MM 225: AI and Data Science

Discrete probability distributions

M P Gururajan and Hina A Gokhale

August 3, 2023

Biometry

A NOTE ON THE STATISTICAL AND BIOMETRIC WRITINGS OF KARL PEARSON.

BY P. C. MAHALANOBIS.

STATISTICAL INDIAN INSTITUTE

UNITY IN DIVERSITY

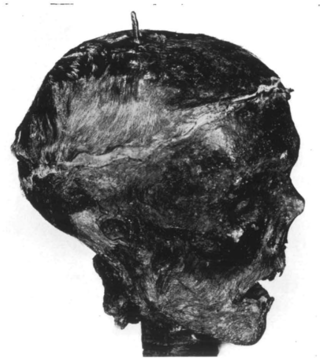
A Note on the Statistical and Biometric Writings of Karl Pearson
Author(s): P. C. Mahalanobis
Source: *Sankhyā: The Indian Journal of Statistics (1933-1960)*, Vol. 2, No. 4 (1936), pp. 411-422
Published by: Indian Statistical Institute
Stable URL: https://www.jstor.org/stable/40383786
Accessed: 03-08-2023 03:44 +00:00

Oliver Cromwell's head

THE WILKINSON HEAD OF OLIVER CROMWELL AND ITS RELATIONSHIP TO BUSTS, MASKS AND PAINTED PORTRAITS.

BY KARL PEARSON, F.R.S. AND G. M. MORANT, D.Sc.

Oliver Cromwell's head



The Wilkinson Head in Right Profile, showing the oak pole and the corroded tip of the iron prong, and the cincture marking the removal of the skull-cap to take out the brain. Note flowing moustache and hair on chin.

Oliver Cromwell's head

The Walker Portrait of Cromwell in the National Portrait Gallery, No. 536.

Outline

1. Plan for next 6 weeks!
2. Discrete probability distributions
3. A speculation interlude
4. Coin toss using python
   - Unpack coin toss
     - Coin toss: win and lead distributions

The plan!

Next 6 weeks

- Python programming
- Basics of probability and statistics
- Some linear algebra
- Some optimization: game theory problems as examples
- Data visualization
- **What is the idea?**
  AI and ML: Using python and concepts from probability, statistics, linear algebra, and optimization to make sense of large scale data

Textbook

Introduction to probability: Second revised edition, Charles M Grinstead, J. Laurie Snell, American Mathematical Society, 1997.
My copy: Reprint Indian edition 2012.
Free copy of the book: http://www.dartmouth.edu/~chance

Lecture 1: Discrete probability distributions

A game and some questions!

Suppose we toss a coin. Assume the coin is fair. If head (H) comes up, G(uru) gets Re. 1 and H(ina) loses Re. 1; if tail (T) comes up, H gets Re. 1 and G loses Re. 1. Suppose the coin is tossed 40 times.

Questions

- Which amount do you think has the maximum probability of winning for G?
- What fraction of time do you expect G to be in the lead?

Speculation interlude: let us play mentimeter

Speculation 1

Maximum probability of winning is for zero rupees. As we move away from zero, such as -2, +3 etc, the probability drops.

- **Link:** https://www.menti.com/aluu9bhhm7bc
- **Code:** Go to menti.com and use code 5778 0645

Results

Result for Speculation 1

Speculation 2

We expect G to be on the lead 50% of the times.

- **Link:** https://www.menti.com/altjnx5pjcjj
- **Code:** Go to menti.com and use code 2260 1193

Results

Result for Speculation 2

Simulations!

How to check our intuition? Make the computer play. In addition, we can try and get some more specific answers to questions such as

- What is the probability that G will win Rs. X in 40 tosses?
- How many times in the 40 tosses will G be in the lead?

Coin toss using python

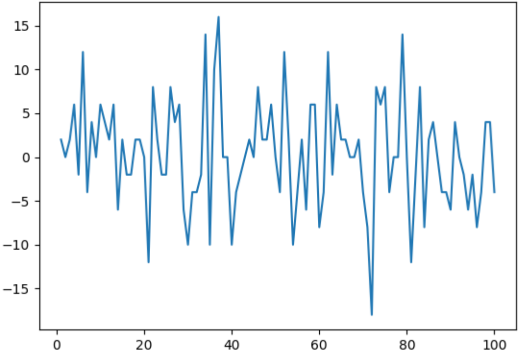
CoinToss.py

```
import matplotlib.pyplot as plt
import numpy as np
import random
M = 100
N = 40
Coin = ['H','T']
y = np.linspace(1,M,M)
E=[]
```

CoinToss.py

```
for j in range(M):
    heads = 0
    tails = 0
    for i in range(N):
        x = random.choice(Coin)
        if(x == 'H'):
            heads = heads  + 1
        else:
            tails = tails - 1
    z = (heads+tails)
    E.append(z)
plt.plot(y,E)
plt.show()
```

Result



Understanding the probability and statistics of coin toss

Random variable

- **Experiment**
  Toss a coin, roll a die, inspect a component, analyse a blood sample, ...
- **Random variable**
  Outcome of an experiment – Head / Tail, 1/2/3/4/5/6, Accept/Reject, Dengue/No dengue
- **Note**
  Random variable because experimental outcome depends on chance

Probability

- Fair coin: we assign equal probability to the outcomes of H and T. $m(H) = m(T)$.
- m: distribution function of the random variable, say X where X is the toss of a fair coin; a non-negative number
- Proabilities add up to unity. $m(H) + m(T) = 1$.
- Since $m(H) = m(T)$, $m(H) + m(T) = 1$, we get $m(H) = m(T) = 0.5$
- $\mathrm{P}(\mathrm{X} = \mathrm{H}) = 0.5$; $\mathrm{P}(\mathrm{X} = \mathrm{T}) = 0.5$
- Frequency concept: If you toss a fair coin a large number of times, 50% of the times you will get H and 50% of the times you will get T.

Expectation

- We assigned a number ($+1$ and -1) to the outcomes H and T
- $E(X) = \sum xm(x) = 0$
- E is known as expectation (or mean μ)
- Our plot: mean is indeed zero
- You can use np.mean commmand to get the average of the plot
- There is a spread around the mean; we will discuss about this spread later

Importance of expectation

Suppose in TechFest, G keeps a stall. The visitors can toss a coin ten times. They get Rs. 2 if H or lose Re. 1 if T. How much should be the entry fee be for playing the game so that G can break even at the end of the day – assuming a large number of the participants do play the game?

Answer (and another question)!

The expectation is $E = 2 * 0.5 + 1 * 0.5 = 1$. So, the visitors should pay Rs. 10 to play the game once.
Check this result by making the computer to play the game – by modifying the script.
If G keeps the entry fee at Rs. 12 (G can be sneaky like that!), and 1000 participants play the game, how much money did he make?

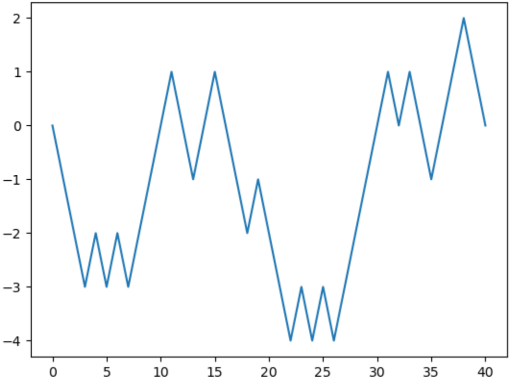
CoinTossWin.py

```
import matplotlib.pyplot as plt
import numpy as np
import random
N = 40
Coin = ['H','T']
y = np.linspace(0,N,N+1)
heads = 0
tails = 0
P = 0
Win=[0]
```

CoinTossWin.py

```
for i in range(N):
    x = random.choice(Coin)
    if(x == 'H'):
        heads = heads  + 1
        P = P + 1
    else:
        tails = tails - 1
        P = P - 1
    Win.append(P)
plt.plot(y,Win)
plt.show()
```

Result



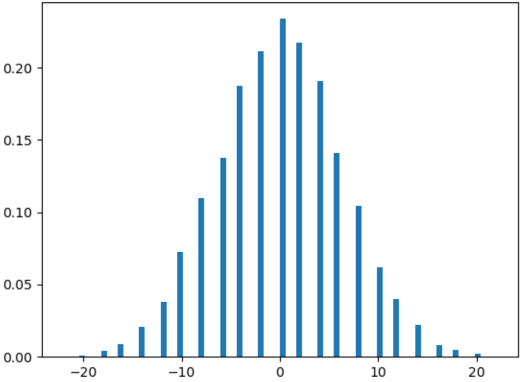
CoinTossWinDistrib.py

```
import matplotlib.pyplot as plt
import numpy as np
import random
M = 10000
N = 40
Coin = ['H','T']
y = np.linspace(0,M,M+1)
Win=[0]
for j in range(M):
    heads = 0
    tails = 0
    P = 0
```

CoinTossWinDistrib.py

```
for i in range(N):
        x = np.random.choice(Coin)
        if(x == 'H'):
            heads = heads  + 1
            P = P + 1
        else:
            tails = tails - 1
            P = P - 1
    Win.append(P)
plt.hist(Win,bins=80,density=True)
plt.show()
```

Result



Comments

- Win: highest probability is indeed for 0
- Does the plot remind you of anything?
- Draw an outer envelope of the spikes!!
- Why? Will discuss in one of the sessions.

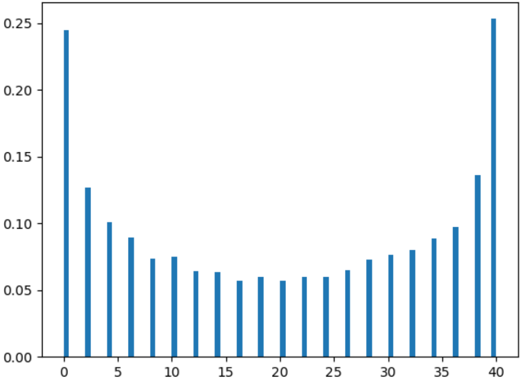
CoinTossWinLeads.py

```
import matplotlib.pyplot as plt
import numpy as np
import random
M = 10000
N = 40
Coin = ['H', 'T']
y = np.linspace(0, M, M+1)
Lead = [0]
for j in range(M):
    heads = 0
    tails = 0
    P = 0
    L = 0
```

CoinTossWinLeads.py

```
    for i in range(N):
        x = np.random.choice(Coin)
        if(x == 'H'):
            if(P == -1):
                L = L - 1
            heads = heads + 1
            P = P + 1
        else:
            tails = tails - 1
            P = P - 1
        if(P >= 0):
            L = L + 1
    Lead.append(L)
plt.hist(Lead, bins = 80, density=True)
plt.show()
```

Result



Comments

- Lead: highest probabilities are not for 0
- The extremes have higher probability
- Why? problem known as random walk
- Many problems with zero mean but finite variance
- Will discuss in detail slightly later

THANK YOU!!!