**Introduction:**

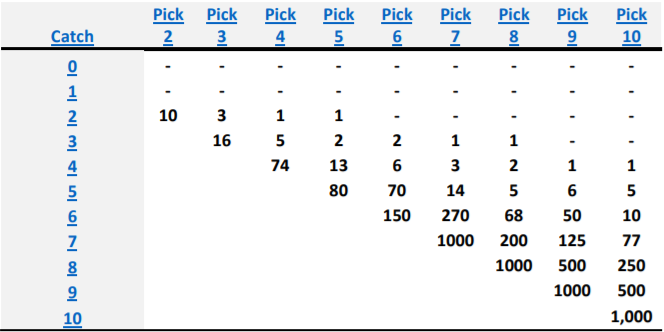
The analysis focuses on Caveman Keno game which is a variation of traditional Keno Game. It is a popular game which is played online or in gambling casinos extensively. This variation of Keno adds drawing of 3 numbers by the computer which become dinosaur eggs. If two or three of the eggs match among the 20 numbers drawn, the payouts are then multiplied. The analysis of this project is application of Hypergeometric distribution in a real-world game.

It has the following rules:

* The player makes a bet and chooses 2 to 10 numbers from 1 to 80.
* The computer randomly picks three of the unpicked numbers and marks them with eggs.
* The computer randomly picks 20 numbers from 1 to 80.
* The player's base prize will pay according to the balls matched with the numbers picked by the computer.
* If two eggs match, then the win is multiplied by 4
* If all 3 eggs match, then the win is multiplied by 8.
* The final award is the product of the base prize and multiplier.

**Analysis:**

We have been provided with following base prizes



The pay table is based if it costs 1$ for each turn. The prizes seem to be intimidating as a person could win 1000$ after inserting a dollar. The amounts would be further increased if bonus eggs are matched. Let us calculate the winning odds to see the probability of winning.

We need to find the probability of pay for which we need to use Hypergeometric distribution.

Hypergeometric distribution is a discrete probability distribution that describes the probability of k successes in n draws, without replacement, from a finite population of size N that contains exactly K objects with that feature, wherein each draw is either a success or a failure [2].

To find P(X), we use HYPGEOM.DIST(sample,number\_sample,population\_s,number\_pop,cumulative)

We need to put values of catch, pick, winning numbers, population and cumulative.

Where winning numbers = 20 and population = 80.

To calculate bonus, we need to use the Hypergeometric distribution again.

P(B) = HYPGEOM.DIST(matched eggs,20-catch, population\_s, 80-pick, cumulative) \* P(X)

where matched eggs are 2 or 3 and accordingly we could get 3x or 6x multiplier bonus and population\_s is 3.

We need to find following parameters:

1. **Average RTP% (percentage Return to player) of Keno**

This value is the return to the player if the player plays it on the long run. It is the amount that is the expected win per game and per dollar spent, excluding the bonus wins.

We make use of HYPGEOM.DIST(sample,number\_sample,population\_s,number\_pop,cumulative) to calculate P(X). Then Sum product of P(X) and pay to calculate the mean which is the average RTP%.

We get the following values:



Hence, we can see that if we play the game on the long run, for every dollar the return is from 60 to 68 cents excluding the bonus wins.

1. **Hit Frequency of Keno**

This is the percentage of time that players win a hand. It is the sum of all the probabilities whose pay is more than 0.

We get the following values:



We can see pick 9 has the highest sum of 39.92 percent.

1. **Pulls per Hit of Keno**

It is the percentage of time that players win a hand. We find it by using

Bonus RTP % = ROUND {1/ hit frequency of keno, 1}



So, we can see that a player with pick 2 has the highest percentage to win a hand.

1. **Bonus Hit Frequency**

This is the percentage of time that players win a bonus. It is the sum of all P(X) for the bonus. It is calculated for 3x and 6x and then added to give total bonus hit frequency.

We get the following results.



1. **Bonus Pulls per Hit**

It is the average number of times a player has to play before he/she wins a bonus.

Pulls per hit= ROUND {1/ hit frequency of keno, 1}

We need to find it for 3x, 6x bonus and then add them.



We can see that on an average a player has to play a lot of games to win any bonus. The best case is for pick 4 where the player has to play around 20 times to win a bonus.

1. **Average RTP% from Bonus**

It is the expected bonus win per game We need to calculate using the Hypergeometric distribution again.

P(B) = HYPGEOM.DIST(matched eggs,20-catch, population\_s, 80-pick, cumulative) \* P(X)

Then Sum product of P(X) and pay to calculate the mean which is the bonus hit frequency.

By using the Formula, we get the following result

This is the sum of 3x and 6x bonus percentage.



1. **Total RTP% of Game**

It is the sum of the RTP% from the base game and the RTP% from the bonus. We need to add both average RTP% of keno and bonus to calculate the Total RTP%.



We can see that on an average total RTP is around 80 percent which means on the long run for every dollar we put we will get around 80 cents back.

1. **Volatility index of the game**

Volatility = standard deviation \* Norm.s.inv(0.975)

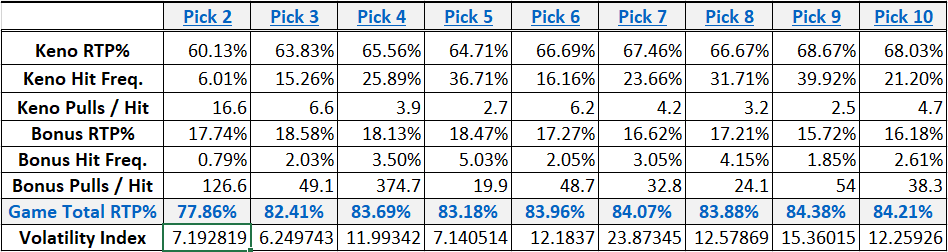
Standard deviation is square root of variance

Where Variance is X^2\*P(X) - mean ^ 2

We get the following results



1. **A Summary table**

****

**Conclusion:**

We have learned about the application of hypergeometric distribution and an interesting game caveman keno. We can see that it is a game of chance but a careful selection of number of picks can make a huge difference. Pick 9 has the highest percentage of total RTP which is 84.38 which means we get back around 84 cents back on every dollar spent. However, pick 4 has the best chance to get a bonus. Pick 4 has the total RTP % of around 83.69 which is not a bad return so pick 4 would be the best choice to play. The pay table is based if it costs 1$ for each turn. The prizes seem to be intimidating as a person could win 1000$ after inserting a dollar and then a multiplier of 6x which means he/she could win 6000$ by inserting a dollar but after doing the calculations we can understand the chances are almost negligible. It seems intimidating but on the long run we can see that the computer always wins and we mostly get around 80 cents back on every dollar we spent.

**References:**

1. Caveman Keno Plus. (n.d.), from <https://wizardofodds.com/games/keno/caveman-keno-plus/>
2. Hypergeometric distribution. (2017, October 20). Retrieved November 22, 2017, from https://en.wikipedia.org/wiki/Hypergeometric\_distribution