Lottery App

Winning against the addiction

Emmanuel Messori

21/09/2021

The context

You are a data analyst at a medical institute. You are assigned to assist in the development of a mobile app intended to guide lottery addicts through exercises that will let them better estimate their chances of winning. The hope is that this app will help them realize that buying too many tickets will do little to improve their chances of winning. The institute has a team of engineers that will build the app, but they need you to build the logic behind the app and calculate probabilities.

For the first version of the app, they want us to focus on the 6/49 lottery and build functions that can answer users questions like:

- What is the probability of winning the big prize with a single ticket?
- What is the probability of winning the big prize if we play 40 different tickets (or any other number)?
- What is the probability of having at least five (or four, or three, or two) winning numbers on a single ticket?

The institute also wants us to consider historical data coming from the national 6/49 lottery game in Canada. The data set has data for 3,665 drawings, dating from 1982 to 2018 (we'll come back to this). The 6/49 lottery is one of the first Canadian lotteries game to allow players to pick their own numbers.

```
library(readr)
lottery <- read_csv("https://dsserver-prod-resources-1.s3.amazonaws.com/409/649.csv",
    show_col_types = FALSE)
lottery$`DRAW DATE` <- as.Date(lottery$`DRAW DATE`, format = "%m/%d/%Y")
```

Tasks

Throughout the project, we'll need to calculate repeatedly probabilities and combinations, so wrapping it in a function will save us a lot of time. We'll start by writing two functions:

1. A function that calculates factorials and
2. A function that calculates the numbers of combinations.

In the 6/49 lottery, six numbers are drawn from a set of 49 numbers that range from 1 to 49. The drawing is done without replacement, so once a number is drawn, it's not put back in the set.

To find the number of combinations when we're sampling without replacement and taking only k objects from a group of n objects, we can use the formula:

$$ {}_nC_k = \binom{n}{k} = \frac{n!}{k!(n-k)!} \tag{1} $$

We'll write a function for calculating a combination taking advantage with the `factorial()` function already implemented in R:

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

Probability of winning

In the 6/49 lottery, six numbers are drawn from a set of 49 numbers that range from 1 to 49. A player wins the big prize if the six numbers on their tickets match all the six numbers drawn. For the first version of the app, we want players to be able to calculate the probability of winning the big prize with the various numbers they play on a single ticket (for each ticket a player chooses six numbers out of 49). So, we'll start by building a function that calculates the probability of winning the big prize for any given ticket.

```
prompt_ = "1 2 3 4 5 6"

one_ticket_probability <- function(numbers = prompt_) {
    numbers = as.numeric(unlist((stringr::str_split(numbers, " "))))
    total_combinations <- combinations(49, 6)
    prob <- 1/total_combinations
    print(paste0("The probability of winning with the combination ", paste(numbers,
        collapse = " "), " is ", sprintf("%.8f", prob * 100), " %"))
}

one_ticket_probability()
```

```
## [1] "The probability of winning with the combination 1 2 3 4 5 6 is 0.00000715 %"
```

Historical Data

The data set contains historical data for 3,665 drawings (each row shows data for a single drawing), dating from 1982 to 2018.

```
head(lottery)
```

```
## # A tibble: 6 x 11
##   PRODUCT `DRAW NUMBER` `SEQUENCE NUMBER` `DRAW DATE` `NUMBER DRAWN 1`
##     <dbl>         <dbl>             <dbl> <date>                  <dbl>
## 1     649             1                 0 1982-06-12                  3
## 2     649             2                 0 1982-06-19                  8
## 3     649             3                 0 1982-06-26                  1
## 4     649             4                 0 1982-07-03                  3
## 5     649             5                 0 1982-07-10                  5
## 6     649             6                 0 1982-07-17                  8
## # ... with 6 more variables: NUMBER DRAWN 2 <dbl>, NUMBER DRAWN 3 <dbl>,
## #   NUMBER DRAWN 4 <dbl>, NUMBER DRAWN 5 <dbl>, NUMBER DRAWN 6 <dbl>,
## #   BONUS NUMBER <dbl>
```

Comparing an actual ticket with historical data

Now we're going to write a function that will enable users to compare their ticket against the historical lottery data in Canada and determine whether they would have ever won by now.

```
# list of all the winning combinations
library(purrr)
wn <- pmap(lottery[, 5:10], c, use.names = FALSE)

check_historical_occurrence <- function(n = prompt_) {
    numbers = as.numeric(unlist((stringr::str_split(n, " "))))
    match <- which(sapply(wn, function(x) setequal(numbers, x)))
    if (length(match) > 0) {
        print(paste("Your combination has already occured ", length(match), " times"))
    } else {
        print(paste("Your combination has never occured."))
    }
    one_ticket_probability(numbers = n)
}

check_historical_occurrence()
```

```
## [1] "Your combination has never occured."
## [1] "The probability of winning with the combination 1 2 3 4 5 6 is 0.00000715 %"
```

Multiple tickets

One situation our functions do not cover is the issue of multiple tickets. Lottery addicts usually play more than one ticket on a single drawing, thinking that this might increase their chances of winning significantly. Our purpose is to help them better estimate their chances of winning — on this screen, we're going to write a function that will allow the users to calculate the chances of winning for any number of different tickets.

```
multi_ticket_probability <- function(n_tickets) {
    total_combinations <- combinations(49, 6)
    prob <- n_tickets/total_combinations
    print(paste0("The probability of winning with ", paste(n_tickets), " tickets is ",
        sprintf("%.8f", prob * 100), " %"))
}

tickets <- c(1, 10, 100, 10000, 1e+06, 6991908, 13983816)

walk(tickets, multi_ticket_probability)
```

```
## [1] "The probability of winning with 1 tickets is 0.00000715 %"
## [1] "The probability of winning with 10 tickets is 0.00007151 %"
## [1] "The probability of winning with 100 tickets is 0.00071511 %"
## [1] "The probability of winning with 10000 tickets is 0.07151124 %"
## [1] "The probability of winning with 1e+06 tickets is 7.15112384 %"
## [1] "The probability of winning with 6991908 tickets is 50.00000000 %"
## [1] "The probability of winning with 13983816 tickets is 100.00000000 %"
```

We see how it's *kind* of hard to win the lottery even if we buy 1 Million of tickets.

3,4 or 5 winning numbers

In most 6/49 lotteries there are smaller prizes if a player's ticket matches three, four, or five of the six numbers drawn. As a consequence, the users might be interested in knowing the probability of having three, four, or five winning numbers.

```
probability_less_6 <- function(n_int) {
    potential_combinations <- combinations(6, n_int)
    total_combinations <- combinations(49, n_int)
    prob <- potential_combinations/total_combinations * 100
    print(paste0("The probability of winning with ", paste(n_int), " numbers is ",
        sprintf("%.8f", prob), " %"))
}

walk(c(3, 4, 5), probability_less_6)
```

```
## [1] "The probability of winning with 3 numbers is 0.10855406 %"
## [1] "The probability of winning with 4 numbers is 0.00707961 %"
## [1] "The probability of winning with 5 numbers is 0.00031465 %"
```

Even if we consider lesser prizes, the probability of winning is still very low, the highest being of the 0.1% with a 3 numbers combination in a single ticket.