Mass function of the number of coin flips to get m consecutive heads

Jan Aalmoes
jan.aalmoes@protonmail.com

January 23, 2024

Abstract

In this paper, we compute the exact probability mass function of the number of coin flips to get m consecutive heads. Meaning that we take D, a random variable modeling the number of coin flips until the coin gives m heads in a row; then we will explicit a function f such that for all n, $f(n) = P(D = n)$. Where $P(D = n)$ is the probability that the m last flips among n total flips are heads while the $n - m$ first flips doesn't contain a string of m uninterrupted heads. Using f we compute the cumulative distribution function (CDF) of D as well as $E(D)$: the expected number of flips. In addition we provide a python program to compute f, the CDF of D and the average of D at the following repository: git.jaalmoes.com/coin_flip.

1 Introduction

Repeating events and their occurrences is of interest in many fields: sports, physics, network, etc. In baseball, successfully repeating an exceptional action such as hitting a home run can be decisive for a player's carrier [1]. In reliability engineering, redundancy is used to reduce the probability of system failure [4]. Hence studying the distribution of consecutive failure of redundant parts allows a deeper understanding of the risk of full system failure. In networking, the wildly used transmission control protocol (TCP) uses re transmission to manage packet loss. Consecutive packet loss perturbs the re transmission scheme [8], hence studying the odds and frequencies of losing multiple packets in a row is a key aspect of building reliable networks.

There are many problems considering streaks for instance the number of streaks among n tries, or the waiting time in-between streaks. In this paper we choose to study the number of tries before a streak of size m. In reliability engineering it corresponds to the number of time a system can be used before full failure. This problem is usually studied either through simulation or by using the inclusion exclusion principle and the Bonferroni inequality to find an

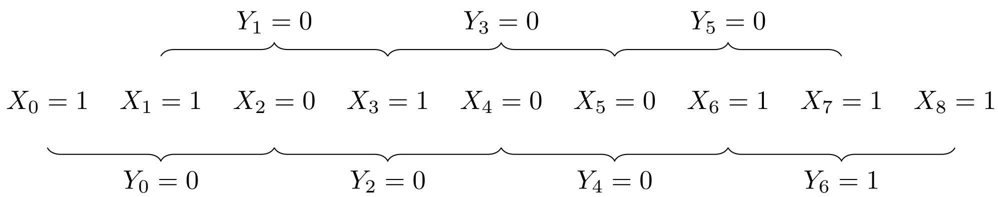

Figure 1: Repeating coin flip until $m = 3$ heads on a row

approximation [4]. The issue with numerical simulations is computing time and eventual cost of the simulation if real-world experiments have to be made [3].

In this work, we compute an exact formula of the probability law of the number of tries before a streak of size m. The formula's computation relies on two key algorithms: root finding [7] and matrix inversion. The degree of the polynomial and the rank of the matrix is m hence complexity of the numerical method depends on m. We use Numpy's implementations for numerical matrix inversion and root finding [6].

2 Problem definition

We consider a coin with a probability $p \in [0, 1]$ to land heads up. Let $X_0, \cdots, X_{n-1}$ be n random variables independent and identically distributed that follow a Bernoulli law of probability $p \in [0, 1]$. Each X_i models one coin flip and takes the value 1 for head and 0 for tail. Let $m \in \{1, \cdots, n\}$. We are interested in computing the probability of m consecutive heads. We model those events using the following vector of random variables :

$$\forall i \in \{0, \cdots, n - m + 1\} \quad Y_i = \prod_{j=i}^{i+m-1} X_i$$

With this model, the ith collection of m consecutive coin flips result in only heads if and only if $Y_i = 1$. On the contrary, if $Y_i = 0$ at least one the m coin flips ends up on tail. We remark that even if (X_i) is mutually independent, (Y_i) is not. This setup is represented in figure 1.

3 Mass function

The probability to obtain m heads on a row after flipping the coin n times and not before is the probability of the following event : "The first collection of m flips to give only heads is the $(n - m + 1)$th". Let $f(n)$ be the probability of this event. We start by computing $f(n)$ for $n \leq 2m$.

$$f(n) = \begin{cases} 0 & \text{if} \quad n < m \\ p^m & \text{if} \quad n = m \\ qp^m & \text{if} \quad m < n \leq 2m \end{cases}$$

Let's now study the case where $n > 2m$.

$$\begin{aligned}
& P\left(Y_0 = 0 \cap \cdots \cap Y_{n-m-1} = 0 \cap Y_{n-m} = 1\right) && (1)\\
=& P\left(\bigcap_{i=0}^{n-m-1} Y_i = 0 \cap \bigcap_{i=0}^{m-1} X_{n-m+i} = 1\right) && (2)\\
=& P\left(\bigcap_{i=0}^{n-2m-1} Y_i = 0 \cap X_{n-m-1} = 0 \cap \bigcap_{i=0}^{m-1} X_{n-m+i} = 1\right) && (3)\\
=& P\left(\bigcap_{i=0}^{n-2m-1} Y_i = 0\right) P\left(X_{n-m-1} = 0 \cap \bigcap_{i=0}^{m-1} X_{n-m+i} = 1\right) && (4)\\
=& P\left(\bigcap_{i=0}^{n-2m-1} Y_i = 0\right)(1-p)p^m && (5)
\end{aligned}$$

Let's now study the sequence $u_n = P\left(\bigcap_{i=0}^{n} Y_i = 0\right)$.

Case 1 $n = 0$ $\quad u_0 = P(Y_0 = 0) = 1 - P(X_0 = 1 \cap \cdots \cap X_{n+m-1} = 1)$. Since $(X_n)_{n\in\mathbb{N}}$ is mutually independent, $u_0 = 1 - p^m$.

Case 2 $n \leq m$

$$u_n = \left(1 - P\left(Y_n = 1 \quad | \quad \bigcap_{i=0}^{n-1} Y_i = 0\right)\right) u_{n-1} = u_{n-1} - (1-p)p^m$$

In this case we can solve the sequence :

$$u_n = 1 - p^m - n(1-p)p^m$$

Case 3 $n > m$ $\quad$ According it Bayes' formula

$$u_n = P\left(\bigcap_{i=0}^{n} Y_i = 0\right) = P\left(Y_n = 0 \quad | \quad \bigcap_{i=0}^{n-1} Y_i = 0\right) u_{n-1}$$

Since Y_n takes values in $\{0, 1\}$,

$$u_n = \left(1 - P\left(Y_n = 1 \quad | \quad \bigcap_{i=0}^{n-1} Y_i = 0\right)\right) u_{n-1} \qquad (6)$$

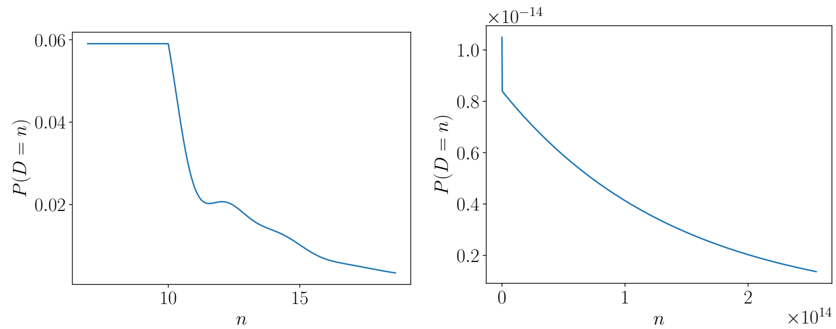

Figure 2: Mass function for $p = 0.9$, $m = 5$ on the left and $p = 0.2$, $m = 20$ on the right.

To compute $P\left(Y_n = 1 \quad | \quad \bigcap_{i=0}^{n-1} Y_i = 0\right)$ we use once more Bayes' formula and obtain :

$$\frac{P\left(Y_n = 1 \cap \bigcap_{i=0}^{n-1} Y_i = 0\right)}{u_{n-1}}$$

By using the same method as in equations 1 to 5 and by substitution in equation 6

$$u_n = u_{n-1} - (1-p)p^m u_{n-m-1}$$

By using the initial conditions described in cases 1 and 2 we can solve the sequence in case 3 by first finding the roots of the characteristic polynomial $Q(\lambda) = \lambda^{m+1} - \lambda^m + (1-p)p^m$ for $\lambda \in \mathbb{C}$. Let those roots be $\lambda_0, \cdots, \lambda_m$. Then $u_n = \sum_{i=0}^{m} c_i \lambda_i^n$ where $(c_i)_{i \in \{0,\cdots,m\}} \in \mathbb{C}^{m+1}$ are found by using the initial conditions of cases 1 and 2. Namely :

$$\begin{pmatrix} c_0 \\ \vdots \\ c_m \end{pmatrix} = \Lambda^{-1} \begin{pmatrix} u_0 \\ \vdots \\ u_m \end{pmatrix} \qquad \text{with } \Lambda = \begin{pmatrix} \lambda_0^0 & \cdots & \lambda_m^0 \\ \vdots & \vdots & \vdots \\ \lambda_0^m & \cdots & \lambda_m^m \end{pmatrix}$$

Finding Λ and inverting it can be done on a case by case basis numerically. Hence for all $n \in \mathbb{N}^*$, we have

$$f(n) = \begin{cases} 0 & \text{if} \quad n < m \\ p^m & \text{if} \quad n = m \\ (1-p)p^m & \text{if} \quad m < n \leq 2m \\ u_{n-2m-1}(1-p)p^m & \text{if} \quad n \geq 2m+1 \end{cases}$$

We display f for $m = 5$ and $m = 20$ on figure 2.

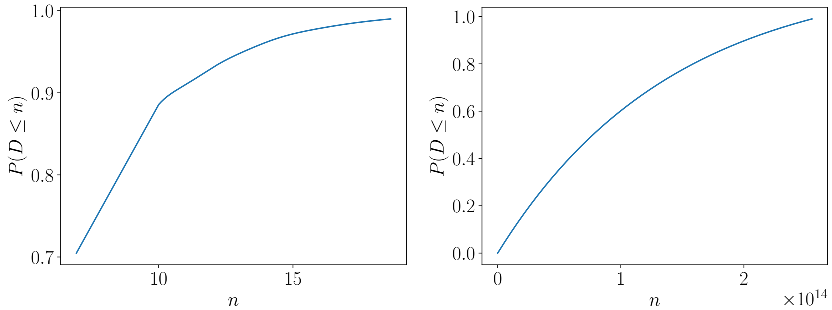

Figure 3: CDF for $p = 0.9$, $m = 5$ on the left and $p = 0.2$, $m = 20$ on the right.

4 Cumulative distribution function

We build the random variable D such that $\forall n \in \mathbb{N}\ P(D_n) = f(n)$. Let F be the CDF of D: for all $N \in \mathbb{N}^*$, $F(N) = P(D \leq N)$ Hence

$$F(N) = \sum_{n=1}^{N} f(n) = \begin{cases} 0 & \text{if} \quad N < m \\ p^m & \text{if} \quad N = m \\ p^m + (N-m)(1-p)p^m & \text{if} \quad m < N \leq 2m \\ p^m + m(1-p)p^m + (1-p)p^m \sum_{n=1}^{N-2m-1} u_n & \text{if} \quad N < m \end{cases} \tag{7}$$

Let's compute on the side the sum of u_n. Let $n \in \mathbb{N}$

$$\sum_{i=0}^{n} u_i = \sum_{i=0}^{n} \sum_{j=0}^{m} c_j \lambda_j^i = \sum_{j=0}^{m} c_j \sum_{i=0}^{n} \lambda_j^i = \sum_{j=0}^{m} c_j \frac{1-\lambda_j^{n+1}}{1-\lambda_j} \tag{8}$$

By substitution of the sum in the last line of equation 7 we obtain:

$$F(N) = \sum_{n=1}^{N} f(n) = \begin{cases} 0 & \text{if} \quad N < m \\ p^m & \text{if} \quad N = m \\ p^m + (n-m)(1-p)p^m & \text{if} \quad m < N \leq 2m \\ p^m + m(1-p)p^m + (1-p)p^m \sum_{j=0}^{m} c_j \frac{1-\lambda_j^{N-2m}}{1-\lambda_j} & \text{if} \quad N < m \end{cases}$$

We display the CDF of D, F in figure 3.

5 Average number of coin flip to get m heads on a row

We compute the expected value of D, noted $E(D)$.

$$E(D) = \sum_{i=m}^{+\infty} iP(D=i) = mp^m + (1-p)p^m \sum_{i=m+1}^{2m} i + (1-p)p^m \sum_{i=2m+1}^{+\infty} iu_{i-2m-1} \tag{9}$$

We are going to study the partial sums of the last equation. Let $A \in \mathbb{N}$ such that $A > 2m+1$.

$$\sum_{i=2m+1}^{A} iu_{i-2m-1} = \sum_{i=0}^{A-2m-1} (i+2m+1)u_i = \sum_{i=0}^{A-2m-1} iu_i + (2m+1)\sum_{i=0}^{A-2m-1} u_i$$

We are going to study separately the two previous sums.
On one hand: According to equation 8:

$$(2m+1)\sum_{i=0}^{A-2m-1} u_i = (2m+1)\sum_{j=0}^{m} c_j \frac{1-\lambda_j^{A-2m}}{1-\lambda_j}$$

Hence

$$\lim_{A\to+\infty} (2m+1)\sum_{i=0}^{A-2m-1} u_i = (2m+1)\sum_{j=0}^{m} \frac{c_j}{1-\lambda_j} \tag{10}$$

On the other hand:

$$\begin{aligned}
&\sum_{i=0}^{A-2m-1} iu_i \\
&= \sum_{i=0}^{A-2m-1} i\sum_{j=0}^{m} c_j\lambda_j^i \\
&= \sum_{j=0}^{m} c_j \sum_{i=0}^{A-2m-1} i\lambda_j^i \\
&= \sum_{j=0}^{m} c_j \frac{1}{1-\lambda_j}\left(\frac{\lambda_j-\lambda_j^{A-2m}}{1-\lambda_j} - (A-2m-1)\lambda_j^{A-2m}\right)
\end{aligned}$$

Hence

$$\lim_{A\to+\infty} \sum_{i=0}^{A-2m-1} iu_i = \sum_{j=0}^{m} \frac{c_j\lambda_j}{(1-\lambda_j)^2} \tag{11}$$

By substitution of the results of equations 10 and 11 in equation 12, we obtain:

$$E(D) = mp^m + (1-p)p^m\frac{3m^2+m}{2} + (1-p)p^m\left(\sum_{j=0}^{m}\frac{c_j\lambda_j}{(1-\lambda_j)^2} + (2m+1)\sum_{j=0}^{m}\frac{c_j}{1-\lambda_j}\right) \tag{12}$$

Using this equation for the expected value, we get insight on the variations of the average of coin flips needed. We do so by plotting once in function of m: the streak size and once in function of p: the probability to get head. We present those two plots in figure 4. We observe that increasing m results in an exponential growth in the average number of coin flips.

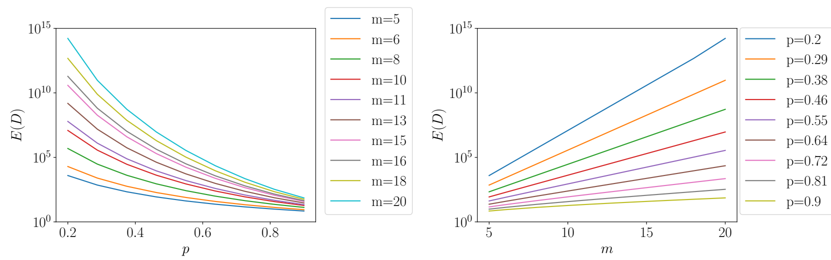

Figure 4: Average number of coin flips to get m head on a row where the probability to get head is p. On the left, average in function of p, on the right, average in function of m. Both figure are in log-scale for the y-axis.

6 Related work

Using Markov chains we find that the expected number of coin flips until m heads on a row is $\frac{p^{-1}-1}{1-p}$ [5]. This formula is easier to work with than the expression we found for $E(D)$ in equation 12; but [5] does not provide an expression for the mass function nor the CDF. Those two functions give more information on the law of D than just the average and can be used to find quantile, moments or event generate sample using inverse transform sampling [2].

7 Conclusion

We have successfully computed an expression of the mass function and the CDF of the number of coin flips required before getting m heads on a row. Those expressions relies on root finding and matrix inversion algorithms with a complexity increasing with m. We provide python code at the following repository git.jaalmoes.com/coin_flip that allows to compute both the mass function and the CDF for arbitrary values of p and m by using state of the art base algorithms of the Numpy [6] python package.

This theory along with our implementation allows to compute confidence intervals, statistical testing and random sampling. Those three usage are foundations of reliable engineering and data analysis.

References

[1] S. Christian Albright. A statistical analysis of hitting streaks in baseball. *Journal of the American Statistical Association*, 88(424):1175–1183, 1993.

[2] Robet Collins. Intro to Sampling Methods. https://www.cse.psu.edu/ rtc12/CSE586/lectures/cse586samplingPreMCMC.pdf, 2005. [Online; accessed 2024-01-23].

[3] Jérôme de Reffye. Numerical simulation of system dynamical reliability. In *2017 2nd International Conference on System Reliability and Safety (ICSRS)*, pages 20–30, 2017.

[4] Margaux Duroeulx. *Évaluation de la fiabilité des systèmes modélisés par arbres de défaillances grâce aux techniques de satisfiabilité*. Theses, Université de Lorraine, March 2020.

[5] Paul Ginsparg. How many coin flips on average does it take to get n consecutive heads? https://www.cs.cornell.edu/ ginsparg/physics/INFO295/mh.pdf, 2005. [Online; accessed 2024-01-17].

[6] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.

[7] Roger A. Horn and Charles R. Johnson. Matrix analysis, 2nd ed. 2012.

[8] Yoshiaki Nishikawa, Kousuke Nogami, and Eiji Takahashi. A method for reducing tcp latency due to consecutive packet losses in mobile networks. In *2016 18th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pages 1–4, 2016.