Probability Simulations

Moti Ben-Ari

http://www.weizmann.ac.il/sci-tea/benari/

February 14, 2023

Contents

Introduction

Simulations are an excellent way of understanding probability, especially, the behavior of process of long duration. These programs enable the user to perform experiments by varying the parameters of problems and analyzing the results, both printed and displayed in graphs. A level of knowledge of probability equivalent to the first few chapters or [2] or [5] is assumed.

Section 1 presents the *Gambler's ruin problem*. Two players A and B divide a finite capital (amount of money between them) and they play a game in which A probability of winning is p and B's is $1 - p$. The loser gives one unit to the winner. The questions are: Given initial parameters, what is the probability that A wins? What is the expectation of the duration of the game?

Technical notes

The programs are written in the Python 3 language and use the `matplotlib` to generate the graphs. Parameters directly related to the problems, such as the probability of success, can be modified interactively. Others, related to the simulation, such as the number of steps in a simulation and the properties of the histograms, are defined in a module `configuration.py` that can be modified.

You need to install the Python (`https://www.python.org/downloads/`) although a knowledge of Python programming is not necessary.

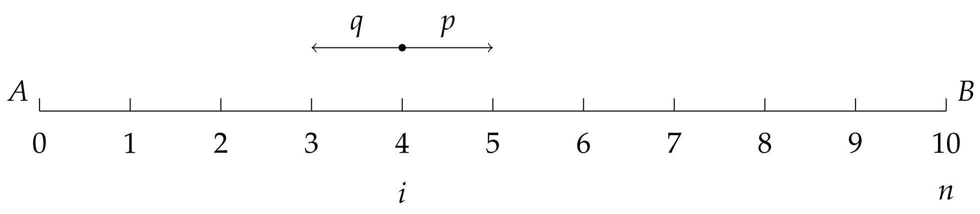
To run in the Visual Studio Code environment, ensure that `Code Runner` is installed. In the extension settings disable `Preserve Focus` and enable `Run In Terminal`.

To run in the IDLE or Thonny environments, change the configuration constant `CLOSE` to true. When the simulation is run multiple times, you have to close each figure before running a new simulation.

1 Gambler's Ruin

Problem Two players A and B compete in a contest. There is an initial finite capital of n units: A has i and B has $n-i$. They repeated play a game such that the probability that A wins is p and the probability that B wins is $q = 1-p$. The loser gives one unit to the winner. When one player has n units the contest is finished.

1. Given initial parameters (p, n, i), what is the probability that A wins?
2. What is the expectation of the duration of the game?



The gambler's ruin is presented in most books on probability such as [2, Section 2.7.2], [5, Section 3.4],[3], [1]. [4, Chapter 2] has a more extensive discussion which includes the solution to the expectation of the duration of the contest.[1]

1.1 Theoretical solutions

Given (p, n, i) the probability that A wins the contest is:

$$P_A(p, n, i) = \left(\frac{1-r^i}{1-r^n}\right),$$

where $r = q/p$. By symmetry, the probability that B wins is:

$$P_B(p, n, i) = \left(\frac{1-(1/r)^{n-i}}{1-(1/r)^n}\right).$$

There are separate solutions for $p \neq 1/2$ and $p = 1/2$. For $p \neq 1/2$ the expectation of the duration of the contest is:

$$P_{duration}(p, n, i) = \frac{1}{q-p}\left(i - n\frac{1-r^k}{1-r^n}\right).$$

For $p = 1/2$ the expectation of the duration of the contest is:

$$P_{duration}(p, n, i) = i(n-1).$$

Of course the duration does not depend on which player wins. If A wins, the contest terminates for B also, and conversely.

[1]Privault presentation asks for the probability that A is ruined, that is, that B wins. I follow other references who ask for A's probability of winning.

1.2 Program structure

The program is divided into three modules:

- `configuration.py` contains declarations of variables which are intended to be constants.
- `gambler_plot.py` contains the functions for plotting the histogram of the duration of the contests. If the simulation is run for multiple probabilities or initial values, a graph of the probability of wins is also displayed. The module imports `matplotlib.pyplot`.
- `gamblers_ruin.py` is the main program which obtains the parameters, runs the simulations and prints the output.

1.3 Running the simulations

The program runs the simulations in a loop, each time asking the user how to run it. You can run the same simulation again with the saved parameters, enter new parameters, or run a sequence of simulations for a range of probabilities or initial values.

A typical output is as follows:

```
Probability = 0.450, capital = 20, initial = 10
Wins = 127, losses = 873, limits exceeded = 0
Proportion of wins     = 0.1270
Probability of winning = 0.1185
Average duration  = 78
Expected duration = 76
```

The results of the simulation are very close to the theoretical probability and duration.

Figure 1 shows the histogram for the duration of the contest, limited to a duration of 200. The vertical line is the expectation.

Figure 2 proportion of wins for multiple probabilities. It also shows histograms for the duration of the contest for these probabilities.

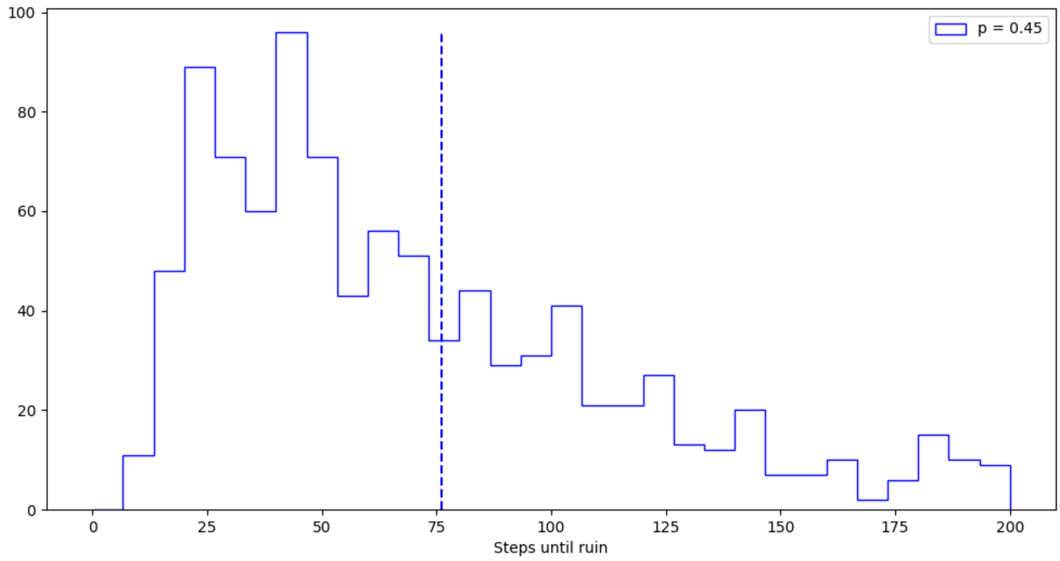


Figure 1: Histogram for $p = 0.45, n = 20, i = 10$

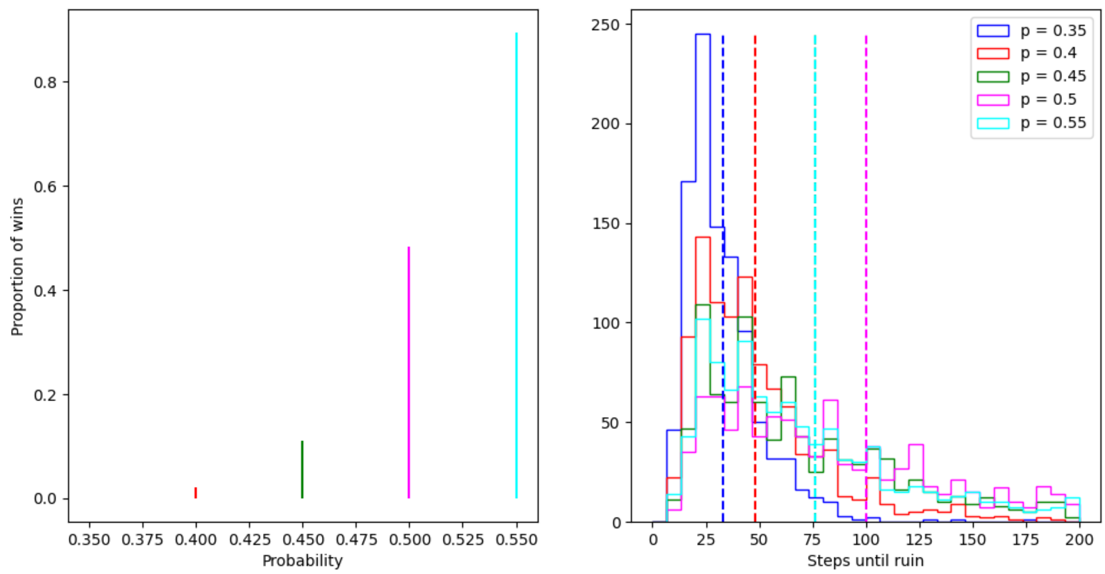


Figure 2: Proportion of wins and histogram for $n = 20, i = 10$ and multiple probabilities

References

[1] Moti Ben-Ari. Mosteller's challenging problems in probability. `https://github.com/motib/probability-mosteller`.

[2] Joseph K. Blizstein and Jessica Hwang. *Introduction to Probability (Second Edition)*. CRC Press, 2019.

[3] Frederick Mosteller. *Fifty Challenging Problems in Probability with Solutions*. Dover, 1965.

[4] Nicolas Privault. *Understanding Markov Chains: Examples and Applications (Second Edition)*. Springer, 2018.

[5] Sheldon Ross. *A First Course in Probability (Tenth Edition)*. Pearson, 2019.