

# Guided Project: Mobile App for Lottery Addiction

In this project, we are going to contribute to the development of a mobile app by writing a couple of functions that are mostly focused on calculating probabilities. The app is aimed to both prevent and treat lottery addiction by helping people better estimate their chances of winning.



we'll start by writing two functions that we'll use often:

- A function that calculates factorials; and
- A function that calculates combinations.

## Core function

```
In [124]: def factorial(n):
          final_product = 1
          for i in range(n,0,-1):
              final_product *= i
          return final_product
          def combinations(n,k):
              a = factorial(n)
              b = factorial(n-k)*factorial(k)
              return a/b
```

## One-ticket Probability

```
In [125]: def one_ticket_probability(user_number):
           result = combinations(49,6)
           answer = 1/result
           percent = answer * 100
           print('Your chance to win with the number {} is are {:.7f}% In other wor
d your chance to win is 1 in {:,} to win'.format(user_number,percent,result))
```

```
In [126]: input_1 = [1,2,3,4,5,6]
           one_ticket_probability(input_1)
```

Your chance to win with the number [1, 2, 3, 4, 5, 6] is are 0.0000072% In other word your chance to win is 1 in 13,983,816.0 to win

## Historical Data Check for Canada Lottery

```
In [127]: import pandas as pd
           lottery_canada = pd.read_csv("649.csv")
           lottery_canada.shape
```

Out[127]: (3665, 11)

```
In [128]: lottery_canada.head()
```

Out[128]:

	PRODUCT	DRAW NUMBER	SEQUENCE NUMBER	DRAW DATE	NUMBER DRAWN 1	NUMBER DRAWN 2	NUMBER DRAWN 3	NUMBER DRAWN 4	NUMBER DRAWN 5
0	649	1	0	6/12/1982	3	11	12	14	15
1	649	2	0	6/19/1982	8	33	36	37	38
2	649	3	0	6/26/1982	1	6	23	24	25
3	649	4	0	7/3/1982	3	9	10	13	14
4	649	5	0	7/10/1982	5	14	21	31	32

## Function for Historical Data Check

### Extract winning price

```
In [129]: def extract_number(row):  
           row = row[4:10]  
           row = set(row.values)  
           return row
```

```
In [130]: winning_numbers = lottery_canada.apply(extract_number,axis = 1)  
          winning_numbers.head(5)
```

```
Out[130]: 0    {3, 41, 11, 12, 43, 14}  
          1    {33, 36, 37, 39, 8, 41}  
          2    {1, 6, 39, 23, 24, 27}  
          3    {3, 9, 10, 43, 13, 20}  
          4    {34, 5, 14, 47, 21, 31}  
          dtype: object
```

### Write the function to show all the winning in the past

```
In [131]: def check_historical_occurrence(user_numbers, historical_numbers):  
           set_user = set(user_numbers)  
           check = historical_numbers == set_user  
           n_check = check.sum()  
           if n_check == 0:  
               print('{} has never occurred'.format(user_numbers))  
           else:  
               print('{} had winning for {:.2f} times in the past {}'.format(user_n  
umbers,n_check))
```

```
In [132]: test_input_3 = [33, 36, 37, 39, 8, 41]  
          check_historical_occurrence(test_input_3, winning_numbers)
```

```
[33, 36, 37, 39, 8, 41] had winning for 1.00 times in the past
```

## Multi-ticket Probability

For the first version of the app, users should also be able to find the probability of winning if they play multiple different tickets. For instance, someone might intend to play 15 different tickets and they want to know the probability of winning the big prize.

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.

```
In [133]: def multi_ticket_probability(n):
            all_result = combinations(49,6)
            probability = (1/all_result)*n
            percent = probability*100
            if percent == 1:
                print('Your chance to win is {:.6f}%'.format(percent))
            else:
                print('Your chance to win is {:.6f}%'.format(percent))
```

```
In [134]: test_inputs = [1, 10, 100, 10000, 1000000, 6991908, 13983816]

            for test_input in test_inputs:
                multi_ticket_probability(test_input)
                print('-----') # output delimiter
```

```
Your chance to win is 0.000007%
-----
Your chance to win is 0.000072%
-----
Your chance to win is 0.000715%
-----
Your chance to win is 0.071511%
-----
Your chance to win is 7.151124%
-----
Your chance to win is 50.000000%
-----
Your chance to win is 100.000000%
-----
```

## Less Winning Numbers — Function

In most 6/49 lotteries, there are smaller prizes if a player's ticket match two, three, four, or five of the six numbers drawn. This means that players might be interested in finding out the probability of having two, three, four, or five winning numbers — for the first version of the app, users should be able to find those probabilities.

```
In [135]: def probability_less_6(number):
            combination = combinations(6,number)
            total_chance = combinations(43,6 - number)
            total_number = total_chance * combination
            probability = total_number/combinations(49,6)* 100
            print('Your chance of winning {} number price is {:.6f}%'.format(n
umber,probability))
```

```
In [136]: for test_input in [2, 3, 4, 5]:  
          probability_less_6(test_input)  
          print('-----') # output delimiter
```

Your chance of winning 2 number price is 13.237803%

-----

Your chance of winning 3 number price is 1.765040%

-----

Your chance of winning 4 number price is 0.096862%

-----

Your chance of winning 5 number price is 0.001845%

-----