# Probability of Winning the Lottery

by KariAnn Harjo

## Task 1: Probability of Winning the Grand Prize

To calculate the probability of winning the grand prize, we can use the concept of combinations. The total number of ways to choose 6 numbers out of 51 is given by the combination formula

$C(n,r)=(n!)/(r!(n-r)!)$

where n is the total number of items to choose from, r is the number of items to choose, n! denotes the factorial of n, and C(n,r) is the number of combinations. Thankfully, there's a function for this to reduce human error. This function uses the comb function from the math module to calculate the total number of combinations for selecting 6 numbers out of 51. The probability of winning is then the reciprocal of this total number of combinations, as there is only one specific combination that results in a win.

In [ ]:
```python
from math import comb

#total number of ways to choose 6 numbers out of 51
total_combinations = comb(51, 6)
#one winning combination divided by total number of combinations
probability_of_winning = 1/total_combinations
print(f"Probability of winning the grand prize: {probability_of_winning}")
#probability in percentage (to satisfy my curiosity)
percentage_of_winning = probability_of_winning * 100
formatted_probability = "{:.10f}%".format(percentage_of_winning)

print(f"Probability of winning the grand prize in percentage: {formatted_probability}")
```

```
Probability of winning the grand prize: 5.552637336155554e-08
Probability of winning the grand prize in percentage: 0.0000055526%
```

## Task 2: Probability of Alexandra Winning at Least Once

This program snippet first calculates the probability of winning the lottery for a single ticket. Then, it calculates the total number of tickets Alexandra buys in her lifetime and the total amount of money she spends on these tickets. Finally, as we learned in Chapter 5 of our textbook, it calculates the probability of Alexandra winning the lottery at least once by using the complement of the probability of never winning across all tickets she purchases (Rogel-Salazar, 2023).

```
In [ ]:
#num of years Alex buys a ticket
lifetime=80
#num of days Alex buys a ticket
days_per_year=365
#total number of attempts to win
attempts = lifetime * days_per_year
#calculate the probability of not winning the lottery even once
prob_not_winning_once = (1 - probability_of_winning) ** attempts
#this is the complement of never winning across all attempts
prob_winning_at_least_once = 1 - prob_not_winning_once
percentage_of_winning_at_least_once = round(prob_winning_at_least_once * 100, 10)
formatted_percentage = "{:.10f}%".format(percentage_of_winning)
# assuming each ticket costs $5
money_spent = attempts * 5

print(f"Probability of winning at least once: {prob_winning_at_least_once}")
print(f"Probability of winning at least once in percentage: {formatted_percentage}")
print(f"Total money spent: ${money_spent}")
```

```
Probability of winning at least once: 0.0016200564374688753
Probability of winning at least once in percentage: 0.0000055526%
Total money spent: $146000
```

## Task 3: Calculating Nmin(epsilon) for Various Epsilon Values

```
In [ ]:
#probability of winning one ticket
prob_single_ticket = 1 / comb(51, 6)

#list of epsilon values
epsilons = [10**-5, 10**-3, 0.1, 0.5]

#iterate through each value
for epsilon in epsilons:
    nmin = 0
    while True:
        prob_not_winning = (1 - prob_single_ticket) ** nmin
        #if the probability of at least one win is greater than or equal to epsilon
        if 1 - prob_not_winning >= epsilon:
            print(f"Nmin for epsilon={epsilon}: {nmin}")
            break
        nmin += 1  #increment nmin
```

```
Nmin for epsilon=1e-05: 181
Nmin for epsilon=0.001: 18019
Nmin for epsilon=0.1: 1897486
Nmin for epsilon=0.5: 12483207
```

ϵ=1×10^-5 : This represents a very low probability threshold. The output indicates that you need to purchase at least 181 tickets to have a greater than or equal to 0.00001 chance (or 0.001%) of winning at least once.

ϵ=1×10^-3 : This is a slightly higher probability threshold. To achieve a probability of winning at least once that is greater than or equal to 0.001 (or 0.1%), you need to purchase at least 18,019 tickets.

epsilon = 0.1 : This represents a 10% chance of winning at least once. The calculation shows that purchasing at least 1,897,486 tickets is necessary to have a 10% or greater chance of winning the lottery at least once.

ϵ=0.5 : This is a 50% chance, or a coin flip's chance, of winning at least once. To reach this level of probability, you need to buy at least 12,483,207 tickets.

## Task 4: Increasing Win Probability Above 0.5 Without Adding More People

One strategy Amir and his friends can use is to ensure their tickets do not have any overlapping numbers, thus covering more unique number combinations. However, given the constraints of the lottery game (selecting 6 numbers from 51), this strategy is limited by the number of unique combinations that can be made and may not be feasible for significantly increasing the probability without adding more people.

The script below calculates the minimum number of friends needed to achieve a win probability of at least 0.5 based on the complement rule and the given probability of winning with a single ticket. It then outlines a conceptual approach to increase the win probability above 0.5, emphasizing the practical limitations of this strategy given the rules of the lottery game.

In [ ]:
```
from math import comb, log
```

```
#given data
probability_of_winning_single_ticket = 1 / comb(51, 6)
desired_probability = 0.5

#calculate Nmin using the formula for the complement rule
Nmin = log(desired_probability) / log(1 - probability_of_winning_single_ticket)
Nmin = int(Nmin) + 1

#calculate Nmin for the given desired probability
print(f"Minimum number of friends required to achieve at least a 0.5 probability of winning: {Nmin}")
```

```
Minimum number of friends required to achieve at least a 0.5 probability of winning: 12483207
```

In [ ]:
```
'''
Double checking my work because over 12 million friends seemed a little absurd.
'''
from math import comb, log

#calculate the probability of winning with one ticket
p = 1 / comb(51, 6)

#calculate the minimum number of friends required to achieve at least a 0.5 probability of winning
Nmin = log(0.5) / log(1 - p)
#checking that N satisfies the condition
Nmin = int(Nmin) + (Nmin > int(Nmin))
print(f"Minimum number of friends required to achieve at least a 0.5 probability of winning: {Nmin}")
```

```
Minimum number of friends required to achieve at least a 0.5 probability of winning: 12483207
```

## Task 5: Probability of Winning the Consolation Prize

In [ ]:
```
#number of ways to choose 3 winning numbers
ways_to_choose_3_correct = comb(6, 3)
#number of ways to choose the remaining 3 numbers out of the 45 non-winning numbers
ways_to_choose_3_incorrect = comb(45, 3)
#total number of ways to choose 6 numbers out of 51
total_combinations = comb(51, 6)
#probability of winning the consolation prize
probability = (ways_to_choose_3_correct * ways_to_choose_3_incorrect) / total_combinations
# Expected value is the probability of winning multiplied by the prize amount (100)
expected_value = probability * 100
percentage = "{:.10f}%".format(probability * 100)
```

```
print(f"Probability of winning the consolation prize: {probability}")
print(f"Probability of winning the consolation prize in percentage: {percentage}")
print(f"Expected value of the consolation prize: ${expected_value}")
```

```
Probability of winning the consolation prize: 0.015758384760009462
Probability of winning the consolation prize in percentage: 1.5758384760%
Expected value of the consolation prize: $1.5758384760009463
```

This program first calculates the probability of winning the consolation prize by considering the combinatorial aspects of choosing 3 correct and 3 incorrect numbers. Then, it calculates the expected value of winning this prize, assuming a specific prize amount. We can theorize that the introduction of a consolation prize could increase sales by making the lottery more attractive to potential players, offering them more opportunities to win something, thereby enhancing the *perceived* value of buying a ticket.

## Sources:

OpenAI. (2023). ChatGPT interaction [Personal communication].

Hartman, G., et al. Apex Calculus. (n.d.). Epsilon-Delta Definition of a Limit. In Calculus 3e. LibreTexts. Retrieved from https://math.libretexts.org/Bookshelves/Calculus/Calculus_3e_(Apex)/01%3A_Limits/1.02%3A_Epsilon-Delta_Definition_of_a_Limit

Hayes, A., Kesa, S., User 123, et al. (n.d.). Probability by complement. Brilliant. Retrieved from https://brilliant.org/wiki/probability-by-complement/

Khan Academy. (n.d.). Combination formula. Retrieved from https://www.khanacademy.org/math/precalculus/x9e81a4f98389efdf:prob-comb/x9e81a4f98389efdf:combinations/v/combination-formula#:~:text=The%20formula%20for%20combinations%2C%20also,*%20(n%2Dr)!

Rogel-Salazar, J. (2023). Definitely Maybe: Probability and Distributions. In Statistics and data visualisation with Python (1st ed.). CRC Press.