THE GAMBLER'S RUIN PROBLEM
Introducing Games of Chance

Thesis to be submitted in partial fulfillment of the requirements for the degree

of

BSc. Statistics Hons.

by

Somjit Roy
Roll No. : 18-300-4-07-0402
Registration No. : A01-1112-0761-18

Under the guidance of

Prof. Pallabi Ghosh

DEPARTMENT OF STATISTICS

ST.XAVIER'S COLLEGE (AUTONOMOUS), KOLKATA

Department Of Statistics
St.Xavier's College (Autonomous), Kolkata
India - 700016

DECLARATION

I affirm that I identify all my sources and that no part of my dissertation paper uses unacknowledged matrials.

Somjit Roy

Somjit Roy

Department Of Statistics
SXC, Kolkata

Date: May, 2021

ACKNOWLEDGEMENTS

During the course of my research, I have received a great deal of support and help.
I would like to thank and acknowledge my supervisor **Prof. Pallabi Ghosh** for her immense support and involvement, when it comes to the formulation of my topic and sharing her resourceful insights.
I would also like to take a moment to acknowledge my other professors in the Department of Statistics at St.Xavier's College, Kolkata, **Prof. Dr. Ayan Chandra**, **Prof. Debjit Sengupta**, **Prof. Dr. Surupa Chakraborty**, **Prof. Dr. Surabhi Dasgupta**, **Prof. Madhura Dasgupta** & **Prof. Dr. Durba Bhattacharya**, who all have helped me to develop the mindset prone to research, which has made it possible for me to complete the project.
It was my pleasure to seek guidance from **Prof. Subhamoy Maitra**, Indian Statistical Institute, Kolkata, of the Applied Statistics Unit (ASU), on the topic of my research.
It would be injustice to not mention the support and help I had received from my peer group, as they presented their ideas, which was duly put to use in my research work.
Lastly, a big thanks goes to my parents for their supporting nature and constant guidance throughout my undergraduate course.

Somjit Roy
SXC Kolkata
Date: May, 2021

ABSTRACT

The Gambler's Ruin Problem, opens a wide avenue of applications to many of the advanced techniques and methodologies of Statistics, Probability, Mathematics and Computer Science. From Markov Chains & Random Walks to Monte Carlo Simulations & user-built algorithms used to develop the structural framework of the Gambler's Ruin Problem. The problem at hand does not only exploits and discovers the theoretical aspects, but also functions on a practical level, being readily used in modelling Stock-Market scenarios, formulation of insurances, illustrating web search algorithms, compartmental models in epidemiology (SIR & SIS model), etc.

This paper comprises of a detailed analysis of the problem, performed at distinct levels, which includes formulation of the problem, solution of the problem, simulation of various setups of the problem, applying the problem to real world scenarios, etc.

Keywords: **GRP** - Gambler's Ruin Problem, **MC** - Markov Chains, **t.p.m** - Transition Probability Matrices, Stochastic Processes, Random Walks, Absorbed States, Game of Chance, Unbiased, Biased.

Contents

List of Figures

List of Tables

Aspects of Simulation, Softwares Used, List Of Packages & Functions Used

1. Software Used :: **R (4.0.4)** [3] & **Tableau Public** [4].

2. Note that, every simulation performed in this paper, has been assigned a seed value of **'987654321'**, i.e., **'set.seed(987654321)'**, for maintaining uniformity of results obtained.

3. The ***User - Built*** Functions used in the paper for simulation are as follows ::

   - ***grp.win.prob(.....)***
   - ***grp.money.trajectory(.....)***
   - ***gamblers.ruin.expected.rounds(.....)***
   - ***rounds.vs.initial.sum(.....)***
   - ***bet.colors(.....)***
   - ***expected.return(.....)***

4. For ***Graphics & Representation Of Data***, the software ***Tableau Public***, along with some functions & packages in ***R (4.0.4)***, is used, which are listed as follows ::

   - For Graphical Representation :: ***ggplot2*** [5], ***viridis*** [6], ***hrbrthemes*** [7], ***gridExtra*** [8].
   - For Data Handling and Representation :: ***tidyverse*** [9], ***tidyr*** [10], ***dplyr*** [11].
   - For Transition Diagrams :: ***markovchain*** [2].

1 Introducing Randomness & Chance Games

1.1 *Randomness - A Part Of our Daily Life!*

Randomness, is a part of our daily life. The idea of randomness plays a very big role in reality. From *"rainfall on a particular day"* to *"the clothes, we wish to wear"*, *"from the marks we get in an examination"* to *"the cell phone number, which we use"*, everything has an element of uncertainty or randomness associated to it.
Such randomness gives rise to a whole branch of Statistics, which we know as ***Probability***. Probability apart from being the silver lining in the world of uncertainty, it also comprises of many applications to the real world phenomena (taking place in our day to day life), posing it as theoretical ideologies and challenge in front of the readers.

» Starting with a Real Life Example!

It is customary in our daily lives to pay our electricity bills within a stipulated time. But have we ever thought that the ***units which our apartments consume every month is entirely random even though we have a rough idea or estimate about our consumption pattern.*** The adjacent diagram (Figure 1) illustrates the units of electricity consumed in my own apartment for the year ***2018 (April) to 2021 (January)***, illustrating a seasonal time series data [12].

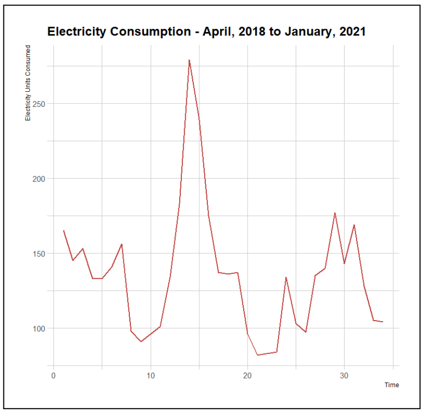

Figure 1: ***A Time Series Representation Of Randomness***

» The Idea to take note of from the above Example!

Speaking about uncertainty in the electricity consumption, the units which we consume monthly or yearly (as the case may be) can never be predetermined or fixed even if we keep a goal and restrict our electricity consumption. This is the juncture where Randomness comes into picture.

But does knowing about the presence of uncertainty in any event seldom helps? Not necessarily, we should be primarily having an idea that ***"How to model this element of randomness?"***. This is a very important area to focus and ***Probability Distributions*** is a great help in this venture.

1.2 *Beginning with Games Of Chance*

Remembering Childhood Games! - It's quite nostalgic to think of those days, when we all used to play "LUDO", "SNAKES and LADDERS", "MONOPOLY", and what not. But little did we think of the fact that these games add to a higher dimension to the theory behind Statistics (especially Probability) apart from the fun and entertainment which it used to provide.

These are a few examples of ***Games of Chance***.

» But Why Game Of Chance?

To each of these games mentioned above, we have an element of probability or chance associated to it, i.e., every move in these games are decided either by ***rolling a die or tossing a coin***. Whether we take a step forward or a step backward, each step has a associated probabilistic viewpoint to it.

Figure 2: ***A Game Of Monopoly***

Following the usual scenario of ***fair games***, in case of rolling a die and making a move, we have an associated probability of $\frac{1}{6}$ and in case of tossing a coin, the probability becomes $\frac{1}{2}$.

Games having an associated Chance Factor, are not only restricted to Board Games and used solely for the purpose of entertainment and enjoyment, but gives rise to an entire branch of theory viz. ***Game Theory, different strategies regarding distinct game plays, usage of hardcore probability and it's offshoots, etc.***

2 Background of the Gambler's Ruin Problem - An Historical Introduction

The first origin of the problem was from a letter given by ***Blaise Pascal*** to ***Pierre De Fermat*** in ***1656*** [13]. In that very year ***Pierre De Carcavi*** in his letter to ***Huygens***, modified the version of the problem as given by Pascal. The problem posed to Huygens was as follows:

*"Let there be two players (A & B) playing a game with three dice. Player A wins a point if **11** is thrown on the three dice and Player B wins a point if 14 turns up. But in lieu of the normal accumulation of the points, a point shall be added to the player's score if his/her opponent is having 0 points otherwise it would be subtracted from his /her opponent's score. The Player who scores 12 points first, is the winner.*
The question is, what are the relative chances of each player winning the game?" [14]

Huygens's on the other hand once again reformulated the problem:

*"Each player starts with **12** points, and a successful roll of the three dice (11 for Player A and 14 for Player B) adds one point to the score of the winning player and subtracts a point from the opponent's score.*
What is the probability of winning the game for each player, if the loser of this particular game is the first one to reach zero points?" [15]

The above two problem statements as proposed by ***Carcavi*** and ***Huygens*** respectively, are the derivative of what is known today widely as the ***Gambler's Ruin Problem***, one of the most frequently visited problems in the world of ***Games of Chance***.

3 Entering into Gambler's Ruin - A Game all about Chances

3.1 *The World Of Gambling alongside Gambler's Ruin*

We all have an idea about how the gambling universe functions. But have we ever thought about the Statistics behind it? Yes, we have and that is the reason of having such elegant gambling theories which encompasses different problems at hand and their solutions, one of which is the ***"The Gambler's Ruin Problem"***.

Figure 3: ***A Game of Cards***

Suppose you play a game with some initial capital at stake, you would primarily be motivated to maximise your own worth, i.e., the amount of money which you are having. But is that guaranteed? Of course not, it may so happen that you loose the entire capital or you make almost 10 times the initial money. It's all about the chance factor coming into play and the Gambler's Ruin problem helps us to formulate or model scenarios like these.

3.2 *A Short Glimpse of the Gambler's Ruin Problem!*

A Real world Problem - Suppose with an amount of **Rs. 100**, you enter into a game and plan to increase upon the sum which you have. You have no upper limit, i.e., the more the merrier, but if you loose all the money you eventually have to stop. So you start playing. You win a particular round of the game with probability 'p' and loose it with probability '1-p', on winning you add 1 rupee to your existing sum and on losing you have to give away

1 rupee from your existing sum. What do you think? Will you get richer or will you loose everything? Let's see.

- ***AFTER 1000 ROUNDS OF THE ABOVE GAME - (p = 0.5)***

After playing for 1000 rounds with a probability of 0.5 of winning each round (i.e., a ***Fair Game***), it can be observed that you add up to the amount of money with which you initially started the game, and maximise the sum somewhere just before the 750th round of the game in these 1000 rounds of game play. ***The Sum of Money at each round of the game with you*** has been plotted along the ***y-axis*** and the ***Game Rounds*** are along the ***x-axis***.

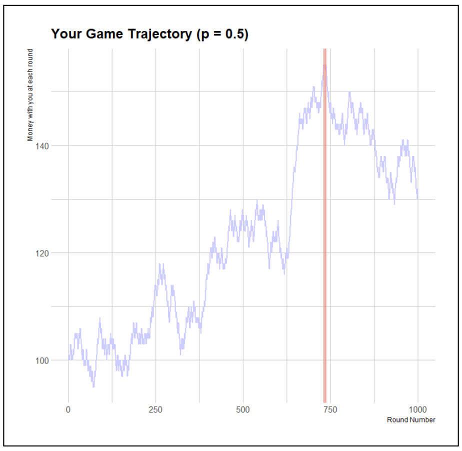

Figure 4: ***A Fair Game Simulation, p = 0.5***

But did the game stop? Not yet. The Game carried until 14679 rounds, i.e., the expected rounds of this fair game to be played until you get broke with an initial amount of Rs. 100 is ***14679***.

Now we could have simulated the above game with different probabilities of winning in each round, making the game eventually ***biased***.

Therefore, we have provided an informal introduction to the Gambler's Ruin Problem.

Now let's see what happens when the Game is Biased.

A Game can be biased in two ways, as follows:-

- ***Biased towards the player***, with a higher win probability at each round.
- ***Biased against the player***, with a lower win probability at each round.

Following are the Simulations of two games, where one is biased towards the player and the other is biased against the player.

- ***SIMULATING A BIASED GAME TOWARDS THE PLAYER - (p = 0.65)***

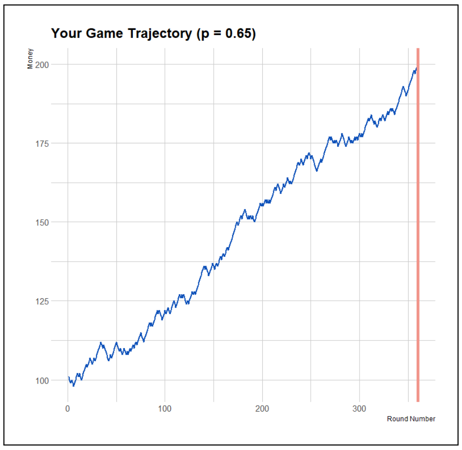

Figure 5: ***A Biased Game towards the player, p = 0.65***

We have set an upper limit of Rs. 200, i.e., as soon as the player reaches to Rs. 200, he or she leaves the game.

It can be observed that the win probability for each round has now been increased to 0.65 from 0.50, making the player to reach the upper limit of the game Rs. 200, somewhere after the 350th round of the game, even though the game has been simulated for 1000 rounds, which is quite natural as the probability of winning each round is quite high in this case.

- ***SIMULATING A BIASED GAME AGAINST THE PLAYER - (p = 0.45)***

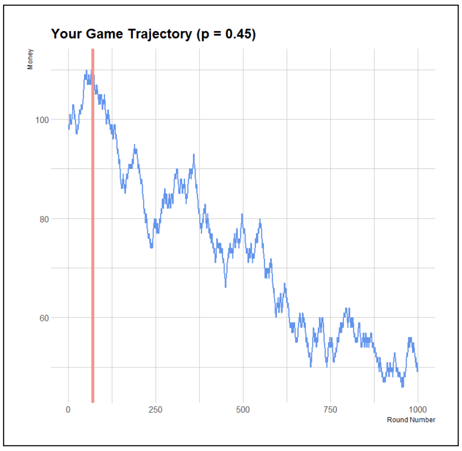

Figure 6: ***A Biased Game against the player, p = 0.45***

On the other hand, if the game is simulated with a lower win probability for each round, the player ultimately gets broke at a certain point of time in the game play (where the maximum sum is represented by the reference line, in between rounds 0 and 100).

4 The Problem Statement & Setup

In the previous sections we have introduced our problem of interest and had given more of an informal definition, illustrating the reality of the problem at hand through some simple simulations.

4.1 *The Problem Statement*

Now we pose the formal problem statement in general, which is as follows:

Let there be two Gamblers (players) A and B. Player A starts the game with an initial amount of Rs.n_1 and Player B starts the game with an initial amount of Rs.n_2, such that the total amount of money available in the game is Rs.($n_1 + n_2$).
Suppose the probability of winning a particular round of the game for A is 'p' and that for B is 'q = 1-p'. When A wins a round, Rs.1 is received by A from B and when B wins, Rs.1 is received by B from A. The game stops when one of the player goes bankrupt and the other wins all the money, i.e., Rs.($n_1 + n_2$). Then what is the probability of A or B winning the game? [16]

Thus, we have the ***Gambler's Ruin Problem***.
Note that, here we will mainly discuss in context of player A, and the results for player B will just follow from that.

4.2 *The Setup of the Problem*

We illustrate the Setup of the problem through a schematic representation (which will be given a more formal term in the later half of the paper). The setup of the Game has been given with regards to Player A.

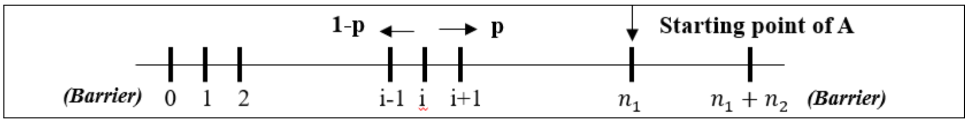

Figure 7: ***The Schematic Representation of the Gambler's Ruin Problem***

5 The Backbone of Gambler's Ruin - A Few Concepts related to the Problem at hand

Before diving into the core of the problem stated above, it is necessary to specify few among many concepts and ideas which forms the backbone of the Gambler's Ruin Problem. They are :-

- ***Stochastic Processes***
- ***Random Walks*** and the associated ***Absorbed States***.
- ***Markov Chains*** *and* ***Transition Probabilities***.

The above mentioned ideologies are the most important ones which could elegantly explain the core of the problem under consideration.

5.1 *Stochastic Processes*

» ***Defn 1*** : A collection or sequence of ***Random Variables (finite or infinite)*** is referred to as a ***Stochastic Process***.

It is generally denoted by $\{X_t\}_{t \in T}$, where T is a subset of $[0, \infty)$. [17]

» When T $= \mathbb{N}$ (or N_0), then $\{X_t\}_{t \in T}$ is called ***A Discrete-Time Stochastic Process***.

» Now if T $= [0, \infty)$, then $\{X_t\}_{t \in T}$ is called ***A Continuous-Time Stochastic Process***.

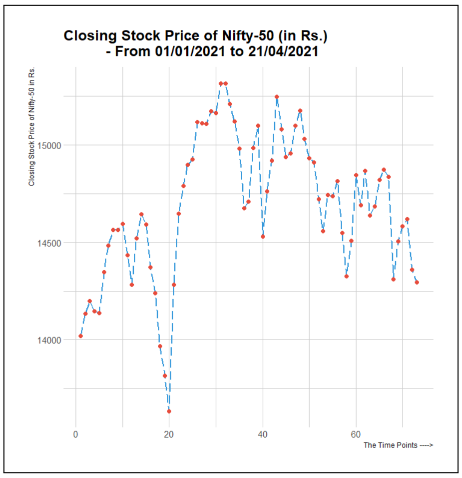

Figure 8: ***An Example Of Stochastic Process - Stock Market Prices***

» ***A Simple Illustration Of Stochastic Process***

The adjacent Figure (8) illustrates the movement of the ***Closing Price***s of the Stock ***Nifty-50*** over January 1, 2021 to April 21, 2021. The source of the data being **National Stock Exchange** [18].

This is purely an example of ***Discrete-Time Stochastic Process***, where $\{X_t\}_{t \in T}$ represents the collection of ***Continuous Random Variables*** denoting the closing price of the stock Nifty-50 at the 't' th time point, and the 't' is here a natural number, since we observe the closing price of the stock at the end of each day starting from 01st January, 2021 to 04th April, 2021.

5.2 *Random Walks*

[***A Beautiful illustration of Random Walk - Two Dimensional, using the movements of Gas Molecules, when they are in motion have been made at this link***]

[***Click here*** - http://web.mit.edu/8.334/www/grades/projects/projects17/OscarMickelin/randomWalk.gif]

An offshoot of Stochastic Processes is ***Random Walks***. It basically gives us an idea about the **trajectory** or the **path** followed by a Stochastic Process. [19]

» ***Defn 2*** : Suppose $\{\epsilon_t\}$ is a purely random process with mean 0 and variance ${\sigma^2}_\epsilon$.

- The Process (X_t) is said to be ***Random Walk***, if $\boxed{X_t = X_{t-1} + \epsilon_t}$.
- The above defined Process has an initial state of 0, i.e., customarily starts at 0 $\Rightarrow$ $X_0 = 0$.
- Therefore, we have $X_1 = \epsilon_1, X_2 = \epsilon_1 + \epsilon_2$, ... and so on. In general we have, $X_t = \sum_{i=1}^{t} \epsilon_i$.
- Random Walks can be both **Biased** as well as ***Unbiased***, i.e., if you toss a fair coin, the sequence Of heads or tails generated in those repeated tosses is an example of an ***Unbiased (Symmetric) Random Walk***.

  Whereas, on the other hand tossing a coin where probability of heads in each toss is p ($\neq \frac{1}{2}$) is an example of ***Biased Random Walk***. [20]
- The Gambler's Ruin Problem can be elegantly modelled through Random Walks.
- The Gambler's Ruin Problem with a Single Player is an example of an ***One-Dimensional Random Walk***, there may be the case of a ***N-dimensional Random Walk***, typically

the ***N-Player Ruin Game***, i.e., the Gambler's Ruin Set up with N-Players.

We will be considering the case of ***N = 2***.

» Defn 3 : The Absorbed States in a Random Walk

The Absorbed State in a Random Walk is that particular point, where the Walk comes to an end or stops. This stopping point of the random walks are also known as the ***Barrier Points***.

5.3 *Illustrating Examples Of Random Walks*

- ***EXAMPLE 1 - The Biased Random Walk in Gambler's Ruin Setup with p = 0.45***

We have already simulated a biased game (i.e., with p = 0.45), which has been diagrammatically represented in Figure **6** on page **8**. This can be considered as an example of a ***Biased Random Walk***, which is itself a ***Stochastic Process***.

To frame in terms of the above stated definition, here $X_0 = 100$ instead of 0, since we have started the game with an initial stake of Rs.100. Also, we have,

$$\Rightarrow \epsilon_t = \begin{cases} 1 & ; \text{ w.p } 0.45 \\ -1 & ; \text{ w.p } 0.55 \end{cases} \tag{1}$$

- ***EXAMPLE 2 - Movements Of A Drunkard***

In this illustrative example of Random Walks, we try to simulate the footsteps taken, or the movements of a Drunkard, i.e., the path (trajectory) followed by a person after he or she gets drunk. It's quite unnatural, but to people's surprise, this is also an example of a Random Walk.

» The General Algorithm Of the Movement

We consider that the drunkard under consideration starts or begins his journey at some integer point (Say 0), i.e., the drunkard begins his journey at the ***Origin (0,0)***, if we consider the two-dimensional x-y plane, where the ***x-axis*** would denote the ***number of Steps taken by the Drunkard*** and the ***y-axis*** would define the ***current position of the Drunkard***.

He takes a step forward or a step backward only with ***equal probability***, i.e., $\frac{1}{2}$, where his position is indicated by the numbers adjacent to the starting point integer. When he takes a step forward, he moves to the higher adjacent integer (say from ***y*** to ***y+1***) and moves to the lower adjacent integer on taking a step backward (say from ***y*** to ***y-1***). Let us visualize the Random Walk.

» The Simulation - Visualizing the Random Walk

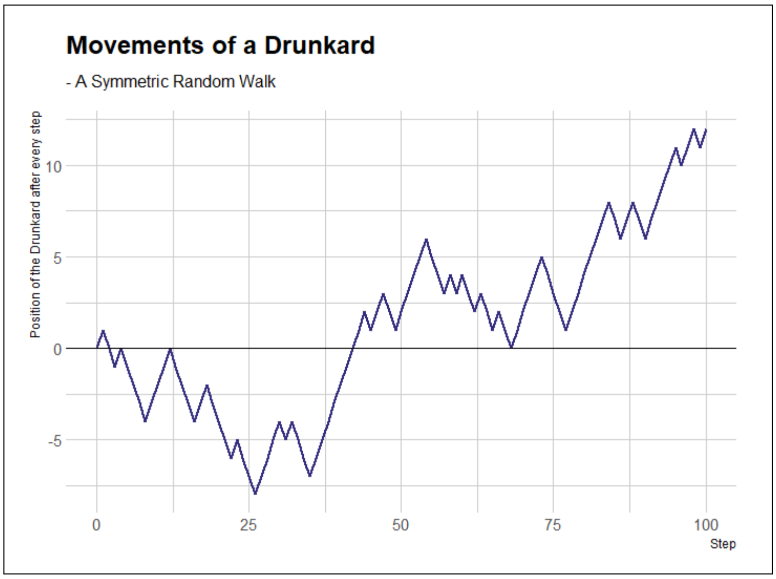

Figure 9: ***An Illustration Of One-Dimensional Random Walk***

The diagram above (Figure **9**) illustrates a Random Walk which is of Symmetric Nature ***(Unbiased)*** as we have hypothesized that the Drunkard takes a step forward and backward with ***equal probability***, i.e., $\frac{1}{2}$.

» Formulation Of the Example

Let X_n be a random variable denoting the direction in which the drunk person moves at the n^{th} step, i.e.,

$$\Rightarrow X_n = \begin{cases} 1 & ; \text{ w.p } 0.5 \\ -1 & ; \text{ w.p } 0.5 \end{cases} \tag{2}$$

Hence $S_n = \sum_{i=1}^{n} X_i$ is the position of the drunkard after n steps, where $X_0 = 0$. Therefore, S_n gives us the Random Walk displayed above. Here n = 100 steps, i,e., the movement of the Drunkard has been simulated only for 100 steps. For further variants of the above problem, refer [21].

5.4 *Markov Chains (MC) & Related Terminologies*

Another major offshoot of Stochastic Processes is the idea of ***Markov Chains***. These chains are used to ***model the evolution of any non-deterministic (Stochastic) System***.

» ***Defn 4*** : ***A General Definition of Markov Chains***.

Consider a system that can be in any one of a finite or countably infinite number of states. Let ζ denote this set of state(s). It can be assumed that ζ is a set of integers. Here ζ is referred to as the ***State Space*** *of the system. If the system is observed at discrete time points n = 0, 1, 2, ... and let X_n denote the state of the system at time point n.* [19]

This system denoted by X_n on satisfying the **Markovian Property** is regarded as a **Markov Chain**.

The Markov Chains are generally used to study and analyze the ***evolution of a process or a system***.

» ***Defn 5*** : ***What is the Markov's or the Markovian Property?***.

The Markovian Property states that,

Given the information about the present state of a system, the past and future are ***conditionally independent***.

Therefore, a Stochastic Process $\{X_n\}$ is said to be a ***Markov Chain***, if for j, k, j_1,..., $j_{n-1} \in \mathbb{N}$, we have,

$$\boxed{\Rightarrow P(X_n = k|X_{n-1} = j, X_{n-2} = j_1, ..., X_1 = j_{n-1}) = P(X_n = k|X_{n-1} = j)} \quad (3)$$

$$\boxed{\Rightarrow P(X_n = k|X_{n-1} = j, X_{n-2} = j_1, ..., X_1 = j_{n-1}) = {p_{jk}}^{n-1,n}} \quad (4)$$

» ***Defn 6*** : ***Transition Probability***

The Probability of moving from one state space to the other, i.e., the probability for a system to make a transition from one state space to the other is known as the ***Transition Probabilities*** *for the system*. [22]

Specifically, ${p_{jk}}^{n-1,n}$ - is the probability of the system making a transition to state 'k' at the time point 'n', when it was initially at state 'j' at the time point 'n-1'.

» ***Defn 7*** : ***Transition Probability Matrices (t.p.m)***

The Transition Probability Matrix, often abbreviated as t.p.m, is the collection of the Transition Probabilities in a matrix form, where the ***rows denote the current state at***

which the system is *and the* ***columns denote the future state to which the system would reach.***

» ***<u>Defn 8</u>*** : ***Transient & Recurrent States***

Let $\{X_n\}$*,* $n \geq 0$ *be a Markov Chain, over the State Space* $\boldsymbol{\zeta}$*, then as mentioned in* ***Defn 6*** *of Transition Probability '*p_{jk}*' is the probability that the Markov Chain initially at state j, will be in state k, as result of which State j is referred to as the* ***Transient State****, if '*$p_{jj} < 1$*', i.e., the Markov Chain has a positive probability of never returning to state j, when initially starting at that particular state itself.*
Likewise, State j is regarded as a ***Recurrent State****, if the probability of never returning to that state is 0, when the Chain initially begins at that state, i.e., '*$p_{jj} = 1$*'.* [19]

Note that : An Absorbed State, i.e., a state where the transition of the Markov Chain stops, for example the state in a Gambler's Ruin Setup, where the gambler loses all his capital or wins the game, referring to the absorbed state or the barrier point where the game stops, is necessarily ***Recurrent***, i.e., the probability of returning to that state, when starting at that state is 1.

» ***<u>Defn 9</u>*** : ***Transition Diagrams***

The Diagrammatic Representation Of a Markov Chain is regarded as the ***Transition diagram*** *of that particular Markov Chain. [23]*
The Diagram illustrates the transitions of the 'MC' from one state to the another along with the Transition Probabilities.

A Transition Diagram has been illustrated in the Setup of the Gambler's Ruin Problem, **Figure 7**.

5.5 *An Example Of Markov Chain*

Following is an illustration of the movements of an ant along the two dimensional plane, i.e., $\mathbb{R}^2$, in regards to an example of Markov Chain. [24]

The Ant Trajectory

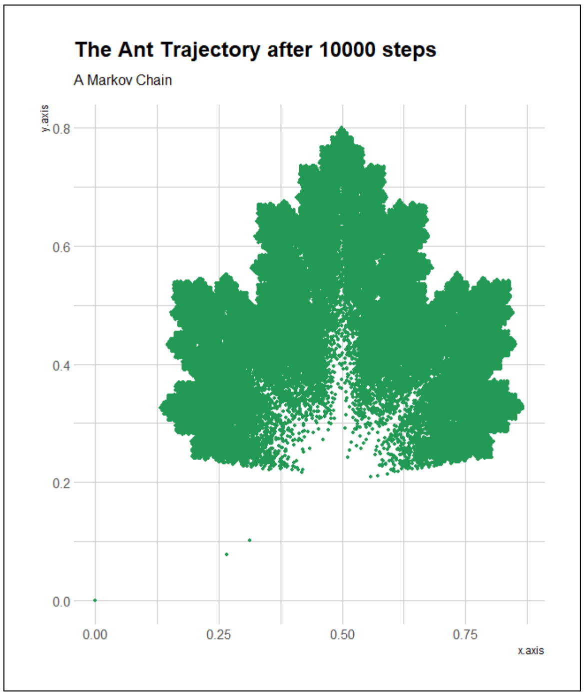

Figure 10: ***Simulating the Ant Trajectory for 10000 steps of the ant***

» *The Algorithm & Motivation*

Consider the 'two dimensional plane' ($\mathbb{R}^2$) and an ant currently at the position (0,0) - the origin, i.e., the ***current state*** of the ant is defined by the coordinates (0,0). We determine the next position of the ant through certain formulae eventually determining the (x,y) coordinates in the two dimensional plane which is the path of travel or movement for the ant. Let us denote the formulae by "F_1, F_2, F_3 and F_4", and each of these formula is chosen with equal probability, i.e., $\frac{1}{4}$ and then the (x,y) coordinates are determined irrespective of the previous position of the ant.

Can this trajectory be regarded as a Markov Chain? Yes, the above path/trajectory followed by the ant is a Markov Chain. Let us suppose that the four formulae mentioned above as F_1, F_2, F_3 and F_4 be the ***four corresponding states***, and where the ant can be in among these states respectively, determines the position of the ant in the $\mathbb{R}^2$ plane.

» What are the things to take note of ?

- Based on the current position of the ant, we determine the future position by selecting any of the four formulae, and determining the (x,y) coordinates in the two dimensional plane. Note that ***the position previous to the current position of the ant does not affect the determination of the future position***. Therefore, the trajectory of the system (i.e., the movement of the ant along $\mathbb{R}^2$) satisfies the ***Markovian Property***, which serves as the theoretical backbone of Markov Chains.
- Note that the values of the (x,y) coordinates in the $\mathbb{R}^2$ plane here, forms the ***State Space*** of the above defined Markov Chain.
- Also note that, the Ant Trajectory under consideration is a case of ***Continuous Space Markov Chain***, as we observe the position of the ant at any desired time point.

» The Ant Trajectory as a beautiful illustration of Statistical Regularity

Apart from being an example of Markov Chain, The Ant Trajectory can also be viewed upon as an illustration to the concept of Statistical Regularity. In other words, the path of the ant is being decided at random or stochastically by selecting any one of the four formulae available. This associated randomness gives a particular "***leaf-like pattern***", when the path is being simulated for a large number of times i.e., the randomness gets diluted in the long run, yielding an observable pattern.

6 Modelling Gambler's Ruin Problem through Random Walks and Markov Chains

In developing the Gambler's Ruin Problem through Random Walks as well as Markov Chains, we consider the usual setup of the problem as stated in Section **4.1**.

6.1 *Gambler's Ruin Problem as One Dimensional Random Walk*

In the two-player setup, comprising of Players A & B, Player A is assumed to enter the game initially with Rs.n_1. The probability that A will win a particular round of the game is p.

Let X_n be a random variable denoting the amount of money with Player A at the n^{th} Gamble, or n^{th} round of the game.
Therefore, we have,

$$\Rightarrow X_n = n_1 + \epsilon_1 + \epsilon_2 + ... + \epsilon_n, n \geq 1 \tag{5}$$

Here, $X_0 = n_1$ and $\{\epsilon_t\}_{t\geq 1}$ - is a sequence of Independently and Identically Distributed (IID) Random Variables, defined as follows,

$$\Rightarrow \epsilon_t = \begin{cases} 1 & ;\ \text{w.p. } p \\ -1 & ;\ \text{w.p. } q = 1 - p \end{cases}$$

Hence, we have the One-Dimensional Random Walk.

6.2 *The Two Absorbed States in the Gambler's Ruin*

Note that, there are two points where the Game being played by Player A will stop - ***two absorbed states***, i.e., the Gambler's Ruin Setup with regards to Player A considered above encompasses two stopping or Barrier Points, which are as follows,

- ***Absorbed State 1*** : Player A loses all the money, getting bankrupted or ruined and Player B has the total sum or capital which was available in the game, i.e., $n_1 + n_2$.

- ***Absorbed State 2*** : Player B loses all the money, getting bankrupted or ruined and Player A has the total sum or capital which was available in the game, i.e., $n_1 + n_2$.

6.3 *Diagrammatic Representation Of the Gambler's Ruin Problem as One-Dimensional Random Walk, along with the Absorbed States*

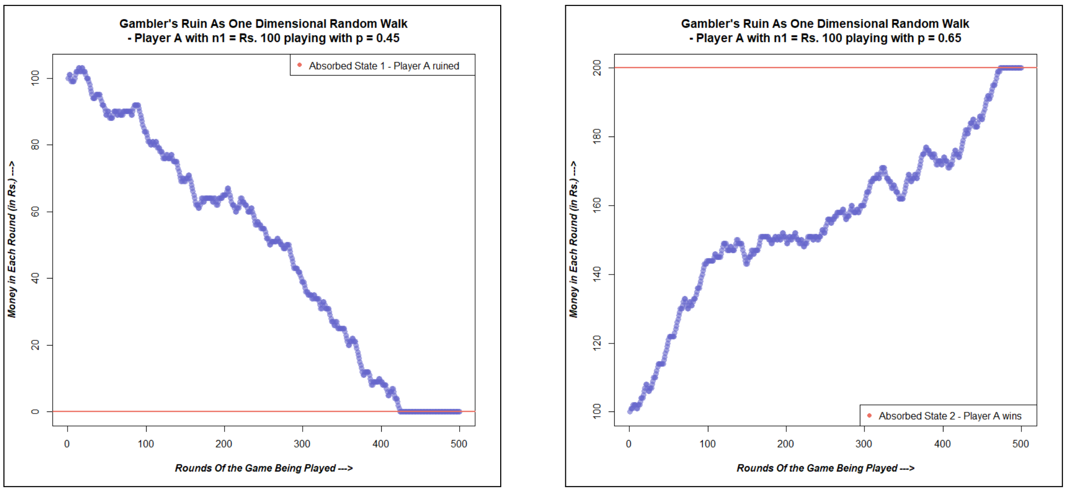

Figure 11: ***Gambler's Ruin - 1D Random Walk and it's Absorbed States***

In the above illustrations, we have shown two separate cases of Gambler's Ruin Problem, where on one hand Player A enters the Game with Rs.100 and has a probability of 0.45 of winning each round along with Player B entering the Game with Rs.100 and having a probability of q = 0.55 of winning each round. This is the situation where we observe

Absorbed State 1 - Player A gets ruined and Player B wins.

On the other hand we consider the same setup, but in this case probability of winning a particular round of the Game for A becomes p = 0.65 and that for Player B becomes q = 0.35. Now we observe ***Absorbed State 2 - Player A wins and Player B gets ruined***.

Both the Simulations done above represents the Gambler's Ruin Problem as a model of One-Dimensional Random Walk as discussed in Section **6.1**.

6.4 *The Gambler's Ruin Problem as Markov Chains*

X_n is a Random Variable denoting ***the money or the fortune*** that is there with the particular Gambler after the n^{th} Gamble or the round of the game.
Then this $\{X_n\}$ denotes a ***Markov Chain*** defined over a ***finite state space*** $\zeta = (0, 1, 2, ..., n_1 + n_2)$. On having Rs.0 (ruined) and having Rs.$n_1 + n_2$ (winning the game), the game stops.

» Is the Markovian Property satisfied in this situation ?
Yes, the property is satisfied in this game of gambling, as the Gambler's Ruin Setup under consideration ensures the fact that **the money which the gambler has with him/her at the n^{th} round of the game is independent of the sum of money which was there at the $(n-1)^{th}$ round of the game and also it won't affect the sum of money, which the gambler will acquire at the $(n+1)^{th}$ round of the game**.
Hence, given the present sum of money, the past and future sum of money during the considered game of gambling are ***conditionally independent***, as a result of which, the system evolving out of this Gambler's Ruin Setup is an example of Markov Chain.

Also this version of the Gambler's Ruin Problem can be regarded as the ***Discrete Time Stochastic Process***, since on winning or losing a game, the gambler adds either +1 or -1 to the existing or present fortune.

6.5 *The Transition Probabilities and Matrices (t.p.m) associated to the Gambler's Ruin Problem*

» *The Transition Probabilities*

Considering the problem statement and setup as mentioned in Section **4.1**, the transition probabilities are defined as follows:

- $p_{i,i+1} = p$ » moving from the i^{th} state to the $(i+1)^{th}$ state of the game with probability **p**, or equivalently winning a particular round of the game and acquiring Rs.(i+1) from Rs.i has an associated probability of **p**, $\forall\ 0 < i < n_1 + n_2$.

- $p_{i,i-1} = 1 - p = q$ » moving from the i^{th} state to the $(i-1)^{th}$ state of the game with probability **1-p = q**, or equivalently losing a particular round of the game and losing Rs.1 from Rs.i has an associated probability of **1-p = q**, $\forall\ 0 < i < n_1 + n_2$.

- For state i, $\forall$ i $\in (0, n_1 + n_2)$, there is a chance that the Gambler never returns to that particular i^{th} state, when starting at that i^{th} state, hence $p_{ii} < 1$. This i^{th} state in the Gambler's Ruin Setup is regarded as the ***Transient State of the MC***.

- $p_{0,0} = p_{n_1+n_2,n_1+n_2} = 1$ » indicating the two **Absorbed States** or **Barrier Points** involved in the game, i.e., probability of staying at state 0 or state $n_1 + n_2$ is **1**. These are also regarded as the ***Recurrent States of the MC*** under consideration.

» *The Transition Probability Matrix (t.p.m)*

Considering the problem statement and setup as mentioned in Section **4.1**, the transition probability matrix (t.p.m) is given as follows:

$$T^{(n_1+n_2+1)*(n_1+n_2+1)} = \begin{pmatrix} \mathbf{1} & 0 & 0 & 0 & 0 & \cdots & 0 \\ \mathbf{q} & 0 & \mathbf{p} & 0 & 0 & \cdots & 0 \\ 0 & \mathbf{q} & 0 & \mathbf{p} & 0 & \cdots & 0 \\ 0 & 0 & \mathbf{q} & 0 & \mathbf{p} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & \mathbf{1} \end{pmatrix}$$

Note that,

- The t.p.m above is of order ’$n_1 + n_2 + 1$’, i.e., a square matrix having $n_1 + n_2 + 1$ rows and $n_1 + n_2 + 1$ columns.

- According to the Gambler’s Ruin Problem Setup considered in Section **4.1**, the **state space** of the Markov Chain under consideration is $\zeta = (0, 1, 2, ..., n_1 + n_2)$.

- The **Rows** of the t.p.m denotes the state at which the gambler is in, during the game, i.e., the ***current state*** of the gambler.
  The **Columns** of the t.p.m denotes the state the gambler will reach after a particular round of the game has been played.

- The transition probabilities has been illustrated through the t.p.m, where it is to be observed that, the ***sum of each row equals 1***. In other words the transition probabilities of making a transition from the current state adds up to 1.

- ***Higher Order t.p.m’s*** such as T^{20}, T^{50},...., facilitates us to visualize the long-run transition probabilities after 20 rounds, or 50 rounds of the game.

6.6 *The Transition Diagram for a selected setup of Gambler's Ruin Problem*

» ***The Setup*** : Consider the structure of the Gambler's Ruin Problem as mentioned in Section **4.1**, with two players A & B, where A enters the game with Rs.1 and B enters the game with Rs.2, such that the total capital available in the game is Rs.3. Considering a ***Biased Game*** towards player A, such that A has a probability of 'p = 0.55' of winning each round, whereas B has a probability of 'p = 0.45' of winning each round. Taking into account the usual Absorbed States, the Markov Chain formed through this setup in regards to Player A, can be represented by the following ***Transition diagram***,

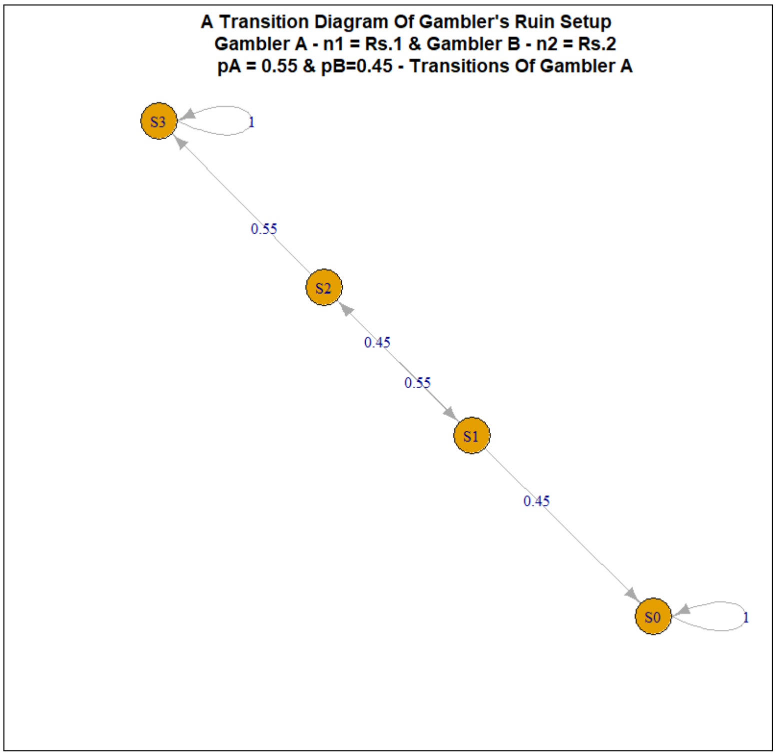

Figure 12: ***The Transition Diagram*** [2]

» ***Explaining the Transition Diagram*** : In the transition diagram illustrated above, i.e., Figure **12**, we observe the different transitions that gambler/player A, can make during the game play.

The state space of the above considered MC is $\zeta = (0, 1, 2, 3)$, which is denoted as ***S0,S1,S2,S3*** respectively in the above diagram.

The ***Recurrent States*** or the ***Absorbed States*** are S0 and S3, with transition probabilities 1 each, and the states S1 and S2 are ***Transient***.

Note that if gambler A is currently at state S1, then the gambler can move to state S2 with a probability of 0.55 or, can move down to state S0 with a probability of 0.45. Similarly, moving up from state S2 to S3 is having an associated transition probability of 0.55, and moving a state down from S2 to S1 has an associated probability of 0.45.

If the gambler reaches state S3, it would imply that the gambler has won the entire game, i.e., winning the total sum of money available in the game, so the gambler will remain in state S3 with an associated probability of 1. Analogously, if the gambler reaches state S0, it would imply the ruin of the particular gambler and staying at that state S0 is having an associated transition probability of 1.

Also, observe, moving up to a higher state from a lower state would imply the win of the gambler for that particular round or transition, and moving down from a higher state to a lower state would imply the loss of the gambler for that round or transition.

7 Solving the Gambler's Ruin Problem - An Algebraic Approach

Methodology Used : We try to express the ***Win Probability of Gambler A - where A starts the game with Rs.i*** in terms of the transition probabilities, i.e., 'p' and 'q', eventually obtaining a ***Recurrence Relation between the Win Probabilities of Gambler A for different amount of starting capitals***. This Recurrence Relation obtained is a ***Second Order Linear Difference Equation Of Homogeneous Type***, which is to be solved algebraically to get the desired probability. [16]

The solution also includes the usage of the ***Theorem Of Total Probability***.

As in the problem stated earlier in Section **4.1**, we have taken into consideration, the Gambler's Ruin scenario w.r.t two Gamblers namely **A** and **B**. Gambler A starts with an initial amount of 'Rs.n_1' and Gambler B starts with the initial amount of 'Rs.n_2' such that the total money available in the game being played is 'Rs.$n_1 + n_2$'.

Probability of any Gambler winning a particular round of the game is 'p' and that of losing is '1-p = q'. The game stops when either of the Gambler gets broke/bankrupt, i.e., having Rs.0, which would clearly imply the other Gambler has Rs.$n_1 + n_2$. Hence the two **Absorbed States** involved in the game.

On winning a round, Rs.1 is given to the winner from the losing Gambler.

OBJECTIVE : ***To determine P (Gambler A winning the game) =?***

Let us define the following:

- P_i = P(A wins the game | A starts the game with Rs.i)
- **X**: denotes that A wins the enitre game.
- **E**: denotes that A wins the first round of the game.
- X_t: denotes the amount of money with A at the t^{th} round of the Game.

To find P_i, let us condition on the first round of the game.

Note that the results of a particular round is independent of the results of the previous round.

By theorem of Total Probability:

$$\Rightarrow P_i = P(X \cap E | X_0 = i) + P(X \cap E' | X_0 = i)$$

$$\Rightarrow P_i = P(E|X_0 = i)P(X|X_1 = i+1) + P(E'|X_0 = i)P(X|X_1 = i-1)$$

$$\boxed{\Rightarrow P_i = pP_{i+1} + (1-p)P_{i-1}, 1 \leq i \leq (n_1 + n_2 - 1)} \tag{6}$$

Take a note of the boundary conditions:

$$P_0 = 0 \textbf{ (A starts bankrupt)} \text{ and } P_{n_1+n_2} = 1 \textbf{ (B starts bankrupt)} \tag{7}$$

Now, Equation (6) is a ***Recursion (Second Order Linear Difference Equation) Of Homogeneous Type*** to be solved.

We can re-write Equation (6) as -

$$x = px^2 + (1-p) \Rightarrow px^2 - x + (1-p) = 0 \Rightarrow x = 1, \frac{1-p}{p}$$

Since the roots are distinct we have,

$$P_i = \alpha 1^i + \beta(\frac{1-p}{p})^i, \text{where } \alpha, \beta \in \mathbb{R} \tag{8}$$

Now, using the boundary conditions as listed in Equation (7), we get,

- $$P_0 = \alpha + \beta = 0 \tag{9}$$

- $$P_{n_1+n_2} = \alpha + \beta(\frac{1-p}{p})^{n_1+n_2} = 1 \tag{10}$$

Solving Equations (9) & (10), we get,

$$\Rightarrow \alpha = \frac{1}{1-(\frac{1-p}{p})^{n_1+n_2}}, \ \beta = \frac{-1}{1-(\frac{1-p}{p})^{n_1+n_2}}$$

Putting α and β as obtained above in Equation (8), we get,

$$P_i = \begin{cases} \frac{1-(\frac{1-p}{p})^i}{1-(\frac{1-p}{p})^{n_1+n_2}} & \text{p} \neq \text{1-p} \\ \frac{i}{n_1+n_2} & \text{p = 0.5} \end{cases} \tag{11}$$

Hence we obtain P_i, which denotes the probability of winning the game considered under Gambler's Ruin Setup, when the gambler initially enters the game with an amount of Rs.i.

<u>FINAL SOLUTIONS</u> : ***The ultimate solutions are obtained by putting n_1 and n_2 respectively in place of i in Equation (11).***

- ***Probability that Gambler A will win the entire game, when initially starting the game with a capital (stake) of Rs.n_1.***

$$P_{n_1} = \begin{cases} \frac{1-(\frac{1-p}{p})^{n_1}}{1-(\frac{1-p}{p})^{n_1+n_2}} & \text{p} \neq \text{1-p} \\ \frac{n_1}{n_1+n_2} & \text{p = 0.5} \end{cases} \tag{12}$$

- ***Probability that Gambler B will win the entire game, when initially starting the game with a capital (stake) of Rs.n_2.***

$$P_{n_2} = \begin{cases} \frac{1-(\frac{1-p}{p})^{n_2}}{1-(\frac{1-p}{p})^{n_1+n_2}} & \text{p} \neq \text{1-p} \\ \frac{n_2}{n_1+n_2} & \text{p = 0.5} \end{cases} \tag{13}$$

8 A Different Approach to solve the Gambler's Ruin Problem - Using Higher Order Transition Probability Matrices (t.p.m's)

Methodology Used : In lieu of using difference equations to solve the problem at hand, we make use of the ***higher order transition probability matrices*** to visualize the value to which the win probability of a gambler playing the game under the Gambler's Ruin Setup, converges.
Hence we define a particular setup and build up the t.p.m on the basis of the defined setup, eventually calculating the higher order t.p.m's to find the probability that a gambler will win the entire game.

» THE SETUP :: Considering the Gambler's Ruin Setup as mentioned in Section **(4.1)**, we consider an ***Unbiased Game***.
Let us suppose that Gambler A enters the game with Rs.2 and Gambler B enters with Rs.3, such that the total money available in the game is Rs.5, a hypothetical consideration. Then the possible states in this game is defined by the set $\zeta = (0, 1, 2, 3, 4, 5)$. Since we have considered an ***Unbiased Game***, the probability of winning each round will be $\frac{1}{2}$, for both the gamblers, viz. A & B.
The ***Absorbed States*** are the points in the game where the game will come to an end, when either of the gamblers win the entire game, implying that the other gambler reaches to Rs.0, i.e., exhausts all the capital and gets ruined.
The ***Transition Probabilities*** are $\frac{1}{2}$ each for moving down from a higher state to a lower

state and also for moving up from a lower state to a higher state. Also, if the gambler reaches Rs.0 state or Rs.5 state at some point of time in the game, then the probability that the gambler stays at that very state is 1.

Therefore, the ***Transition Probability Matrix*** is given by,

$$\text{T} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0.5 & 0 & 0.5 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0.5 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0.5 & 0 & 0.5 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Let us determine the t.p.m after 20 rounds of game play, i.e.,

$$T^{20} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0.7942 & 0.0004 & 0 & 0.0064 & 0 & 0.1953 \\ 0.5924 & 0 & 0.1044 & 0 & 0.0064 & 0.3907 \\ 0.3907 & 0.0064 & 0 & 0.0104 & 0 & 0.5924 \\ 0.1953 & 0 & 0.0064 & 0 & 0.0039 & 0.7942 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Observe that, the rows of t.p.m denotes the state in which the gambler is currently in and the columns of the t.p.m denotes the state to which the gambler would reach, and hence the elements of the t.p.m, which are the transition probabilities can be interpreted accordingly.

Let's see what happens after 1000 rounds of the game in long-run, viz. the higher order transition probability matrix, i.e.,

$$T^{1000} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0.8 & 0 & 0 & 0 & 0 & 0.2 \\ 0.6 & 0 & 0 & 0 & 0 & 0.4 \\ 0.4 & 0 & 0 & 0 & 0 & 0.6 \\ 0.2 & 0 & 0 & 0 & 0 & 0.8 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

According to the solution of the Gambler's Ruin Problem, using difference equations in Section **(7)**, the probability that gambler A will win the entire game under this ***Unbiased*** setup is $\frac{2}{5} = 0.4$.

From the above defined higher order t.p.m, viz. T^{1000}, the probability that the gambler A would make a transition from State 2 to State 5 is 0.4, which is the required win probability, as reflected by the **(3,6) th element** of the t.p.m T^{1000}.

Similarly, for Gambler B, the win probability comes out to be $\frac{3}{5} = 0.6$, also reflected by the **(4,6) th element** of the t.p.m T^{1000}.

Hence, the higher order t.p.m's imparts a different dimension, when it comes to the solution of the Gambler's Ruin Problem, where the probability of winning the entire game is determined by visualizing the long-run convergence of the probability through higher order t.p.m's.

9 An Important Aspect Of Gambler's Ruin Problem - Expected Duration Of Play

9.1 *Solving The Expected Duration Of Play*

So far, we have dealt with the solution of the Gambler's Ruin Problem in the previous Section **7**. In other words, we now have an idea of the probability of winning a game under the Gambler's Ruin Setup, if we initially start the game with a capital or an amount of Rs.n_1. Following the similar analogy, here we try to determine ***the expected rounds to play the game, until we hit one of the Absorbed States or the Barrier Points***.

Methodology Used : We try to express the ***Expected Number Of Rounds to be played until Gambler A hits one of the Barrier Points - where A starts the game with Rs.i***, in terms of the expected rounds with a lower amount of initial sum, eventually obtaining a ***Recurrence Relation between the Expected Rounds to be played by Gambler A for different amount of starting capitals***. This Recurrence Relation obtained is a ***Second Order Linear Difference Equation Of In-Homogeneous (Non-Homogeneous) Type***, which is to be solved algebraically to get the desired result. [25]

Note that, the letter ***"E"*** in the following solution denotes the ***"Expectation Of a Random Variable"***.

Considering the usual structure of the Gambler's Ruin Problem as mentioned in Section **4.1**, we define the following,

- $\mathbb{E}_i$ = Expected Rounds to be played by the player to hit one of the stopping points, i.e., either get ruined or win the entire game, when starting the game with an initial amount of Rs.i.
- ψ = Number Of Rounds to be played until we reach one of the Absorbed States.
- X_t: denotes the amount of money with the Gambler at the t^{th} round of the Game.
- W: denotes that the Gambler wins the first round of the game.

OBJECTIVE : ***To Find the Expected Number of Rounds until we reach one of the Barrier Points with initially Rs.i =*** $\mathbb{E}_i$.

To find $\mathbb{E}_i$, let us condition on the first round of the game.

Note that the results of a particular round is independent of the results of the previous round.

$$\Rightarrow \mathbb{E}_i = E(\psi|X_0 = i) \tag{14}$$

By theorem of Total Probability:

$$\Rightarrow \mathbb{E}_i = E(\psi|X_0 = i \cap W)P(W) + E(\psi|X_0 = i \cap W')P(W')$$

$$\Rightarrow \mathbb{E}_i = E(\psi|X_1 = i+1)p + E(\psi|X_1 = i-1)(1-p)$$

$$\Rightarrow \mathbb{E}_i = E(\psi|X_1 = i+1)p + E(\psi|X_1 = i-1)(1-p)$$

$$\Rightarrow \mathbb{E}_i = [1 + E(\psi|X_0 = i+1)]p + [1 + E(\psi|X_0 = i-1)](1-p)$$

From Equation (14), we have,

$$\Rightarrow \mathbb{E}_i = 1 + p\mathbb{E}_{i+1} + (1-p)\mathbb{E}_{i-1}$$

$$\boxed{\Rightarrow p\mathbb{E}_{i+1} - \mathbb{E}_i + (1-p)\mathbb{E}_{i-1} = -1, 1 \leq i \leq (n_1 + n_2 - 1)} \tag{15}$$

Equation (15) is a ***Linear Second-Order Non-Homogeneous Difference Equation***, which is to be solved.

Depending on the Absorbed States taken into consideration for the given problem at hand, we have the following two ***Boundary Conditions***,

Take a note of the boundary conditions:

$$\mathbb{E}_0 = 0 \text{ and } \mathbb{E}_{n_1+n_2} = 0 \tag{16}$$

The above conditions signify that, the Gambler is already in the Barrier Point or the Absorbed State, hence expected rounds to be played under both of the above situations are 0.

» CASE I : p ≠ $\frac{1}{2}$, A Biased Game Play

First, let us consider the ***Homogeneous counterpart of Equation (15)***, which is as follows,

$$\Rightarrow p\mathbb{E}_{i+1} - \mathbb{E}_i + (1-p)\mathbb{E}_{i-1} = 0 \tag{17}$$

which is analogous to Equation (6), hence the solution of Equation (17) is given by,

$$\mathbb{E}_{ip} = \alpha 1^i + \beta(\frac{1-p}{p})^i, \text{where } \alpha, \beta \in \mathbb{R} \tag{18}$$

We now guess a particular solution of Equation (18), i.e.,

$$\Rightarrow \mathbb{E}_{i\,particular} = \xi_1 + i\xi_2, \text{ where } \xi_1, \xi_2 \in \mathbb{R} \tag{19}$$

Note that, the above assumed solution, expresses $\mathbb{E}_{i\,particular}$ as a ***linear function*** of the starting capital 'i' of the Gambler.

Putting the form of $\mathbb{E}_{i\,particular}$ as given by Equation (19) in Equation (15), we get,

$$\Rightarrow p(\xi_1 + (i+1)\xi_2) - \xi_1 - i\xi_2 + (1-p)(\xi_1 + (i-1)\xi_2) = -1$$

$$\Rightarrow p\xi_1 + ip\xi_2 + p\xi_2 - \xi_1 - i\xi_2 + \xi_1 + i\xi_2 - \xi_2 - p\xi_1 - ip\xi_2 + p\xi_2 = -1$$

Simplifying, we obtain,

$$\Rightarrow \xi_2 = \frac{1}{1-2p} \tag{20}$$

Note that, there is no restriction or constraint in ξ_1, thereby setting $\xi_1 = 0$, we get,

$$\Rightarrow \mathbb{E}_{i\,particular} = \frac{i}{1-2p} \tag{21}$$

Therefore,

$$\Rightarrow \mathbb{E}_i = \mathbb{E}_{ip} + \mathbb{E}_{i\,particular} \tag{22}$$

The Final Form of the Solution is obtained by plugging Equations (18) and (21) in Equation (22), thus obtaining,

$$\boxed{\Rightarrow \mathbb{E}_i = \alpha 1^i + \beta(\frac{1-p}{p})^i + \frac{i}{1-2p}} \tag{23}$$

Now we are to solve for α and β using the boundary conditions as stated in Equation (16). We get,

- $$\mathbb{E}_0 = 0 = \alpha + \beta \tag{24}$$

- $$\mathbb{E}_{n_1+n_2} = 0 = \alpha + \beta(\frac{1-p}{p})^{n_1+n_2} + \frac{n_1+n_2}{1-2p} \tag{25}$$

Solving Equations (24) and (25) simultaneously, we obtain,

$$\Rightarrow \alpha = -\beta \text{ and } \beta = \frac{\frac{-(n_1+n_2)}{1-2p}}{(\frac{1-p}{p})^{n_1+n_2} - 1}$$

Hence putting the values of α and β in Equation (23), we get,

$$\boxed{\Rightarrow \mathbb{E}_i = \frac{i}{1-2p} - \frac{n_1+n_2}{1-2p}\frac{(\frac{1-p}{p})^i - 1}{(\frac{1-p}{p})^{n_1+n_2} - 1},\ p \neq \frac{1}{2}} \tag{26}$$

» <u>CASE II</u> : p = $\frac{1}{2}$, An Unbiased Game Play

In this case, from equation (20), we have,

$$\Rightarrow (1-2p)\xi_2 = 1$$

$$\Rightarrow 0 = 1,\ \text{putting } p = \frac{1}{2},\ \textit{A Contradiction !}$$

Therefore, we choose our assumed solution of $\mathbb{E}_i$ to be a higher degree polynomial, rather than a linear function of 'i'. We take,

$$\Rightarrow \mathbb{E}_{i\ particular} = \xi_1 i^2 + \xi_2 i + \xi_3,\ \text{where } \xi_1, \xi_2, \xi_3 \in \mathbb{R} \tag{27}$$

Plugging this form of $\mathbb{E}_{i\ particular}$ into the recurrence relation as obtained in Equation (15), and on some algebraic manipulations, we obtain,

$$\Rightarrow \xi_1 = -1, \xi_2 = 0 = \xi_3$$

Therefore,

$$\Rightarrow \mathbb{E}_{i\ particular} = -i^2 \tag{28}$$

When p = $\frac{1}{2}$, the ***Homogeneous Solution*** is,

$$\Rightarrow \mathbb{E}_{ip} = \alpha i + \beta \tag{29}$$

Using Equations (28) and (29) along with the boundary conditions solving for α and β, we get,

$$\boxed{\Rightarrow \mathbb{E}_i = i(n_1 + n_2 - i),\ \text{p} = \frac{1}{2}} \tag{30}$$

<u>FINAL SOLUTIONS</u> : ***The ultimate solutions are obtained by putting n_1 and n_2 respectively in place of i in Equations (26) and (30)***

- ***The Expected Rounds to be played of the Game under Gambler's Ruin Setup, to hit one of the barrier points (either get broke or win the entire game), when initially starting with Rs.n_1 are,***

$$\mathbb{E}_{n_1} = \begin{cases} \frac{n_1}{1-2p} - \frac{n_1+n_2}{1-2p} \frac{(\frac{1-p}{p})^{n_1}-1}{(\frac{1-p}{p})^{n_1+n_2}-1} & \text{p} \neq 0.5 \\ n_1 n_2 & \text{p} = 0.5 \end{cases} \tag{31}$$

- ***The Expected Rounds to be played of the Game under Gambler's Ruin Setup, to hit one of the barrier points (either get broke or win the entire game), when initially starting with Rs.n_1 are,***

$$\mathbb{E}_{n_2} = \begin{cases} \frac{n_2}{1-2p} - \frac{n_1+n_2}{1-2p} \frac{(\frac{1-p}{p})^{n_2}-1}{(\frac{1-p}{p})^{n_1+n_2}-1} & \text{p} \neq 0.5 \\ n_1 n_2 & \text{p} = 0.5 \end{cases} \tag{32}$$

9.2 *Some Observations on The Expected Duration Of Play*

- ***OBSERVATION 1*** : As per the expression of $\mathbb{E}_i$ for a ***biased game*** where $p \neq \frac{1}{2}$, obtained in Equation **(26)**, we have,

$$p < \frac{1}{2}, \Rightarrow \mathbb{E}_i \leq \frac{i}{1-2p} \tag{33}$$

and,

$$\text{As } (n_1 + n_2) \to \infty, \Rightarrow \mathbb{E}_i \to \frac{i}{1-2p} \tag{34}$$

From Equation (**34**), we note that as the total sum of money available in the game becomes very large, on an average it requires $\frac{i}{1-2p}$ rounds of the game to be played by a gambler for hitting one of the Barrier Points, i.e., either win or lose the game being played, when the gambler enters the game with Rs.i.

- ***OBSERVATION 2*** : In case of an ***unbiased game***, the expression of $\mathbb{E}_i$ is given as in Equation (**30**), i.e., $\mathbb{E}_i$ = (i)*$(n_1 + n_2 - i)$, implying that in a symmetric or fair game, the expected number of rounds to be played to hit one of the absorbed states, i.e., either win or lose the game, the player has to play a number of rounds which is equal to the ***Starting Capital of the player times the amount or sum, the player is willing to lose during the game play***.

- ***OBSERVATION 3*** : This observation is an offshoot of Observation 2. We have already made the proposition that, for an ***unbiased*** game, a player needs to play [(i)*(the amount of money the opponent is starting with)] rounds of the game, (where 'i' is the initial capital of the player), in order to reach one of the absorbed states.
Suppose Player A has a very trivial sum of Rs.1 to start the game with, and the opposition, i.e., player B, starts with a sum of Rs.10,000, in that case, even though Player A has a very small sum, A is expected to play 10,000 rounds of the game to hit one

of the barrier points (either to win or lose the game being played), which is quite difficult to believe, irrespective of the mathematics lying in front of us.

10 Simulations, Results & Observations

10.1 *SIMULATION I - Simulating the Solution Of the GRP*

10.1.1 Motivation & Objective

The primary objective of simulating the solution of the Gambler's Ruin Problem is to visualize the convergence of the long-run probability of winning a particular game being played under the Gambler's Ruin setup.
The simulation also facilitates in providing a different aspect to the solution of the Gambler's Ruin Problem, which has been previously solved using ***Second Order Linear Homogeneous Difference Equation*** in Section **7**.

10.1.2 The Algorithm

The Simulation is centralized around the user-built function ***'grp.win.prob(n1,n2,p,ngames,r)'***. The function grp.win.prob() simulates the setup of the Gambler's Ruin Problem cloning the structure as mentioned in Section **4.1**.

» The Function Parameters ::

- ***n1*** :: denotes the initial capital (in Rs.) with which the gambler A enters the game.
- ***n2*** :: denotes the initial capital (in Rs.) with which the gambler B, playing against gambler A, enters the game.

- ***p*** :: denotes the probability with which gambler A wins each round of the game, implying that the probability with which gambler A's opponent wins each round is ***q = 1-p***.
- ***ngames*** :: denotes the number of games to be played.
- ***r*** :: denotes the number of times each game is to be simulated or repeated.

The function *grp.win.prob()*, takes user-input for the above listed parameters and simulates each game for 'r' times, where gambler A enters the game with Rs.n_1 and gambler B enters the game with Rs.n_2. In these 'r' repetitions of the 'ngames' games, we get the probability that A wins that particular game which is simulated for 'r' times. [26]

*» **Return*** :: A vector of length 'ngames', denoting the win probability of the gambler under consideration (A or B) in each of the 'ngames' games being simulated.

10.1.3 The Results

» The Results for varying values of the parameters for ***Gambler A*** are as follows:

(n_1, n_2)	***p***	***ngames***	***r***	***Simulated win prob of A***	***Theoretical win prob of A***
(1,3)	0.5	1000	1000	0.249	0.25
(10,5)	0.5	100	100	0.655	0.66
(25,25)	0.5	100	1000	0.499	0.50
(8,10)	0.56	100	1000	0.922	0.92
(8,10)	0.45	100	1000	0.178	0.18

Table 1: ***Simulation Results Of Win Probability Of Player A***

» The Results for varying values of the parameters for ***Gambler B*** are as follows:

(n_1, n_2)	*q*	*ngames*	*r*	*Simulated win prob of B*	*Theoretical win prob of B*
(1,3)	0.5	1000	1000	0.748	0.75
(10,5)	0.5	100	100	0.324	0.33
(25,25)	0.5	100	1000	0.5006	0.50
(8,10)	0.44	100	1000	0.077	0.078
(8,10)	0.55	100	1000	0.823	0.821

Table 2: ***Simulation Results Of Win Probability Of Player B***

10.1.4 Visualization & Observations

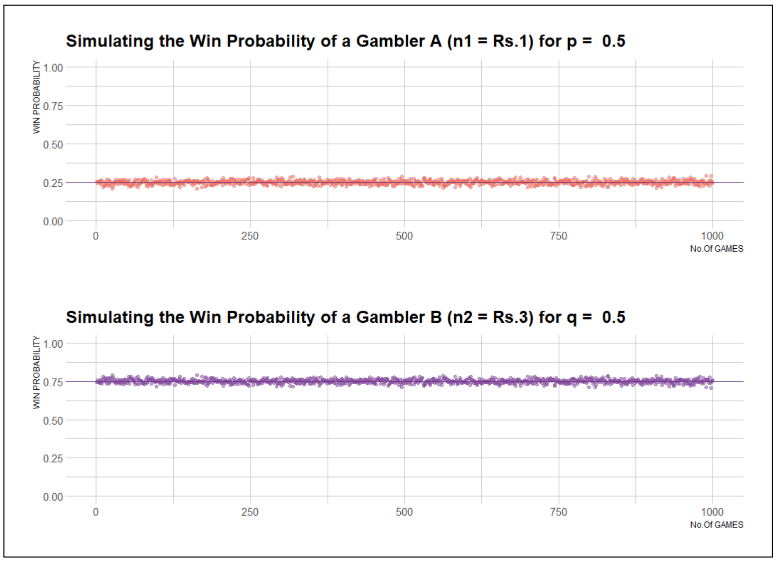

Figure 13: ***Convergence Of Win Probability in Gambler's Ruin Setup***

» Comment :: The above tabular representations (Table 1 & Table 2) shows the ***average*** of the simulated win probabilities for gamblers A & B, for different choices of initial capital and number of games to be played.

It can be clearly noted that the simulation results mimic the theoretical observations of the win probability obtained from the relation, by solving the required difference equation in Section **7**.

It is to be noted that, for an ***Unbiased Game***, i.e., p = 0.50, starting the game with a higher capital at one's disposal would result in a higher win probability.

On the other hand, as far as the win probability in the ***Biased Games***, i.e., $p \neq \frac{1}{2}$, is concerned, the gambler having a larger magnitude of initial capital to start the game with, will tend to be more probable to win the game, irrespective of the nature of bias of the game (i.e., value of 'p').

A diagrammatic representation illustrating the above fact is as follows:

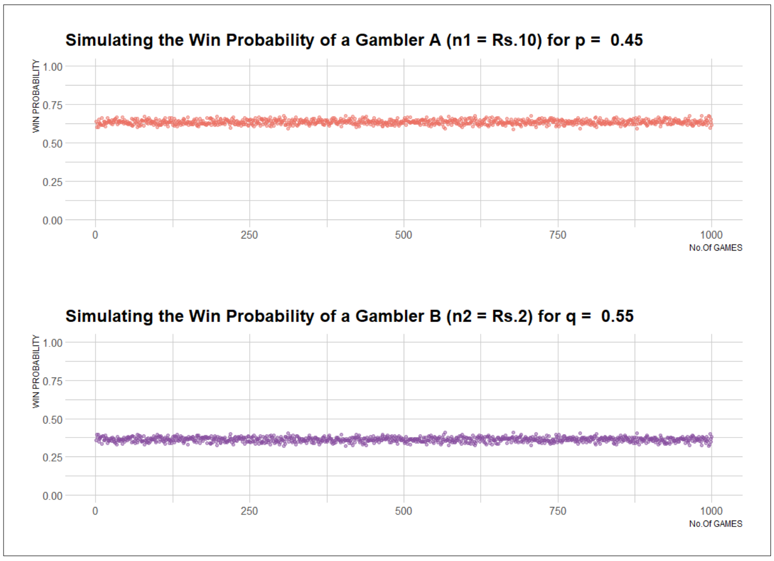

Figure 14: ***Win Probability independent of Nature of Bias of the game***

Even though Gambler A, has a lower probability of winning each round of the gambling game, but tend to have a higher win probability, since starting with an initial capital of Rs.10, which is larger in comparison to the capital with which gambler B enters the game, i.e., Rs.2.

10.2 *SIMULATION II - Trajectory Of Money at each round of the game, with the Gambler*

10.2.1 Motivation & Objective

In this simulation performed, we expect to give a visual representation of the capital (in Rs.) with the gambler, after each bet or round of the game being played under the structural skeleton of Gambler's Ruin Setup, taking into consideration different choices for 'p', hence simulating for both ***Unbiased*** as well as ***Biased*** games.

10.2.2 The Algorithm

The function performing the simulation under consideration takes user-input for the initial capitals (in Rs.), with which the gamblers A & B start the game, along with the probability of A winning each round. The game gets simulated for a particular number of rounds eventually giving the money with the gambler at each round of the game.
The user-defined function is ***'grp.money.trajectory(n1,n2,p,nrounds)'***, as explained above.

» The Function Parameters ::

- ***n1*** :: denotes the initial capital (in Rs.) with which gambler A enters the game.
- ***n2*** :: denotes the initial capital (in Rs.) with which gambler B enters the game.
- ***p*** :: denotes the probability with which gambler A wins each round of the game, implying that the probability with which gambler A's opponent wins each round is **q = 1-p**.
- ***nrounds*** :: denotes the number of rounds to be played in a game.

» ***Return*** :: The Diagrammatic Representation of the Trajectory Of Money with the gamblers at each round in the game being played under the Gambler's Ruin Setup.

10.2.3 Visualization & Observations

- ***A Fair Game Play - An Unbiased Game with p =*** $\frac{1}{2}$***,*** n_1 ***= Rs.100,*** n_2 ***= Rs.100***

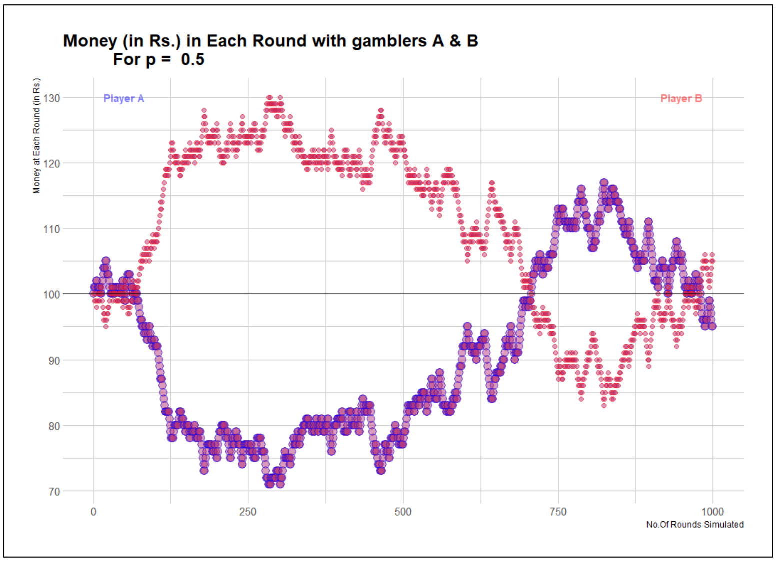

Figure 15: ***The Trajectory of Money in an Unbiased Game***

» ***Comment*** :: In this game play both the gamblers A as well as B, starts with the same initial capital, i.e., Rs.100, such that the total money available in the game is Rs.200, with p = $\frac{1}{2}$, i.e., a fair game.

Figure **(15)**, illustrates the amount of money with the gamblers at each stage or round of the game being played.

Observe that, in this unbiased game considered, initially Gambler A starts on a lower note, losing money every round in comparison to Gambler B, but increasing on the amount of capital per round after the **500th round**, crossing Gambler B just before the **750th round**, and finally once again starting to lose money just before the completion of 1000 rounds, for which the game has been simulated.

- ***Two Biased Game Plays with p = 0.55 & p = 0.45, keeping the starting capitals fixed at Rs.100 each.***

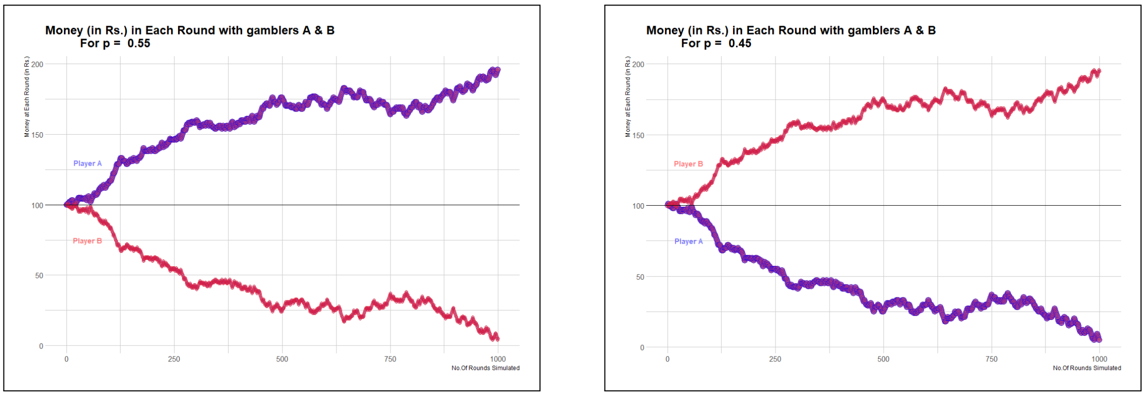

Figure 16: ***The Trajectory of Money in two Biased Games***

» Comment :: In the above figures, two nature of biases have been illustrated.

On one hand, we have,

Case I : The game being biased towards gambler A and biased against gambler B, with p = 0.55.

And, on the other hand, we have,

Case II : The game being biased against gambler A and biased towards gambler B, with p = 0.45.

In both the cases I as well as II, we see that the gamblers, starting the game with equal amounts of capital, has the game inclined towards himself/herself when having a higher probability of winning each round of the game. The above two games has been simulated for 1000 rounds each, and during these rounds, there is a permanent increase or addition to the capital of the gambler, who is having a higher chance to win each round, and a constant decrease or deduction from the capital of the gambler who is having a lower probability of winning each round in comparison to the opponent, eventually making the game to stop as one of the two gamblers reach the upper limit, thereby winning the entire game, that is the total money available in the game (Rs.200 in our simulation), making the other gambler to reach to Rs.0, and eventually losing the game. These two points also portray the two ***Absorbed States*** in the Gambler’s Ruin Problem.

10.3 *SIMULATION III - Simulating the Expected Duration Of Play*

10.3.1 Motivation Objective

We have already taken into consideration the notion of ***Expected Duration Of Play***, which is one of the most important aspects in the Gambler’s Ruin Problem, thereby providing a solution for it in Section **(9.1)**, for both biased as well as unbiased setup of the game.
Here we try to simulate the expected rounds of the game to be played by a particular gambler under the Gambler’s Ruin Setup, till the gambler reaches one of the ***Absorbed States***, i.e., either winning the game or losing the game.

10.3.2 The Algorithm

The user-defined function ***’gamblers.ruin.expected.rounds(n1,n2,p,ngames)’***, performs the simulation of the expected rounds to be played until hitting one of the barrier points per games in a simulation of total **’ngames’** games under the Gambler’s Ruin Setup.
The function being an user-built function generates the environment that prevails in a Gambler’s Ruin Setup, taking input for the initial capital of the two gamblers A & B, along with the probability of A winning each round of the game and the number of games to be played as **’ngames’**, and therefore eventually returns the number of rounds to be played in each of the **’ngames’** games, until one of the two absorbed states are encountered.

» The Function Parameters ::

- ***n1*** :: denotes the initial capital (in Rs.) with which gambler A enters the game.
- ***n2*** :: denotes the initial capital (in Rs.) with which gambler B enters the game.
- ***p*** :: denotes the probability with which gambler A wins each round of the game, implying that the probability with which gambler's A opponent wins each round of the game is **q = 1-p**.
- ***ngames*** :: The number of games to be simulated.

» Return :: A vector of length **'ngames'** giving the rounds to be played each game to hit one of the barrier points, which is then diagrammatically represented through a ***Histogram***, for different values of **'p'**.

10.3.3 Visualization & Observations

We simulate the game under the following choices of the parameters,

Cases	***n1***	***n2***	***p***	***ngames***
Case I	30	25	p = 0.40,0.45,0.50,0.55,0.60,0.65	1000
Case II	25	30	p = 0.40,0.45,0.50,0.55,0.60,0.65	500
Case II	25	25	p = 0.40,0.45,0.50,0.55,0.60,0.65	1000

Table 3: ***Parameter Choice for Simulation Of Expected Duration Of Play***

All the above listed cases, i.e., Case I, Case II and Case III, are simulated for varying choices of 'p', to illustrate the nature of bias of the game both towards gambler A & B.

- ***CASE I*** **:: Here n_1 = Rs.30, n_2 = Rs.25 and games simulated is 1000**

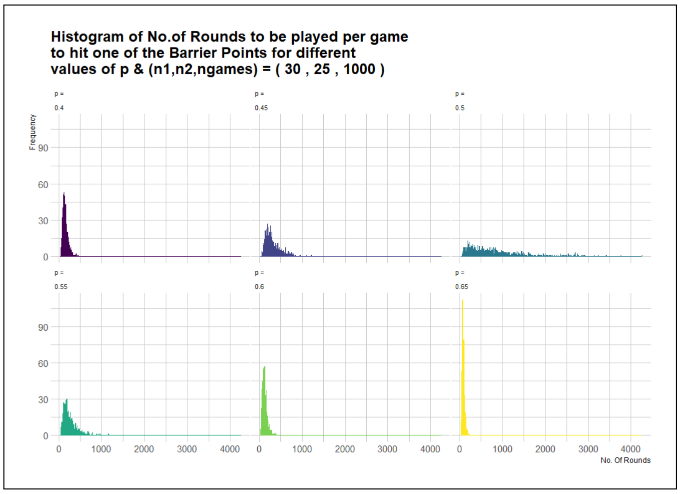

Figure 17: ***Histogram Of Expected Duration Of Play - Case I***

- ***CASE II*** **:: Here n_1 = Rs.25, n_2 = Rs.30 and games simulated is 500**

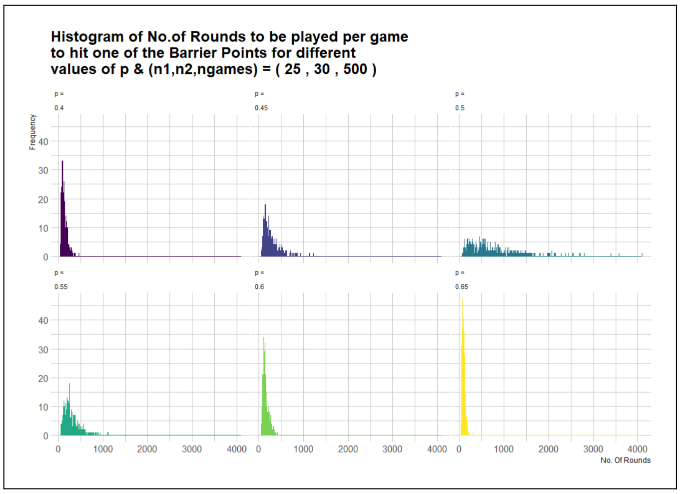

Figure 18: ***Histogram Of Expected Duration Of Play - Case II***

- ***CASE III*** **:: Here n_1 = Rs.25, n_2 = Rs.25 and games simulated is 1000**

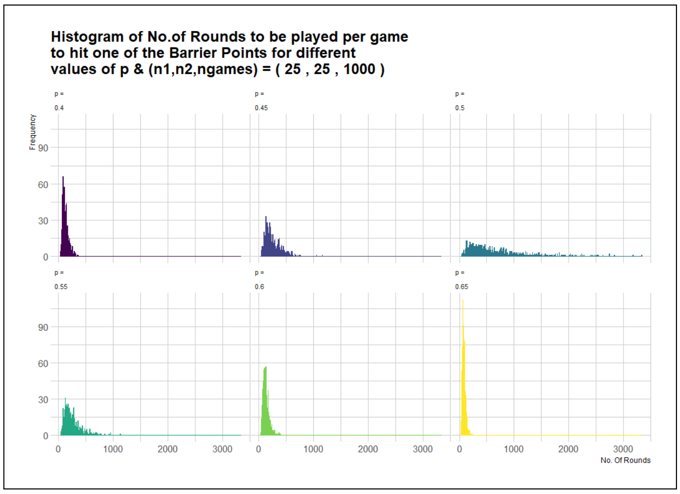

Figure 19: ***Histogram Of Expected Duration Of Play - Case III***

» Comment :: In the above Histograms considered, with varying values of initial capitals, p and ngames, we observe unanimously that in all of the three cases mentioned above, when the game drifts from lower values of p to higher values of p, the ***kurtosis*** of the number of rounds to be played until reaching one of the Absorbed States, changes, keeping a ***highly positively skewed distribution*** for the expected number of rounds to be played.

In other words, when we allow the value of 'p' to vary from a ***small range***, say ***less than 0.5 (biased against A and biased towards B)***, to a ***higher range***, say ***greater than 0.5 (biased towards A and biased against B)***, the histograms become ***highly leptokurtic***, i.e., having ***high peaks***.

Note that as we move towards an unbiased game, the histograms become flat peaked, i.e., the case when the value of p tend to move towards 0.5 from both the sides, i.e., approaching 0.5 from a lower or a higher value of p.

10.4 *SIMULATION IV - Expected Duration Of Play vs Initial Amount Of Capital with the Gambler*

10.4.1 Motivation & Objective

Here, we try to visualize or determine a relationship between the ***Expected Duration Of Play*** & the ***Initial Capital with which the gambler enters the game***, for varying values 'p'. Even though, an ***Unbiased Game***, seems to be more justified on the part of the gambler, simulations have also been done for the different nature of bias, that may be present in a particular game.

This would essentially give an idea about the expected duration of the play, i.e., in other words, for how long would the game continue until one of absorbed states is encountered by a gambler playing the game, for different amounts of initial capital (in Rs.) with which the gambler starts playing, thereby establishing an association between the number of rounds for which the game runs and the initial stake/capital.

10.4.2 The Algorithm

Here, once again an user-built function has been put to use, taking input for the ***initial stake/capital*** with which the gambler is aspiring to enter the game, along with the ***upper limit***, i.e., the sum of money which the gambler desires to win by gambling. The function also uses the ***value of the probability that the gambler wins each round of the game***, i.e., 'p' as an input.

The simulation is done on the basis of these inputs, eventually giving us the ***expected number of rounds to be played, until completion***, which is then used to make a parallel representation of the expected rounds and the initial stake.

The function being used is ***'rounds.vs.initial.sum(sum.ini,upper.limit,p)'***.

» ***The Function Parameters*** ::

- ***sum.ini*** :: The initial sum of money or capital (in Rs.), with which the gambler starts gambling.
- ***upper.limit*** :: An upper limit of money, which the gambler aspires to attain by gambling.
- ***p*** :: The probability with which the gambler wins each round of the game.

» ***Return*** :: A vector giving the ***Expected Rounds*** to be played along with the different choices of ***Initial Capital***.

10.4.3 Visualization & Observations

- ***Keeping the bias towards the gambler -*** $p \geq 0.5$

Here we simulate the game for $p \geq 0.5$, upper limit = Rs.200 and a choice of initial capitals ranging from Rs.1 to Rs.100 - a hypothetical consideration.

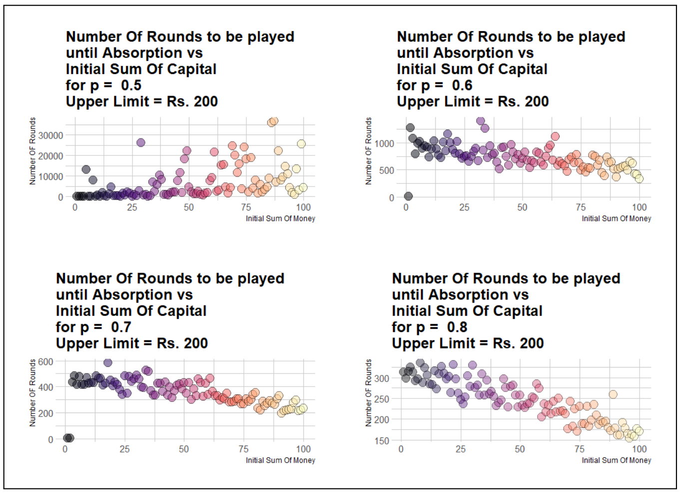

Figure 20: ***Expected Duration Play vs Initial Capital,*** $p \geq \frac{1}{2}$

» ***Comment*** :: In the Figure **(20)** presented above, it is observed that, as we tend to increase the value of p from 0.5 to a higher level, the number of rounds to be played before completion of the game becomes ***inversely related*** to the initial capital/stake of the gambler. In other words, entering the game with higher amount of initial capital would help the game reach one of it's absorbed states faster (maybe a win or a loss for the gambler), for formidably higher values of p, i.e., a ***decreasing relationship*** between expected rounds to be played and the initial capital, when values of p are greater than equal to 0.5.

But the important factor, which is to be noted is that, during an ***Unbiased Game***, even though we enter the game with a high amount of money, i.e., Rs.100, aiming to reach the upper limit of Rs.200, there are a few possibilities ,that the game will be played under the Gambler's Ruin Setup for even more than 30,000 rounds before hitting either of the absorbed states.

- ***Keeping the bias against the gambler -*** $p \leq 0.5$

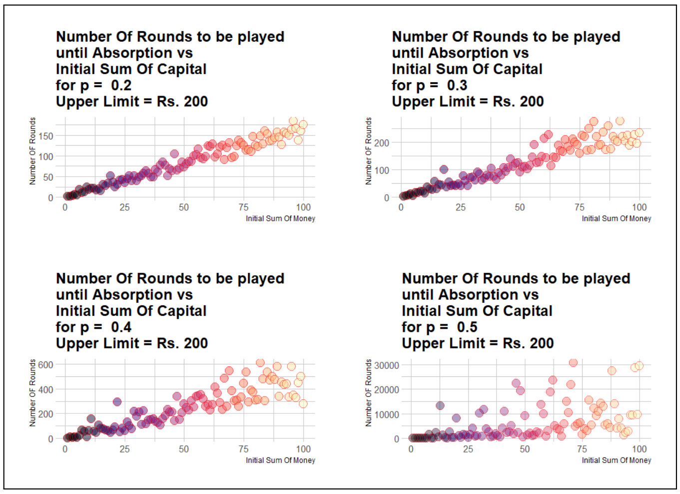

Figure 21: ***Expected Duration Play vs Initial Capital,*** $p \leq \frac{1}{2}$

» Comment :: In the Figure **(21)** presented above, it is observed that, as we tend to decrease the value of p from 0.5 to a lower level, the number of rounds to be played before completion of the game becomes ***increasingly related*** to the initial capital/stake of the gambler. In other words, entering the game with higher amount of initial capital will take more number of rounds, before the game reaches an absorbed state (win or loss for the gambler), for seemingly smaller values of p, which is quite natural, as the gambler is now having lesser probability to win each round in the game being played.

In this situation also, we observe that in case of simulating ***Unbiased Games***, the ***variability*** in the rounds being played is more, corresponding to higher levels of initial capital/stake.

10.5 *SIMULATION V - Simulating Gambler's Ruin Scenario in a game of European Roulette - The N-Player Ruin Game*

10.5.1 Motivation & Objective

In this section, a ***game of roulette (European)*** [27] is simulated, keeping the ***Gambler's Ruin Setup*** in mind.

The classic approach to the game of roulette, where betting is done on the 36 available colors on the roulette board, is featured through this simulation. Out of the three possible betting strategies in a game of roulette, which includes ***Betting On Colors***, ***Betting On Blocks*** and ***Betting On Numbers***, the first one is depicted here.

10.5.2 Structure of the Roulette Game

The game concerning to the system of betting on colors, have the ***odd numbers***, i.e., 1,3,...,35 assigned to the color ***black***, whereas the ***even numbers*** 2,4,...,36 are assigned to the color ***red***. There is a ***green pocket*** in the roulette table depicting the number ***'0'***, where if the roulette ball lands, the gambler loses out on the wagered amount. This makes the game to be regarded as the game of ***European Roulette***.

The bet is customarily placed by the gambler on any of the two available colors, i.e., Red or Black, and on spinning the roulette wheel, if the ball lands on the pocket, which associates to the color on which the bet was placed, then the gambler wins and the wagered amount for that particular round is added to the existing capital. On losing the bet, the wagered amount gets subtracted from the gambler's existing capital.

Note that, the probability that the gambler wins each round of the game is $\frac{1}{2}$, as half of the numbers from 1 to 36 are listed as the color black, and the remaining half as the color red.

10.5.3 The Algorithm

The user-built function ***'bet.colors(capital,col,wager.cap)'*** simulates the roulette game, i.e., performing the spin of the roulette wheel, after taking input for the ***capital*** with which the gambler enters the game, the ***color*** on which the gambler wants to place the bet and lastly ***wager.cap*** which is the amount, the gambler wants to wager during each round of the game. The function after each round of the roulette game play, ***adds*** the wagered amount to the existing capital of the gambler, when the gambler wins the round, otherwise ***deducts*** the wagered amount from the existing capital of the gambler.

The ***wagered amount*** during each round of the game is considered to be $\frac{1}{4}^{th}$ of the existing capital (a hypothetical consideration), that is there present along with the gambler.

Keeping the Gambler's Ruin Scenario in mind, the ***Absorbed States*** are defined as follows:

- The first absorbed state is the situation, when the existing capital of the gambler becomes less than 10^{-2} of the initial capital, with which the gambler started the Roulette game play.
- The second absorbed state is the situation, when the existing capital of the gambler becomes at least the upper limit, which the gambler wishes to attain.

Under the above two scenarios, the Roulette game comes to an end, thereby signifying the two ***barrier points*** in this game play.

» The Function Parameters ::

- ***capital*** :: The amount of money (in Rs.), with which the gambler enters this game play of roulette.
- ***col*** :: This depicts the color on which the gambler wishes to bet, where the input of the number '1' denotes a bet on the red color and the input of the number '0' denotes a bet on the black color.
- ***wager.cap*** :: The amount to be wagered each round along with the bet on the chosen color. In this simulation, the wager has been fixed at $\frac{1}{4}$th of the initial capital.

Here, the bet on the color is sampled from 0 and 1 with equal probability for each round of the roulette game, and this '0' is not to be confused with the '0' depicting the green pocket in the roulette table.

» Return :: The function bet.colors() upon simulating the game of roulette, returns the number of gamblers, who crosses the upper limit and also the ones who gets ruined due to insufficient capital in the game play, among a set of gamblers, playing the game under the defined environment.

Since we consider here a framework of the ***Gambler's Ruin Problem***, with more than 2 players in our setup, this can be regarded as the ***N-Player Ruin Game***, where ***'N'*** is the number of gamblers participating in the game being played.

The game being played by these array of gamblers is then repeated for ***'r'*** times, in order to visualize the convergence of probability of exceeding the upper limit or the probability of getting ruined, due to insufficient capital, in this roulette game being played, under the Gambler's Ruin Setup.

10.5.4 The Results

The ***Wagered Amount*** each round is $\frac{1}{4}^{th}$ of the existing capital of the Gambler.

Here $'p^{*\prime}$ represents the proportion of gamblers among N gamblers exceeding the upper limit.

And, $'q^{*\prime}$ represents the proportion of the gamblers among N gamblers getting ruined.

'N' denotes the total number of gamblers participating in the Roulette game play.

The ***Upper Limit*** is considered to be ***3 times*** the initial capital, with which the gamblers enter the game.

Capital	***Upper Limit***	***No. Of Gamblers (N)***	***r***	$\overline{p^*}$	$\overline{q^*}$
500	1500	100	1000	0.2361	0.7639
500	1500	1000	1000	0.2354	0.7646

Table 4: ***The Results Of European Roulette Simulation***

10.5.5 Visualization & Observations

» Comment :: As per the results obtained in Table **4**, when the game of roulette is simulated for different number of gamblers participating in the game play, each with equal amount of initial capital, i.e., Rs.500, and the probability for each of them winning a particular round of the game being 'p = $\frac{1}{2}$', it is observed that, each time the proportion of gamblers attaining at least the upper limit of the money considered, is way ***lesser*** than the proportion of gamblers that gets ruined during the game play.

Now, the remarkable phenomenon about the above observation made, is that, it happens even though the game is considered to be an ***Unbiased Game***. Therefore, naturally the question arises, that whether the game being played on a large scale is really unbiased? The answer to this question reveals the fundamental idea behind the great earnings that are bagged by the casinos all over the globe.

The game of ***European Roulette*** that gets simulated is actually ***not*** purely an ***Unbiased Game***, since there is a green pocket, i.e., the number '0' apart from 1,2,..,36, where the ball can land, once the roulette is given a whirl. Hence when the gambler bets on one of the colors, i.e., either black or red, assigned to the numbers ranging from 1,2,...,36, we should also consider the probability that is associated with the ball entering the green pocket, making the game being played, a ***Biased*** one.

As a result of which, after accounting for the total games played in a particular casino, huge profits are earned by them at the end of the day.

A diagrammatic representation of the proportion of gamblers attaining at least the upper limit in the game of European Roulette is presented below. The figure also highlights the

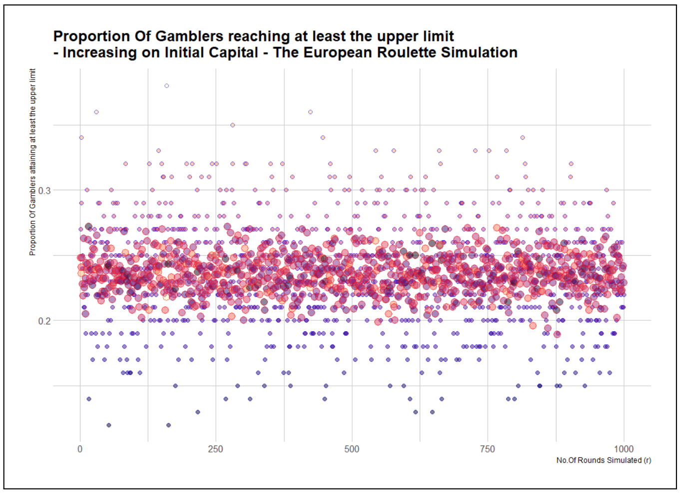

Figure 22: ***Proportion Of Winning Gamblers in the Game Of European Roulette***

N-Dimensional Gambler's Ruin Problem, often commonly regarded as the ***N-Player Ruin Game***. The Roulette Game has been simulated under the Gambler's Ruin Setup, considering a total of 100 and 1000 respective gamblers participating in the game.

11 The Secret Behind Casino's Success - Hosting Games on a Large Scale

As discussed in the previous section (**10.5**), about the game of roulette, which was greatly inclined towards the end of the casino, even though it seemed to be a fair game, technically speaking, it had a higher ***house-edge***. [28]

Most of the games that are offered to the gamblers in a casino are inclined towards the house.

There are certain policies, which the casinos tend to follow in order to attain hiked profits. One of the major concepts among few of their schemes, is to make the gamblers play higher number of rounds of the game being played, i.e., they are never interested on ***short-run bets***, but they always find interest in the potential ***long-run bets***.

On making a visit to the casinos for gambling, it is often observed that, the officials want to keep the gamblers in the game and allow them to play for a very long time.

But why do they do that?

The motive is to generate a large number of games, where each game has a contribution towards the house, i.e., having a particular percentage of house-edge. This percentage gets integrated to a very large sum, when higher number of games are being hosted by the casino, as a result of which the casino accounts for a larger portion of their profits earned from this contributing factor.

Therefore, one of the factors or rather a concept, due to which the world of gambling sees or witnesses a bizarre amount of high profits, is the ***Law Of Large Numbers***, a well-known theory in Statistics.

The Law of Large Numbers, dictates,

"Keeping other factors fixed or constant, as the sample size increases, results tend to

become more reliable and stable."

Thereby, concluding, that if games are being played on a ***Large Scale***, the contribution towards the casinos profit margin is high, as the integration to the house-edge becomes larger due to higher number of games, helping these gambling hosts to earn huge amount of money.

11.1 *A Simulation to show how gambling corners earn greater profits - Illustration Of Law Of Large Numbers in Gambling*

11.1.1 Motivation & Objective

This simulation is mainly concerned with the generation of the ***Expected Return*** on the part of the gambler from the initial capital, with which the gambler enters the game. The motive is to visualize the expected return when the gambler plays the game on a large scale, thereby determining the profits made by the casino.
N.B :: The game played by the gambler is considered to be a fair game, i.e., p = $\frac{1}{2}$.

11.1.2 The Algorithm

The user-built function ***'expected.return(ini.sum,p,ngames,sim)'***, simulates a game under the Gambler's Ruin Scenario taking into account the concerned parameters, and returning the ***expected return*** (in %) on the initial capital, with which the gambler started the game.

» The Function Parameters ::

- ***ini.sum*** :: The Initial Capital (in Rs.), with which the gambler enters the game.
- ***p*** :: The probability that the gambler wins a particular round of the game. We take p to be $\frac{1}{2}$, which is the case the case of an ***Unbiased Game***.

- ***ngames*** :: Number of games to be played.
- ***sim*** :: The number of times each game is repeated or simulated.

» ***Return*** :: A vector depicting the percentage return on the different choices of the initial capital considered.

11.1.3 Visualization & Observations

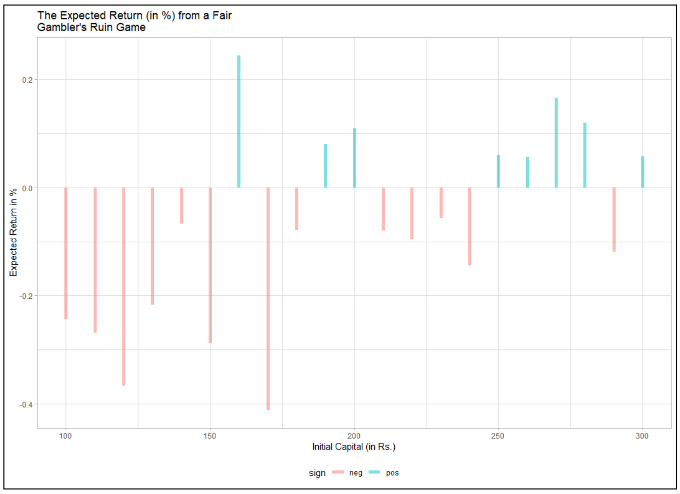

Figure 23: ***The Expected Return (in %) in a Gambler's Ruin Setup -*** $p = \frac{1}{2}$

» ***Comment*** :: Here lies the secret of the casinos earnings. even though we simulate a fair game under the ***Gambler's Ruin Setup***, on the long-run, i.e., when the gambler plays the game for a very long period of time, the ***Expected Return*** of the gambler tends to be ***more negative than positive***, although, the expected return being very close to 0, which is quite natural in an ***Unbiased Game***.

Henceforth, the casinos would want the gamblers to play for a very long time, so the portion of house-edge included associated to a game, greatly adds up to the whole day's profit earnings for the casino.

12 Applications

Because of the potential and advanced statistical concepts that forms the structural framework of the Gambler's Ruin Problem, the problem not only remains restricted to the world of gambling, having connections only with the chance games, but has developed its applicability on an interdisciplinary level, i.e., being applied both in the **practical** as well as the **theoretical** fields.

12.1 Gambler's Ruin in Stock Market & Insurance

» **The Stock Market** :: Consider a hypothetical situation where a stock market investor owns shares of a particular stock whose present worth is Rs.**X**. The investor decides to sell off his shares if the stock price goes up to 'Rs.X_{up}' or if it goes down to 'Rs.X_{low}'. Now each change of stock price is either **up** by Rs.**1** with probability **p** or is **down** by Rs.**1** with probability **1-p**, and the successive changes in the stock prices are assumed to be **independent**. Then the investor needs to get an idea of the probability that he/she retires a winner?
Here the above situation can be exactly modelled through the notion of Gambler's Ruin Problem, for which the **transition diagram** is given as follows.

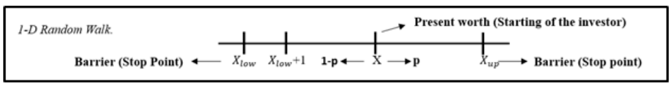

Figure 24: ***Transition Diagram for Stock Prices***

Note that, here both the points X_{low} & X_{up} are the winning points of the investor. Finding out the probability w.r.t one point would give us the other. Hence the Gambler's Ruin Problem, applied in this scenario helps the investor to get an idea about the conclusions of the investments made in the stock market.

» **Insurance** :: As the investor, investing in a particular stock, gets an highlight of the effectiveness of the investment done, the Gambler's Ruin problem is also applied in cases of **insurance**, where it guides an insurance company to know about the overall gain or loss of its own reserves.

Let us consider that an insurance company, starts initially with a reserve of **$10 millions**. Each day, either the **$ 1 million** of cash is added to the existing reserves, which has an associated probability of **p**, or **$ 1 million** of cash is deducted (loss) from the existing reserves. The question is, What is the probability that the insurance company will eventually go broke, i.e., lose all its reserves? [29]

This can be effectively modelled through the Gambler's Ruin setup, helping to address queries about the interest rates to be set, the premiums to be collected from the customers, etc. provided that the situation prevailing over the market is as described above.

12.2 Google's *PageRank* Algorithm through Gambler's Ruin Problem

Today, **Google** is one the most widely used search engines. The company was founded in 1998 by two Stanford graduate students. The original search algorithm known as, ***'PageRank'***, named after the co-founder **Larry Page**, was invented as a part of a research project. The company went public in 2004, and its parent company **alphabet** is valued today at $550 billions.

In today's world, other search algorithms are used in addition to the ***'PageRank'*** algorithm. The **goal** of search algorithms is to make a display of the search results in a particular pattern based on the search inputs and queries made by the user. Hence, given a finite set of web pages, each page is to be assigned a **non negative score** which represents the importance of that particular web page. The scores are assigned based on the link structure of the web. The search engines providing more applicable and appropriate results, are greatly preferred.

PageRank Algorithm as a Stochastic Process :: Suppose the web consists of **'n'** web pages, which are to be ranked. More specifically, the scores $\theta_1, \theta_2, ..., \theta_n$ are to be assigned to each of the n pages. The PageRank algorithm is a **stochastic process** model of a web-user who begins with a randomly chosen page and traverses his way online, randomly through the web. After a certain amount of browsing time, the user will eventually get bored and then would move to another randomly chosen site. Here the scores θ_i's represent the **long-run proportion of time spent** on each of the n web pages.

Initially, we assume that each web page has the same importance, i.e.,

$$P(X_0 = i) = \frac{1}{n} \; \forall \; i = 1, 2, ..., n$$

Consider the links as potential transitions. Now, since the user chooses the links at random, a page containing **'m'** outgoing links would assign each of these possible transitions a probability of $\frac{1}{m}$. [30]

Viewing PageRank Algorithm in terms of the Gambler's Ruin Problem :: Consider the **'n'** web pages to be the **'n'** gamblers in our setup, where the i^{th} gambler starts with an **initial capital** - $\theta_i \; \forall \; i$. This assignment of initial capital has an associated probability of $\frac{1}{n}$, which is same for all the gamblers under consideration. Now for a transition from one round to the other, we take into account the **'m'** outgoing links in each of the web page, that has an associated transition probability of $\frac{1}{m}$, i.e., a gambler can make a transition in any of the **'m'** possible ways from the current state (current round of the game), with an equal probability of $\frac{1}{m}$. The absorbed state in this particular setup, is the point where the web-user stops the browsing activity.

Thus we have a setup of the Gambler's Ruin Problem.

The **Transition Diagram** of the **PageRank** algorithm, with four web pages, namely Web A, Web B, Web C and Web D, is shown below,

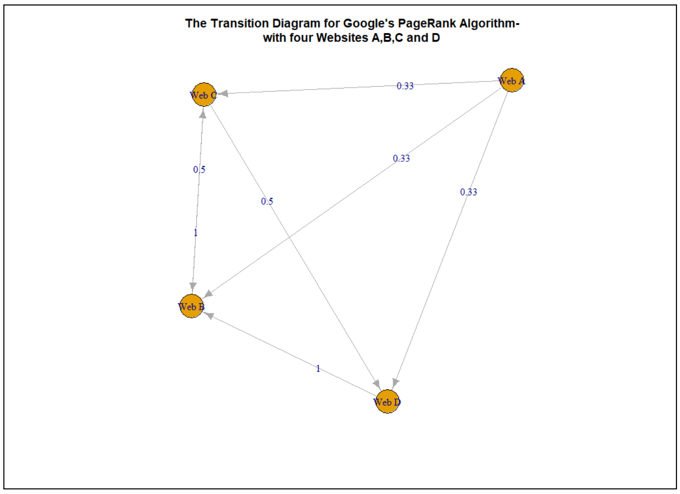

Figure 25: ***Transition Diagram for Google's PageRank Algorithm***

12.3 The Banach's Matchbox Problem as a case of Gambler's Ruin

Banach's Matchbox Problem, is one of the most popular problems discussed in the field of probability, which is attributed to **Stefan Banach**. Feller [31] associates the origin of the problem to the smoking habit that was developed by Banach, which made him eventually to come up with the problem as stated below.

» **Problem Statement** :: Suppose a mathematician always carries **2** match boxes initially each containing **'N'** match sticks. When a matchstick is required, the mathematician selects one of the two boxes at random and takes a stick out of it. What is the probability that one of the match boxes will be found empty when the other contains exactly **'r'** match sticks?

» **Re-framing and Modelling** :: Let us consider one of the matchboxes to be Gambler A and the other to be Gambler B, starting with an initial amount of Rs.**N** and Rs.**N** respectively, such that the total money in the game is Rs.**2N**. Here selection of any match box is equally likely that is with probability p=$\frac{1}{2}$ and the **match boxes are selected independently**. On selection of a match box, **1** matchstick is withdrawn from it implying that the Gambler A **loses** Rs.**1**. Hence, we consider **selection of a match box** to be a **loss** for Gambler A, and if Gambler A **wins** i.e., when the box is not selected then no money is withdrawn from Gambler A. The process of selecting match sticks stops when one of the two match boxes are found to be empty, i.e., when one of the two gambler gets ruined, indicating the **absorbed state** in the setup. Hence this is a **fair (unbiased) setup** of the Gambler's Ruin Problem, since selection of any matchbox takes place with equal probability, i.e., $\frac{1}{2}$.

N.B: In this application to the Gambler's Ruin Problem, if a Gambler wins, it is equivalent to a match box not getting selected by the mathematician, i.e., the Gambler does not add money to his existing stake figures, indeed his sum remains the same, i.e., if any matchbox is not selected, it implies that no matchstick is taken out of it, which is considered here to be analogous to the winning scenario of a gambler.
The following is a **transition diagram** under the setup formulated above with **N=4** matchsticks in each of the two matchboxes, i.e., a gambler starts the game with Rs.**4**.

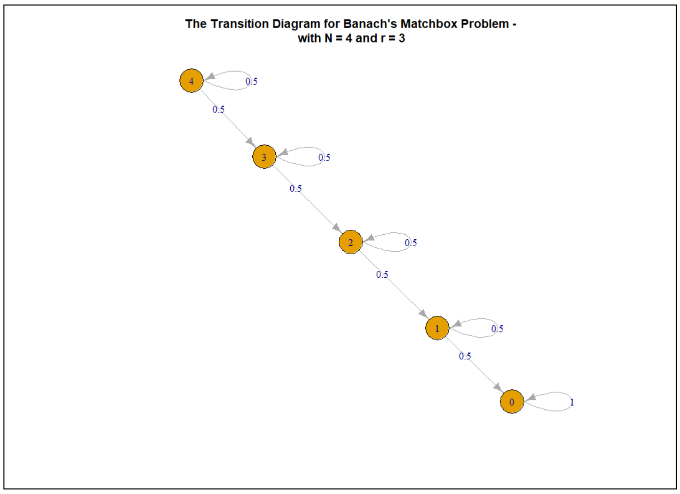

Figure 26: ***Transition Diagram for Banach's Matchbox Problem***

13 Conclusion

After studying the Gambler's Ruin Problem in details, it can be interpreted as the classic modelling of the scenario of a gambler playing a game of chance. It accounts for the winning and the losing probability of the gambler along with the trajectory of the capital with which the gambler starts the game, through various statistical as well as mathematical techniques and formulations as we have explored and discovered throughout the course of the paper.
Starting from fair (unbiased) games to biased ones, we have formulated and solved the long run probability of winning and losing the game as well the expected time to completion, considering two absorbed states under consideration.

The paper has closely associated the concepts of Stochastic Processes, Random Walks and Markov Chains, explaining their roles in the problem at hand.

Simulations have been done considering different dimensions of the problem, which has helped in the analysis of the matter on a deeper level. Along with the detailed analysis of the problem, simulations have also facilitated in the unfolding of the truth behind the games hosted by the casinos and various gambling corners, exploring the motive behind placing and maintaining a certain amount of house-edge in each and every gambling game conducted.

Using the problem at hand, applications have been encouraged both at a theoretical as well as a practical level.
Due to the immense gravity of the problem, prospective works and schedules have also been discussed, hoping for more advanced and revealing results, observations and conclu-

sions. The Gambler's Ruin problem has almost imparted a different mindset to the world of gambling through its applications, simplistic nature and the power to explain and justify the gambling world both to the gamblers as well as the gambling corners, giving them an opportunity to optimize and maximize their profits.
In other words, the problem has been successful in shaping the world of chance games in a brand new way.

14 Prospective Study & Future Work

The Gambler's Ruin Problem, as discussed throughout has unfolded many advanced aspects of both Statistics as well as Probability. Adding to the applications in the practical as well as the theoretical field. Following are some of the further areas and prospects of the problem at hand, which could be analyzed in the future.

- To deduce a relationship (possibly a **regression analysis**) between the **initial amount of money** with which the gambler enters the game and the **number of rounds** played by the gambler to win the game or get ruined.

- Since the variable - **number of rounds to be played** in the game until completion is a **count variable**, the above suggested regression analysis can be preferably a **Poisson Regression**, but a study could also be made to model the nature of the **number of rounds**, i.e., to develop any distribution through which the **number of rounds** to be played could be modelled. One of the possible ways is to simulate the number of rounds of a game, by repeating a particular game for a very large number of times, and then to analyze the nature of the available big data at hand.

- We could introduce a component of **Machine Learning** into the Gambler's Ruin Problem, which could be ground breaking, as it would help us to understand the betting patterns of the gambler, facilitating a close analysis of the gambler's mindset during the game. In the above simulations considered, we have always fixed the amount to be wagered during a particular round, which is a crude approach, but a model could be developed undertaking the parameter, which relates to the intention and the gambling approach of the gambler under consideration, eventually determining on its own, the amount which is to be wagered each round depending on the current capital and the mindset of the gambler.

- The problems of **optimization** could be considered as a field of analysis under the Gambler's Ruin setup. In other words, data could be collected from the games being hosted which obeys the dimensions as put forward by the Gambler's Ruin, and pattern recognition could be done, so as to determine the **optimal number of rounds** to be played to maximize on the capital or stake gambled, by a particular gambler.
  Considering the perspective of the gambling houses or the casinos, this optimization problem could also help them, as it helps the individual gamblers. The casinos could optimally design their games by studying and analysing the betting patterns and the gambling nature of the usual gamblers, who often pay visit to these gambling centers, thereby inducing a **higher house-edge** in the games being played under the Gambler's Ruin setup, enabling the casinos to earn greater profits.

15 Related Works - Developing an R-Package & a Web Application for simulating the Gambler's Ruin Problem

This section deals with two of the developments related to Gambler's Ruin Problem. An R-Package and a Web Application has been developed to reach out among the readers, in turn spreading the importance of the problem at hand in explaining many real world situations as well as the theoretical integrity associated with the problem.

15.1 The R-Package : '*gamblers.ruin.gameplay*' [1]

15.1.1 Motivation & Objective

The objective of this developed package (hosted in **CRAN**) is to simulate a gambling game under the Gambler's Ruin framework as mentioned in this paper, thereby giving a visual representation of the trajectory of money with the gambler at each round of the game. The package also determines the **overall probability of the gambler winning the entire game being simulated or played**.

15.1.2 The Algorithm

The master function ***grp.gameplay(ini.stake,p,win.amt)*** takes user input for the initial capital, with which the gambler wishes to enter the game, the probability of the gambler winning each round of the game - p and lastly the target amount which the gambler wishes to earn from the game being played. Taking these inputs the game is simulated, eventually notifying the user about the overall probability of winning the game, the number of rounds played and a feedback about the result of the game.

15.1.3 Application of the Package

Suppose a gambler enters the game with 100 units of money and wishes to earn 500 units of money from the game. The gambler prefers to play an unbiased or fair game, i.e., $p = \frac{1}{2}$. Applying the package yields,

```
# Loading the required package
library(gamblers.ruin.gameplay)

# running the required function
grp.gameplay(100,0.5,500)
```

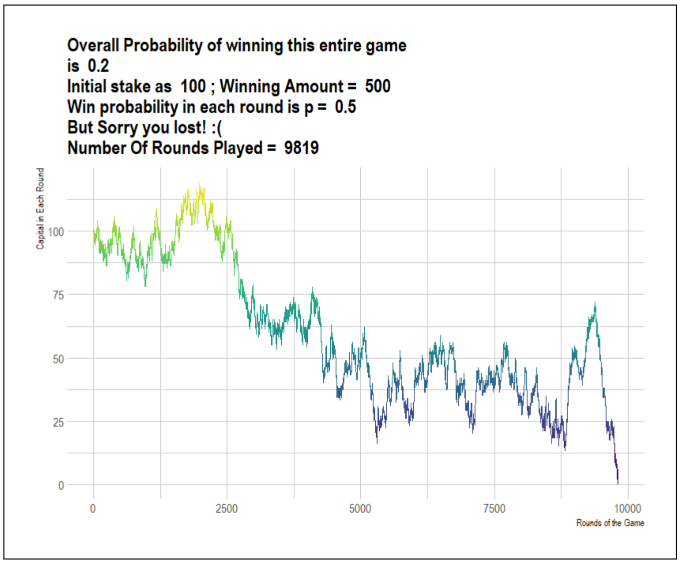

Figure 27: ***The Game Trajectory***

The above game results into a **loss** for the gambler, with an overall probability of winning the game for the gambler as **0.20**. Also the gamblers continues to play for a total of **9819** rounds.

15.2 The Web Application - Gambler's Ruin Simulator

15.2.1 Motivation & Objective

As an extension to the above developed package, the intention behind developing this interactive web application was to spread the beauty that Gambler's Ruin Problem entails. It helps a web-user to play an online simulation of the Gambler's Ruin Problem, by inputting the money the user has, the money the user wishes to earn from the game and also the probability of winning each round of the game. Through the user's input, the application runs a simulation of the problem and displays the appropriate results of the game.

Apart from the interactive platform, which the application provides, it also enables the users to cite an example of the major concepts related to Statistics and Probability, viz., a **Stochastic Process**, a **One-Dimensional Random Walk** as well as a **Markov Chain**.

15.2.2 Visuals of the App

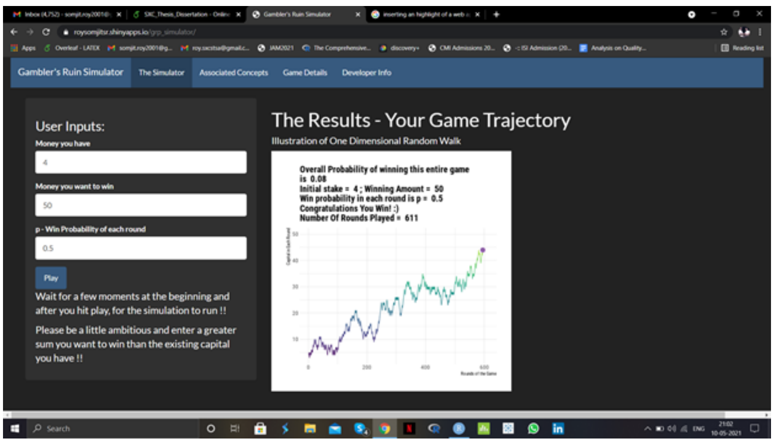

Figure 28: ***A view of the Web App - The Gambler's Ruin Simulator***

It allows you to change your **initial sum**, the **amount of money you desire to win** as well as the **probability ('p')** of winning each round of the game, thereby simulating the gambler's ruin problem for you.

15.2.3 The Development

The application has been developed by using the **'shiny'** [32] and **'shinythemes'** [33] library in **R**, which helps to build and create, useful and interactive web applications.

The stage of the web app development is mainly controlled by the integration of **user-interface** :: **ui.R** and **server interface** :: **server.R**, as follows,

```
# Define UI
ui <- fluidPage(.....
..........
..........
) # fluidPage
# Define server function
server <- function(input, output)
{
  .........
} # server
# Create Shiny object
shinyApp(ui = ui, server = server)
```

The second stage entails **hosting** the above developed app, which has been done through a free platform **Shiny Apps** - https://www.shinyapps.io/.

Play the game at :: https://roysomjitsr.shinyapps.io/grp_simulator/.

References

[1] Somjit Roy. *gamblers.ruin.gameplay: One-Dimensional Random Walks Through Simulation of the Gambler's Ruin Problem*. R package version 4.0.5, URL : https://cran.r-project.org/package=gamblers.ruin.gameplay.

[2] Giorgio Alfredo Spedicato. Discrete time markov chains with r. *The R Journal*, 07 2017. R package version 0.6.9.7,URL:https://journal.r-project.org/archive/2017/RJ-2017-036/index.html.

[3] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2021, URL:https://www.R-project.org/.

[4] LLC, a Salesforce Company. Tableau public, version : 2021.1.1, date : 22-04-2021, url:https://public.tableau.com/en-us/s/.

[5] Hadley Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016, URL:https://ggplot2.tidyverse.org.

[6] Garnier, Simon, Ross, Noam, Rudis, Robert, Camargo, Antônio Pedro, Sciaini, Marco, Scherer, and Cédric. *viridis - Colorblind-Friendly Color Maps for R*, 2021. R package version 0.6.0, URL:https://sjmgarnier.github.io/viridis/.

[7] Bob Rudis. *hrbrthemes: Additional Themes, Theme Components and Utilities for 'ggplot2'*, 2020. R package version 0.8.0, URL:https://CRAN.R-project.org/package=hrbrthemes.

[8] Baptiste Auguie. *gridExtra: Miscellaneous Functions for "Grid" Graphics*, 2017. R package version 2.3, https://CRAN.R-project.org/package=gridExtra.

[9] Hadley Wickham, Mara Averick, Jennifer Bryan, Winston Chang, Lucy D'Agostino McGowan, Romain François, Garrett Grolemund, Alex Hayes, Lionel Henry, Jim Hester, Max Kuhn, Thomas Lin Pedersen, Evan Miller, Stephan Milton Bache, Kirill Müller, Jeroen Ooms, David Robinson, Dana Paige Seidel, Vitalie Spinu, Kohske Takahashi, Davis Vaughan, Claus Wilke, Kara Woo, and Hiroaki Yutani. Welcome to the tidyverse. *Journal of Open Source Software*, 4(43):1686, 2019.

[10] Hadley Wickham. *tidyr: Tidy Messy Data*, 2021. R package version 1.1.3, URL:https://CRAN.R-project.org/package=tidyr.

[11] Hadley Wickham, Romain François, Lionel Henry, and Kirill Müller. *dplyr: A Grammar of Data Manipulation*, 2021. R package version 1.0.5, URL:https://CRAN.R-project.org/package=dplyr.

[12] The Calcutta Electric Supply Corporation (India) Ltd. (CESC). Lt consumer bill history, https://www.cesc.co.in/?page_id=1606.

[13] Florence Nightingale David. *Games, Gods, and Gambling : A History Of Probability and Statistical Ideas*. Courier Dover Publications, 1998.

[14] J.W.F. Edwards. Pascal's Problem : The Gambler's Ruin. *Revue Internationale de Statistique*, pages 73–79, April 1983.

[15] Jan Gullberg. *Mathematics from the birth of numbers*. W. W. Norton Company, 1997.

[16] Frederick Mosteller. *Fifty Challenging Problems in Probability*. Dover Publications, 1965.

[17] Charles J.Stone Paul G.Hoel, Sidney C.Port. *Introduction to Stochastic Processes*. Houghton Mifflin Company, Boston, University of California, Los Angeles, 1972.

[18] National Stock Exchange (NSE). Nifty 50 stock prices, https://www.nseindia.com/.

[19] J Medhi. *Stochastic Process*. New Academic Science Limited, Guwahati University, India, 2012.

[20] Gordan Žitkovic. *Introduction to Stochastic Processes - Lecture on Random Walks*. Department of Mathematics, University of Texas at Austin.

[21] Sahil Singla. Advanced algorithm design - random walks, markov chains and how to analyse them, https://www.cs.princeton.edu/~smattw/teaching/fa19lectures/lec12/lec12.pdf.

[22] Kai Lai Chung. *Elementary Probability theory with Stochastic Processes*, chapter 8 - From Random Walks to Markov Chains. Narosa Publishing House Pvt Ltd, Stanford University, USA.

[23] Anders Tolver. *An Introduction to Markov Chains*. Department of Mathematical Sciences, University of Copenhagen, Denmark, 2016.

[24] S Roy. The code for ant trajectory - a markov chain, https://github.com/roy-sr-007/the-ant-trajectory---illustration-of-markov-chains.git.

[25] Ronitt Rubinfield Tom Leighton. *Random Walks*. Computer Science and Artificial Intelligence Laboratory (CSAIL), Massachusetts Institute of Technology.

[26] Sheldon Ross. *A First Course in Probability*, chapter 9 - Simulation. Pearson, University of Southern California.

[27] Caasino News Daily. European roulette, https://www.casinonewsdaily.com/roulette-guide/european-roulette/.

[28] Jeremiah Booker. 6 secrets about casinos, https://www.bestuscasinos.org/blog/6-secrets-the-casinos-dont-want-you-to-know/.

[29] Jason Carlson. About gambler's ruin, https://jxxcarlson.medium.com/about-gamblers-ruin-6c30903bbbe3.

[30] Martina Bremer. Application to gambler's ruin problem, http://www.math.sjsu.edu/~bremer/teaching/math263/lecturenotes/lecture04.pdf.

[31] William Feller. *An Introduction to Probability Theory And its Applications*, chapter VI, Section 8. Wiley, 1968.

[32] Winston Chang, Joe Cheng, JJ Allaire, Carson Sievert, Barret Schloerke, Yihui Xie, Jeff Allen, Jonathan McPherson, Alan Dipert, and Barbara Borges. *shiny: Web Application Framework for R*, 2021. R package version 1.6.0, URL:https://CRAN.R-project.org/package=shiny.

[33] Winston Chang. *shinythemes: Themes for Shiny*, 2021. R package version 1.2.0, URL: https://CRAN.R-project.org/package=shinythemes.

16 Appendix

Appendix A :: R-Codes for Master Functions used to develop the Simulations

Few of the potential **R-Codes** used to develop the simulations stated in this paper are given below.

Simulation I - Simulating Solution of the GRP :: Code for grp.win.prob()

```r
set.seed(987654321) # To maintain the uniformity of results obtained.
grp.win.prob = function(n1,n2,p,ngames,r)
{
  # ngames :: The number of games to be simulated.
  # r :: The number of times each round is to be simulated.
  # p :: Probability of A winning a round of the game.
  ct.A=0
  win.prob.A=array(dim=1) # To store the win probability of gambler A.
  ct.B=0
  win.prob.B=array(dim=1) # To store the win probability of gambler B.
  for(i in 1:ngames)
  {
    for(j in 1:r)
    {
      n1.A = n1           # Initial Capital of Gambler A.
      n2.B = n2           # Initial Capital of Gambler B.
```

```
      while(n1.A > 0 & n2.B > 0) # Absorbed States.
      {
        if(rbinom(1,1,p) == 1) # A wins.
        {
          n1.A = n1.A + 1
          n2.B = n2.B - 1
        }else{                 # B wins.
          n1.A = n1.A - 1
          n2.B = n2.B + 1
        }
      }
      if(n2.B > 0){ct.B = ct.B+1}
      if(n1.A>0){ct.A = ct.A+1}
    }
    # proportion of games, A wins in these 'r' simulations.
    win.prob.A[i] = ct.A/r
    # proportion of games, B wins in these 'r' simulations.
    win.prob.B[i] = ct.B/r
    ct.A = 0
    ct.B = 0
  }
  df = data.frame(prob.A = win.prob.A,prob.B = win.prob.B,Games = 1:ngames)
  # Returning the win probabilities for both the gamblers A and B.
  return(df)
}
```

Simulation III - Simulating the Expected Duration Of Play :: Code for grp.ruin.expected.rounds()

```
set.seed(987654321) # To maintain uniformity of results obtained.

gamblers.ruin.expected.rounds = function(n1,n2,p,ngames)
{
  # ngames :: The number of games to be simulated.
  # p :: Probability of A winning a round of the game.
  n.rounds = array(dim=1) # To store the no.of rounds to be played.

  for(i in 1:ngames)
  {
    ct.round = 0
    # Initial capitals of Gamblers A and B.
    n1.A = n1; n2.B = n2

    while(n1.A>0 & n2.B>0) # Absorption States.
    {
      ct.round = ct.round + 1
      res = rbinom(1,1,p)
      if(res == 1)          # A wins.
      {
        n1.A = n1.A + 1
        n2.B = n2.B - 1
      }else                 # B wins.
```

```
      {
        n2.B = n2.B + 1
        n1.A = n1.A - 1
      }
    }
    n.rounds[i] = ct.round
  }
  # Returning the expected number of rounds to be played each game,
  # until one of the Absorption States are hit.
  return(n.rounds)
}
```

Simulation V - Simulation of the European Roulette game - The N-Player Ruin Game :: Code for bet.colors()

```
set.seed(987654321) # To maintain the uniformity of the results obtained.

bet.colors = function(capital,col,wager.cap)
{
  # capital :: Initial capital of the gambler.
  # col :: The color on which the bet is placed.
  #wager.cap :: The amount of money wagered each bet.
  flag = 2; sum = capital
  x = sample(0:36,1)               # Spinning the Roulette.
  if((x > 0) & (x %% 2 == 0))      # red has appeared.
  {
```

```
      flag = 1
    }else if((x>0) & (x %% 2 != 0)) # black has appeared.
    {
      flag = 0
    }else
    {
      flag = -1
    }
    if(col == flag)                  # I win the game.
    {
      sum = sum + (wager.cap)
    }else                            # I lose the game.
    {
      sum = sum - (wager.cap)
    }
    return(sum)                      # The sum after one round of the game.
  }

  gameplay = function(ngamblers,sim)
  {
    # ngamblers :: Number of gamblers participating in the game.
    # sim :: The number of times, each game to be simulated.
    indicator = array(dim=1)
    prob.win = array(dim=1); prob.lose = array(dim=1)
    for(r in 1:sim)
    {
```

```
    for(i in 1:ngamblers)
    {
      capital = 500               # Each gambler starting with Rs.500.
      limit = capital
      upper.limit = 3*capital     # Upper limit to be attained
                                  # by the gambler.
      flag = 0
      count = 0
      while(capital>0)
      {
        count = count + 1
        col = sample(c(0,1),1)    # Sampling the color on which the bet
                                  # is to be placed.
        wager.cap = capital/4     # Setting the wager amount.
        capital = bet.colors(capital,col,wager.cap) # Calling bet.colors().
        if(capital >= upper.limit | capital < limit*(10^-2))
        # Taking care of the two Absorbed States.
        {
          if(capital >= upper.limit)
          {
            flag = 1
          }
          break
        }
      }
      indicator[i] = flag
```

```
    }
    # Proportion of gamblers attaining atleast the upper limit
    # of money set during the game.
    prob.win[r] = length(which(indicator==1))/ngamblers
    # Proportion of gamblers getting ruined due to insufficient
    # capital.
    prob.lose[r] = length(which(indicator==0))/ngamblers
  }
  df = data.frame(prob.win,prob.lose)
  return(df)
}
```

Appendix B :: Discussion

Following is the questions which is often posed and could come to the reader's mind while glancing through the paper.

Question :: Can the game under the Gambler's Ruin Setup continue forever?

The setup of the Gambler's Ruin Problem considered in this paper has the presence of **two absorbed states**, i.e., those points where the game stops. Hence for an unbiased as well as the biased nature of the games, we have done away with the possibility of an infinitely long game. Now we consider two possibilities.

Possibility I :: It is to be noted that, if $p \neq \frac{1}{2}$, i.e., the game being played is biased, then there is a possibility that, it can continue for an infinitely long number of rounds, under the assumption that the gamblers involved in the game do not plan to quit until they get ruined.

Possibility II :: The most important query is, will a fair game go on forever?

The **claim** is that, even though if a gambler enters the game with a fair chance of winning each round, i.e., here $p = \frac{1}{2}$, and possess and infinite amount of wealth, say a whooping 100000 million dollars, the gambler playing the game will eventually go broke. Also, if the gambler continues to play the fair game under the Gambler's Ruin setup and chooses not to quit, at some point of time, the gambler will eventually go bankrupt (get ruined).

Let us justify our claim mathematically.

$$\Rightarrow \mathbb{P}(\text{going broke}) \geq \mathbb{P}(\text{going broke before reaching } n_1 + n_2) \ \forall \ n_1 + n_2$$

$$= 1 - \mathbb{P}(\text{reach } n_1 + n_2 \text{ before going broke})$$

$$= 1 - \frac{\text{Initial Capital}}{n_1 + n_2} \text{ (since a fair game)}$$

Hence, we have,

$$\Rightarrow \mathbb{P}(\text{going broke}) \to 1 \text{ as } n_1 + n_2 \to \infty$$

Therefore, both with an infinite amount of starting capital and the desire to not quit the game, is not of much help in an unbiased game, as the gambler will surely go broke.

17 Glossary Of Real Life Situations/Examples and Simulations

1. ***Illustration Of Electricity Consumption*** to explain the idea of ***Randomness*** in Section **1.1 - Page 2**.

2. ***A Real World Problem Of Gambling*** to give a glimpse of the Gambler's Ruin Scenario in Section **3.2** - Page **6**.

3. ***Closing Stock Prices Of Nifty-50 from 01/01/2021 to 21/04/2021*** as an example of ***Stochastic Process*** in Section **5.1** - Page **11**.

4. ***The Movement of Gas Molecules in motion***, to illustrate Random Walks has been cited with the help of a link in Section **5.2** - Page **12**.

5. ***An Example Of One-Dimensional Random Walk illustrated through simulating Movements of a Drunkard*** in Section **5.3** - Page **13**.

6. ***Simulating the Movements of an Ant in*** $\mathbb{R}^2$ ***plane - The Ant Trajectory***, to illustrate Markov Chains in Section **5.5** - Page **18**.

7. ***Simulating the Solution of the Gambler's Ruin Problem***, aimed at visualizing the Convergence of the probability of winning a game - Section **10.1** - Page **41**.

8. ***A Simulation to represent the Trajectory Of Money at each round of the game*** - Section **10.2** - Page **45**.

9. ***Simulating the Expected Duration Of Play under the Gambler's Ruin Setup***, illustrating the expected number of rounds to be played before hitting one of the Barrier Points - Section **10.3** - Page **48**.

10. ***Simulating and Illustrating the association between Expected Duration Of Play & Initial Capital with the Gambler*** - Section **10.4** - Page **52**.

11. ***Simulating the Game Of European Roulette***, an illustration of ***N-Player Ruin Game***, i.e., the ***N-Dimensional Gambler's Ruin Problem*** - Section **10.5** - Page **55**

12. ***Simulation of the Expected Return in an Unbiased Game under the Gambler's Ruin Setup***, to highlight the Casino's profit margin - Section **11.1** - Page **61**

Somjit Roy

Department Of Statistics

St.Xavier's College (Autonomous), Kolkata

West Bengal, India

https://orcid.org/0000-0002-3012-8430

roy.sxcstsa@gmail.com

https://github.com/Roy-SR-007

linkedin.com/in/somjit-r-6b0965114