```
##   PRODUCT DRAW.NUMBER SEQUENCE.NUMBER DRAW.DATE NUMBER.DRAWN.1 NUMBER.DRAWN.2
## 1     649           1               0 6/12/1982              3             11
```

```
## 2     649           2               0 6/19/1982              8             33
## 3     649           3               0 6/26/1982              1              6
##   NUMBER.DRAWN.3 NUMBER.DRAWN.4 NUMBER.DRAWN.5 NUMBER.DRAWN.6 BONUS.NUMBER
## 1             12             14             41             43           13
## 2             36             37             39             41            9
## 3             23             24             27             39           34
```

```
tail(can_data, 3)
```

```
##      PRODUCT DRAW.NUMBER SEQUENCE.NUMBER DRAW.DATE NUMBER.DRAWN.1
## 3663     649        3589               0 6/13/2018              6
## 3664     649        3590               0 6/16/2018              2
## 3665     649        3591               0 6/20/2018             14
##      NUMBER.DRAWN.2 NUMBER.DRAWN.3 NUMBER.DRAWN.4 NUMBER.DRAWN.5 NUMBER.DRAWN.6
## 3663             22             24             31             32             34
## 3664             15             21             31             38             49
## 3665             24             31             35             37             48
##      BONUS.NUMBER
## 3663           16
## 3664            8
## 3665           17
```

There are 3665 rows and 11 columns in the dataset.

pmap practice

```
data1 <- c(1, 3, 5)
data2 <- c(2, 4, 6)
data3 <- c(8, 9, 7)
unnamed_list <- list(data1, data2, data3)
named_list <- list(first=data1, second=data2, third=data3)
first_item_sum <- sum(named_list$first[1], named_list$second[1], named_list$third[1])
averages <- pmap(unnamed_list, function(x,y,z) {(x+y+z)/3})
first_average <- as.vector(unlist(averages[1]))
```

Historical Data Check Function

Users compare their ticket against historical lottery data in Canada. function must include the following details: *

```
historical_lots <- pmap(
  list(
    one <- can_data$NUMBER.DRAWN.1,
    two <- can_data$NUMBER.DRAWN.2,
    three <- can_data$NUMBER.DRAWN.3,
    four <- can_data$NUMBER.DRAWN.4,
    five <- can_data$NUMBER.DRAWN.5,
    six <- can_data$NUMBER.DRAWN.6
  ),
  .f <- function(one,two,three,four,five,six) { c(one,two,three,four,five,six)}
)
```

```
check_historical_occurrence <- function(lot) {
  historical_matches <- map(historical_lots, function(x) {setequal(x,lot)})
  num_past_matches <- sum(unlist(historical_matches))
  s <- paste("The combination you entered has appeared ", num_past_matches, " times in the past. ", "You
  return(s)
}

check_historical_occurrence(c(3, 12, 11, 14, 41, 43))
```

```
## [1] "The combination you entered has appeared 1 times in the past. Your chance of winning the big pri
```

```
check_historical_occurrence(c(1, 2, 3, 4, 5, 6))
```

```
## [1] "The combination you entered has appeared 0 times in the past. Your chance of winning the big pri
```

Multi-ticket Probability

```
multi_ticket_probability <- function(n) {
  total <- combination(49,6)
  probability <- n/total
  actual_prob <- sprintf("%1.9f",probability)
  statement <- paste("After buying ", n, " tickets, you have a ", actual_prob, "% chance of winning the
  return(statement)
}

multi_ticket_probability(1)
```

```
## [1] "After buying 1 tickets, you have a 0.000000072% chance of winning the big prize."
```

```
multi_ticket_probability(10)
```

```
## [1] "After buying 10 tickets, you have a 0.000000715% chance of winning the big prize."
```

```
multi_ticket_probability(100)
```

```
## [1] "After buying 100 tickets, you have a 0.000007151% chance of winning the big prize."
```

```
multi_ticket_probability(10000)
```

```
## [1] "After buying 10000 tickets, you have a 0.000715112% chance of winning the big prize."
```

```
multi_ticket_probability(6991908)
```

```
## [1] "After buying 6991908 tickets, you have a 0.500000000% chance of winning the big prize."
```

```
multi_ticket_probability(13983816)
```

```
## [1] "After buying 13983816 tickets, you have a 1.000000000% chance of winning the big prize."
```

Less Winning Numbers - Function

```
probability_less_6 <- function(n) {
  n_combinations_ticket = combination(6,n)
  n_combinations_remaining = combination(49-n, 6-n)
  successful_outcomes = n_combinations_ticket*n_combinations_remaining
  n_combinations_total = combination(49,6)

  prob = (successful_outcomes/n_combinations_total)*100
  actual_prob <- sprintf("%1.9f",prob)
  statement <- paste("You have a ", actual_prob, "% chance of winning a smaller prize with ", n, " winni
  return(statement)
}
```

```
probability_less_6(3)
```

```
## [1] "You have a 2.171081198% chance of winning a smaller prize with 3 winning numbers."
```

```
probability_less_6(4)
```

```
## [1] "You have a 0.106194189% chance of winning a smaller prize with 4 winning numbers."
```

```
probability_less_6(5)
```

```
## [1] "You have a 0.001887897% chance of winning a smaller prize with 5 winning numbers."
```