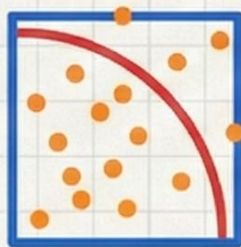




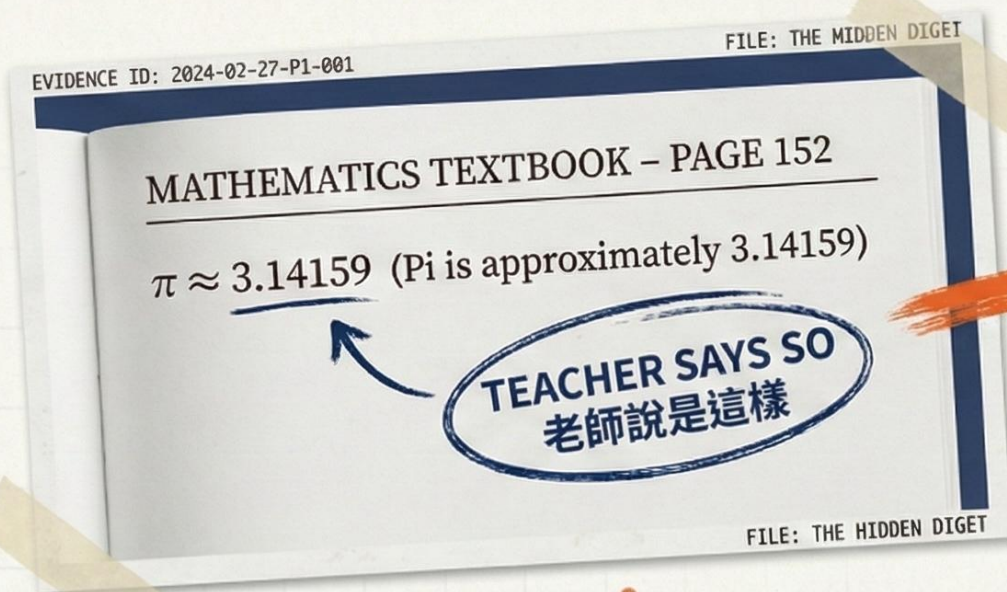
撒豆成兵算圓周： 蒙地卡羅法的大挑戰

國小獨立研究報告——從懷疑教科書到親
手驗證無理數的旅程



核心任務：利用機率統計（蒙地卡羅法）
與大數法則，親手計算出 π 值。

為什麼 π 是 3.14？ 教科書沒說的秘密



The Doubt (懷疑)



老師說 π 是「無限不循環的小數」。但既然算不完，我們現在用的 3.14 是怎麼來的？直覺上很難相信。

The Goal (目標)

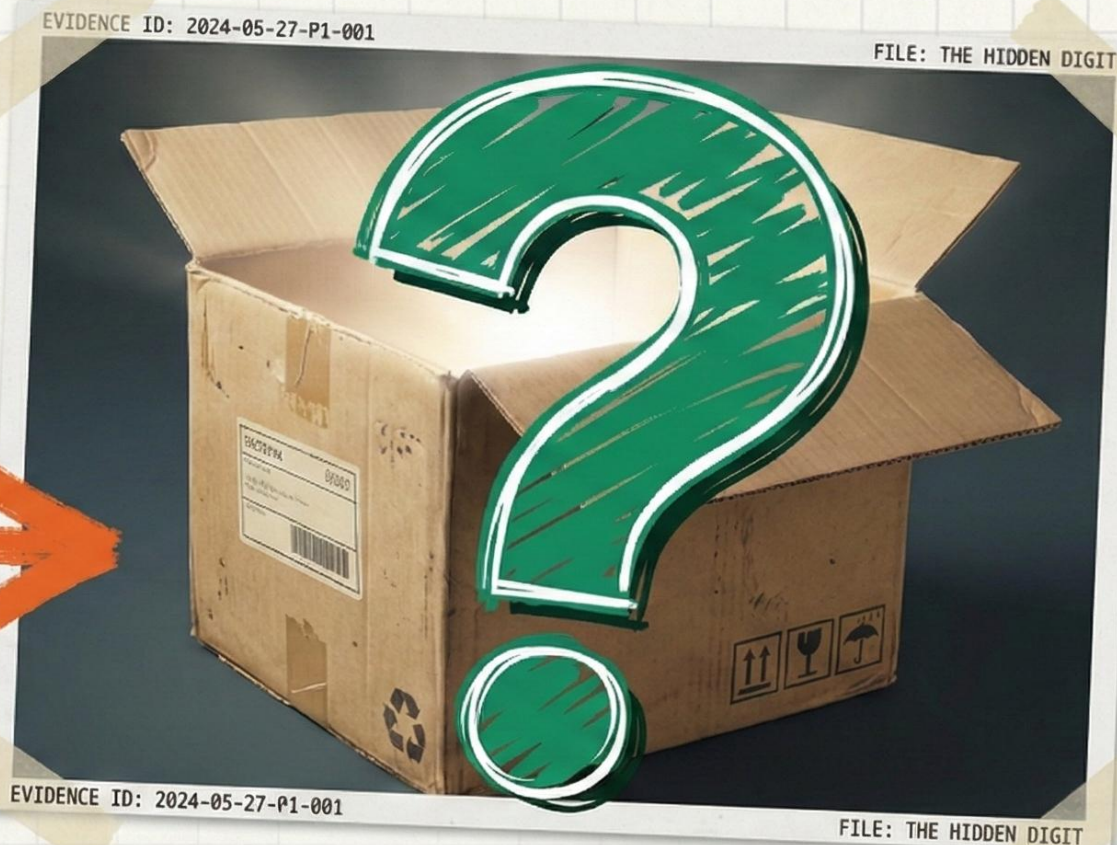


不只是背公式。我們決定拒絕生物觀察或物理實驗，我們要親手「找出這個隱藏的數字」。

The Obstacle (阻礙)



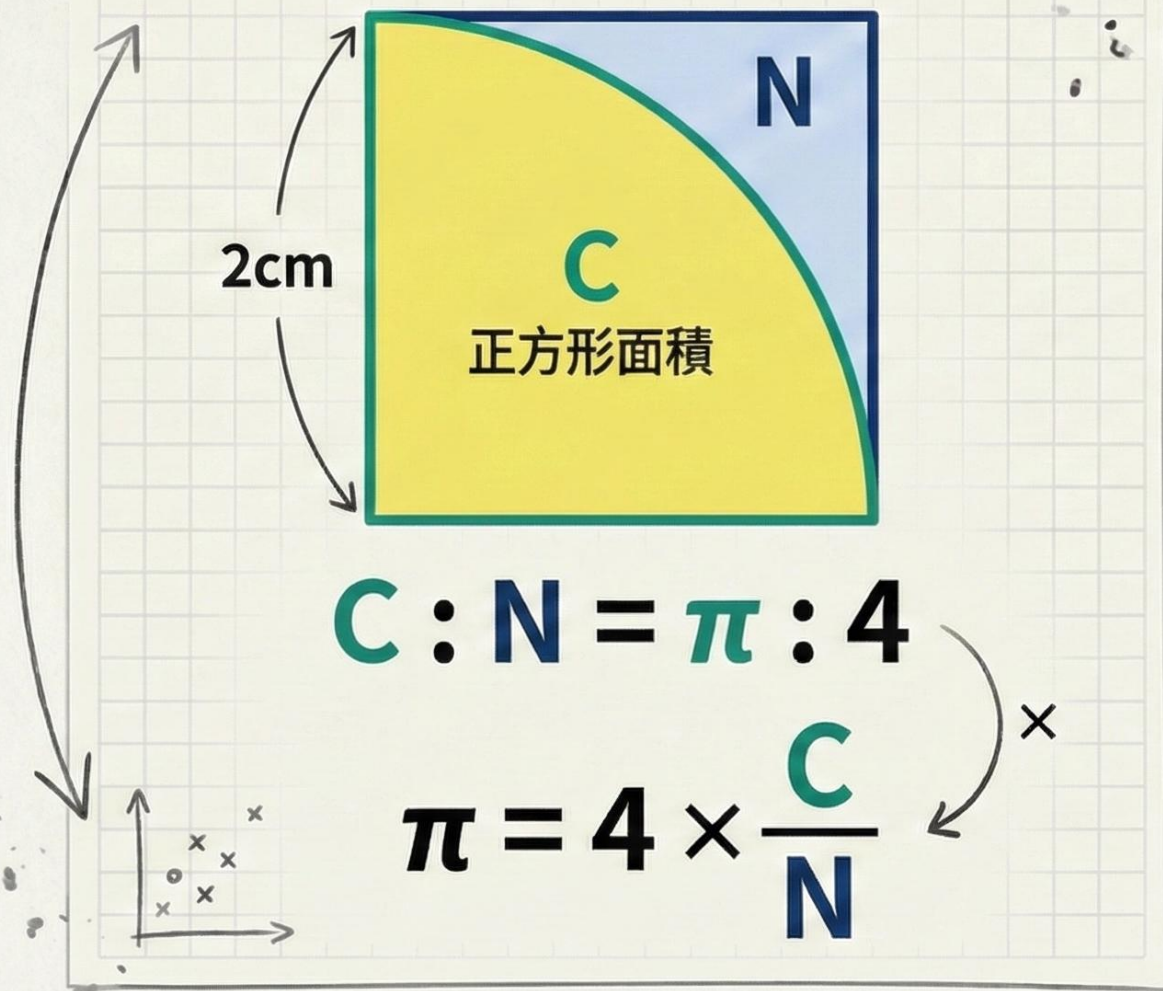
古代方法（如祖沖之割圓術）需要開根號與三角函數，這超國小數學範圍。我們需要一把不同的武器。



蒙地卡羅法：用「運氣」與「亂數」尋找真理

FILE: MONTE CARLO
EVIDENCE ID: 100122875823

Visual Explanation



Text Explanation

原理說明 (The Concept) :

蒙地卡羅法 (Monte Carlo Method) 是一種統計試驗法。它利用隨機抽樣來解決數學問題。

生活譬喻 (Life Metaphor) :

想像矇眼射飛鏢。只要投擲次數夠多，命中靶心的比例就能告訴我們靶的大小。



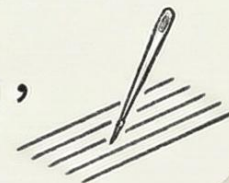
數學公式 (Mathematical Formula) :

正方形面積 (N) : $\frac{1}{4}$ 圓面積 (C) = 4 : π

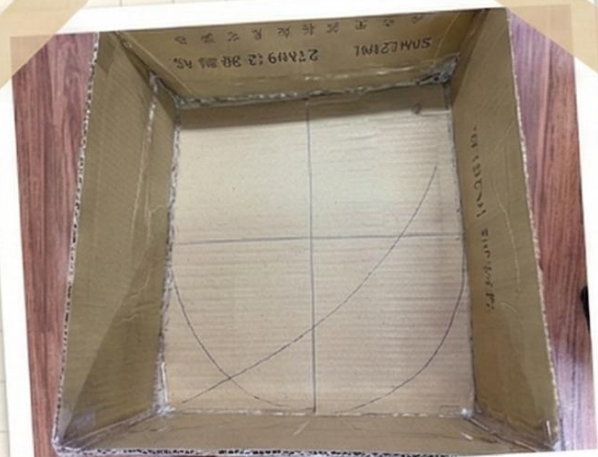
推導出 : $\pi = 4 \times \frac{C}{N}$

歷史淵源 (Historical Origin) :

源自蒲豐投針 (Buffon's Needle) 實驗，是用「運氣」破解數學難題的先驅。



實驗場景搭建：紙箱、橡皮擦與投擲規則



Field Equipment



1. 競技場 (The Arena)

- 器材：50x50cm 紙箱
- 標記：底部繪製半徑 50cm 的四分之一圓

2. 士兵 (The Soldiers)

- 投擲物：造型橡皮擦 (直徑約 1.5cm)
- 特性：不規則形狀增加彈跳隨機性

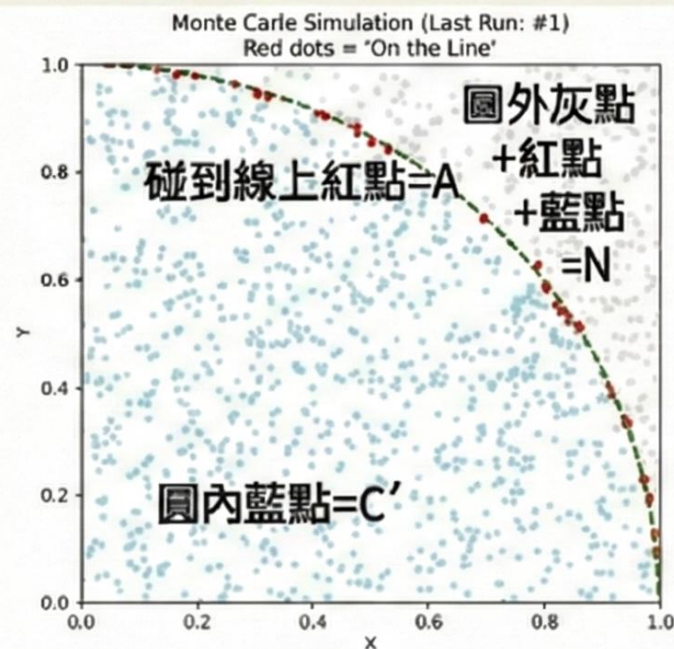
3. 投擲守則 (Protocol)

- 距離：距離箱子一個摺疊墊的寬度
- 姿勢：手臂彎曲置於胸前，確保力道統一
- 紀錄：計算落在「圓內」、「線上」、「圓外」的數量

邊界上的難題：壓在線上的橡皮擦，算圓內還是圓外？

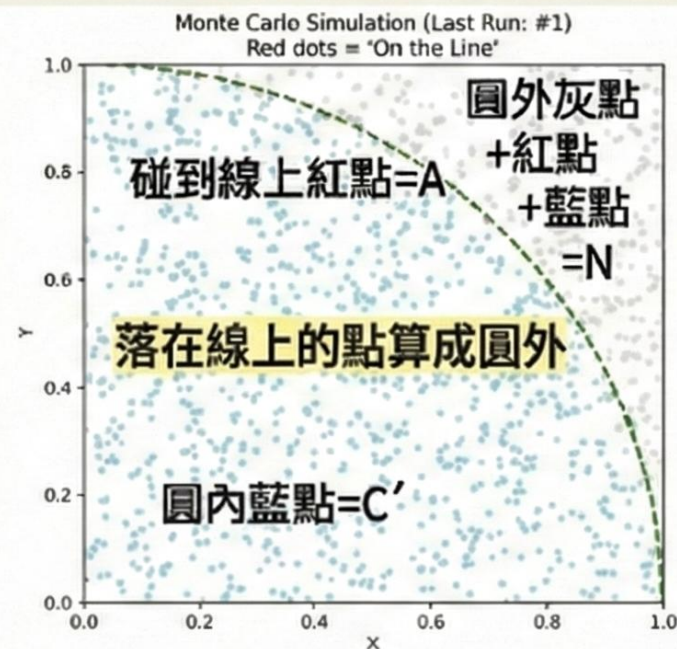
一個橡皮擦的體積，引發了核心的變數定義衝突。

寬鬆算法 Loose



線上 = 圓內 ($C = C' + A$)

嚴格算法 Strict



線上 = 圓外 ($C = C'$)

實驗假設：

物理世界的物體有體積。我們將分別計算這兩種定義，並與真實的 π (3.141592) 對決，看看哪種模型最符合現實。

實地投擲：累積數據的耐力賽



實驗過程 (The Grind):

投擲 → 等待彈跳靜止 → 判定位置 → 人工紀錄

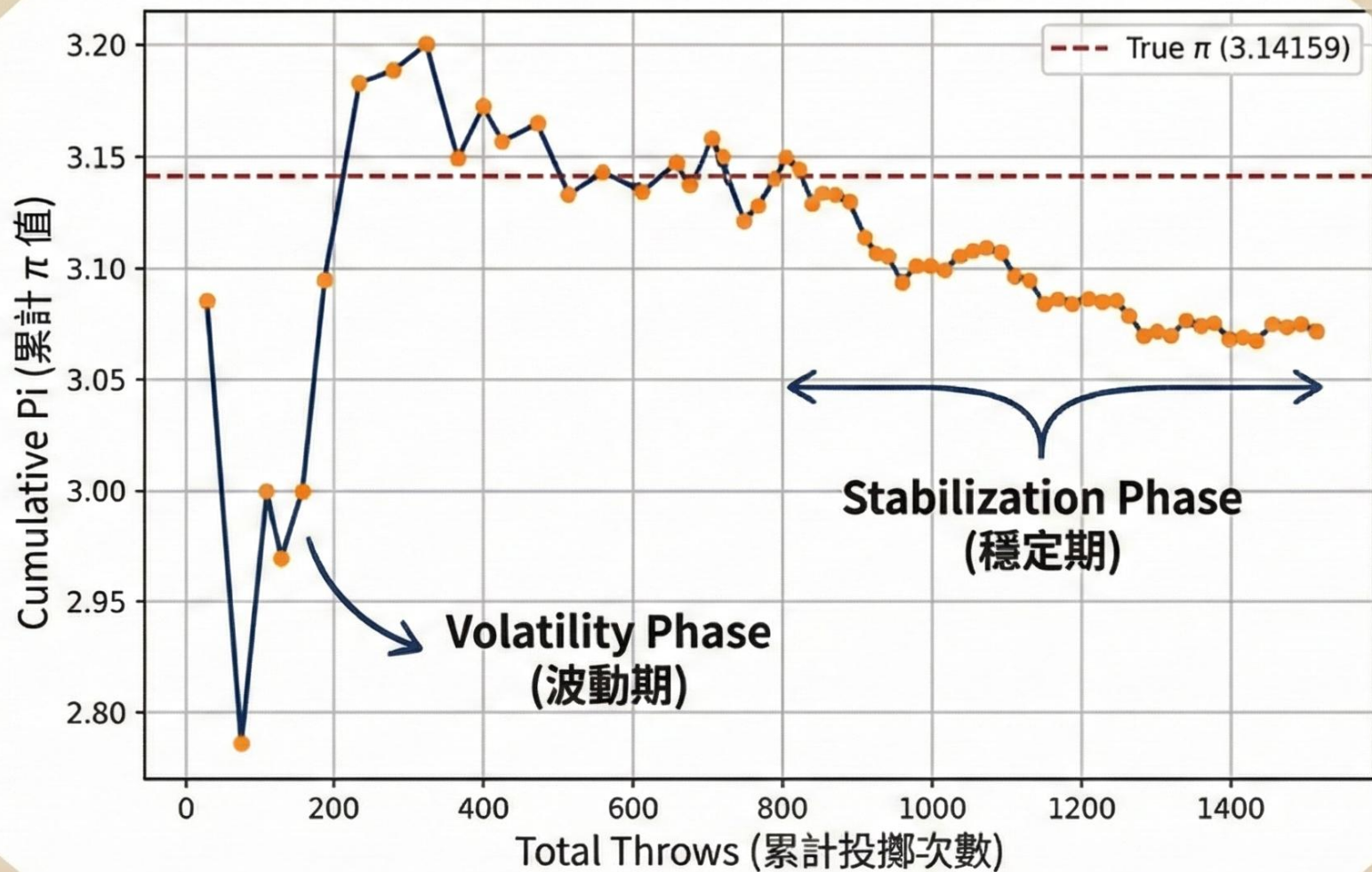
數據收集變數:

- N (總投擲數)
- C (圓內數量)
- A (線上數量)

心得:

這是一場與疲勞的對抗，但將枯燥的計數轉化為尋找真理的遊戲。

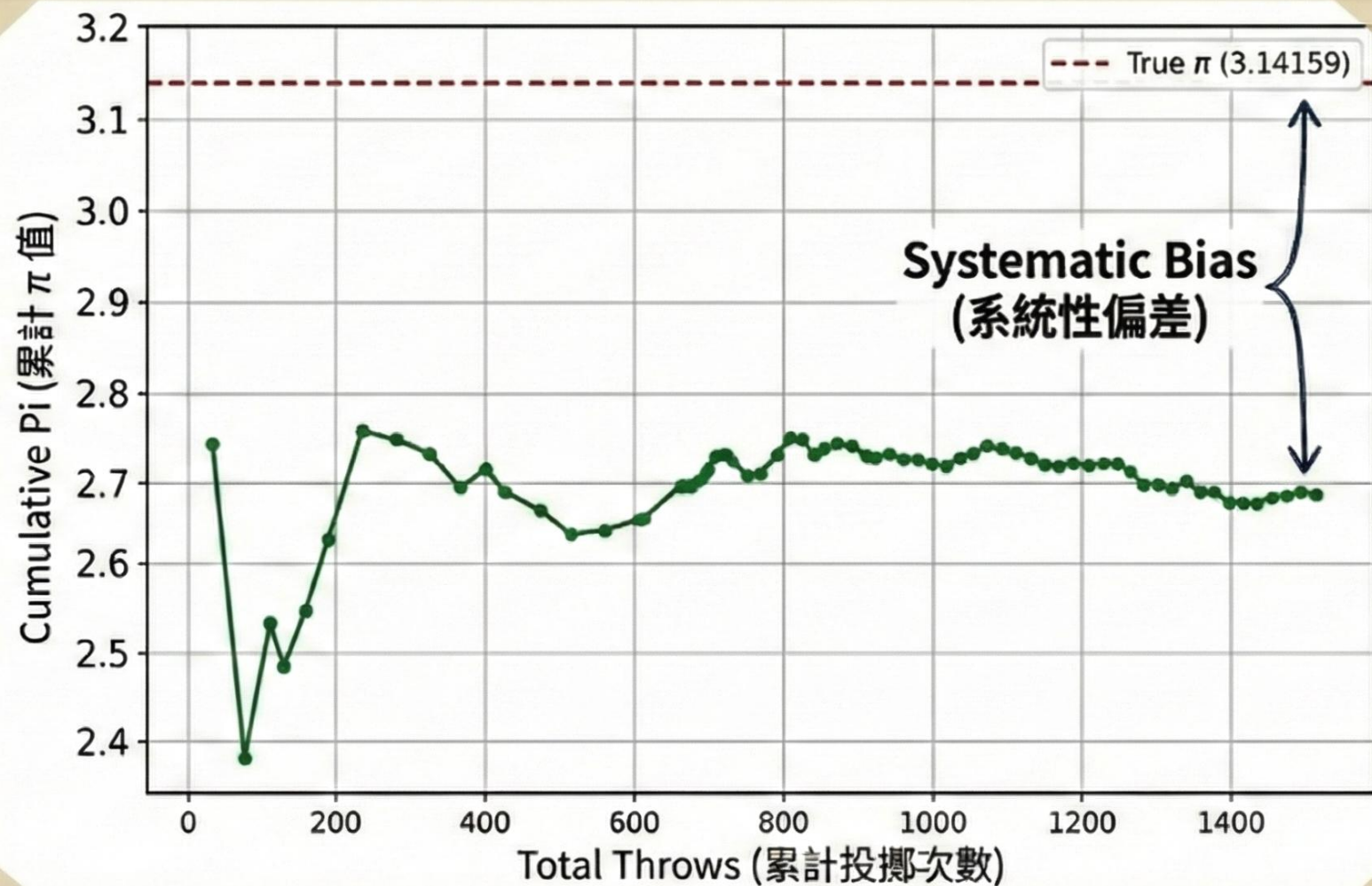
實驗結果 A：寬鬆設定 (線上視為圓內)



數據分析：

- 趨勢：初期波動劇烈，約 500 次投擲後趨於穩定。
- 收斂值：計算結果徘徊在 3.106 左右。
- 誤差率：約 1.39%。
- 發現：將「線上」納入計算，似乎補償了橡皮擦之的空隙，讓結果更接近 3.14。

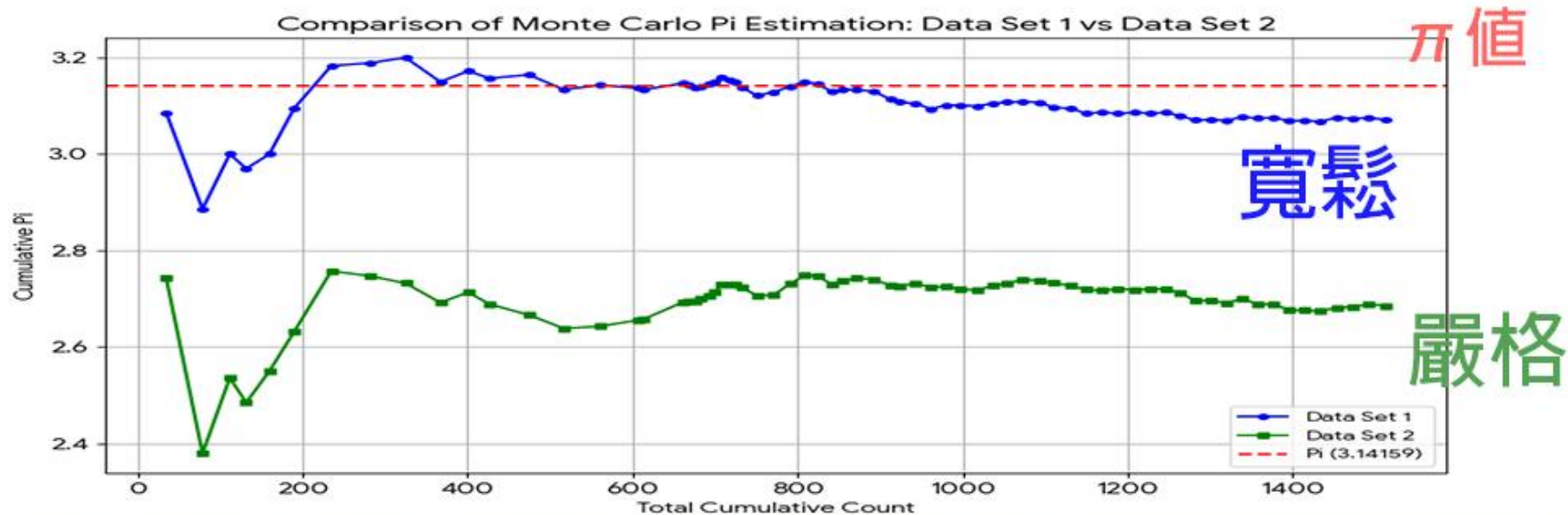
實驗結果 B：嚴格設定 (線上視為圓外)



數據分析：

- 趨勢：曲線在 700 次後變得非常平滑穩定 (高精確度)。
- 收斂值：數值落在 2.696 左右，遠低於 3.14。
- 誤差率：高達 14%。
- 發現：嚴格排除邊界加上物體本身的體積，導致面積被嚴重低估。雖然穩定，但是「準確地算錯了」。

數據對決：寬鬆 vs. 嚴格



關鍵洞察 (Key Takeaway)

1. 寬鬆算法 (Loose) → 較準確 (Accurate)：雖然波動大，但納入邊界能補償誤差，更接近真理。
2. 嚴格算法 (Strict) → 較穩定 (Stable)：數據一致性高，但存在系統性偏差，必須經過校正才有用。

結論：用橡皮擦算圓周率，必須對邊界「寬宏大量」。

召喚電腦：Python 模擬與大量數據的威力

為什麼需要模擬？

- 體力限制：手臂會痠，但電腦不會。
- 大數法則：我們需要更大的 N （總投擲數）來驗證收斂性。

模擬策略：

1. 模擬 A (擬真)：程式碼加入「線條粗細 (0.01)」與「物體大小」變數，模仿現實誤差。
2. 模擬 B (理想)：標準蒙地卡羅法 (無體積質點)，測試數學理論極限。

```
# 召喚電腦：Python 根據設定
import pandas as pd
import matplotlib.pyplot as plt
import random

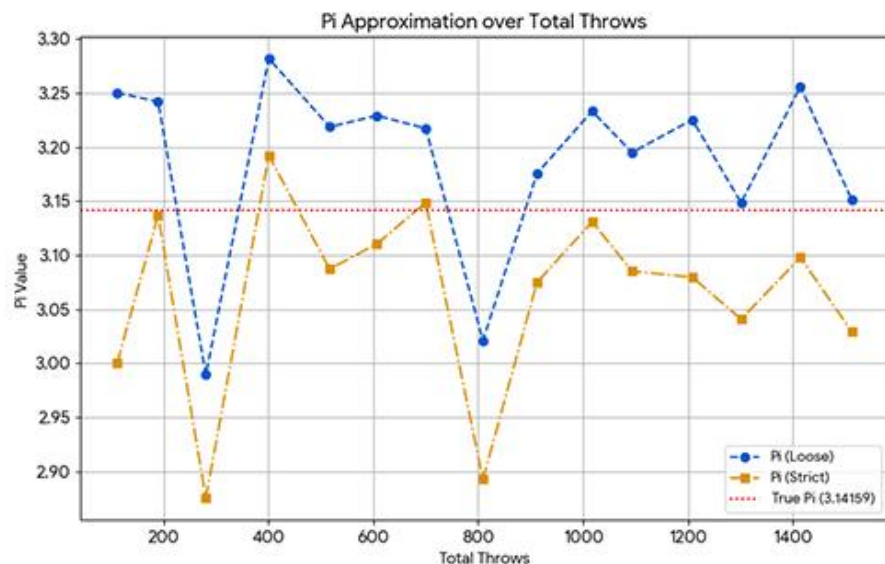
# 模擬參數 (Simulation Parameters)
LINE_THICKNESS = 0.01 # 線條粗細 (Simulation A)
OBJECT_RADIUS = 0.005 # 物體大小 (Simulation A)
TOTAL_THROWS = 1000000 # 大量數據 (Big N)

# 模擬 B: 理想蒙地卡羅法 (Ideal Monte Carlo)
def simulate_pi(n_throws):
    hits = 0
    for _ in range(n_throws):
        x = random.uniform(0, 1)
        y = random.uniform(0, 1)
        if x**2 + y**2 <= 1:
            hits += 1
    return (hits / n_throws) * 4

# 執行模擬
pi_estimate = simulate_pi(TOTAL_THROWS)
print(f"Estimated Pi with N={TOTAL_THROWS}: {pi_estimate}")
```



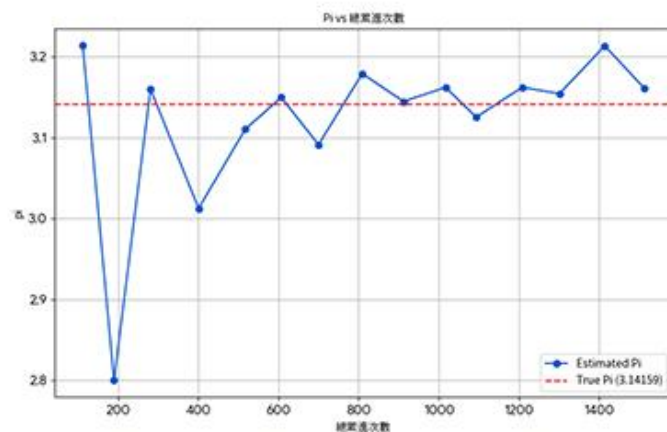
模擬結果：當樣本數達到極限



電腦模擬A

(物體有面積，
線有厚度)

電腦模擬B



數位驗證 (The Verdict):

- 電腦模擬 A*：即使在數位世界，只要考慮「線寬」，寬鬆算法 (Blue) 依然比嚴格算法 (Orange) 更接近 π 。
- 理想模型 B*：當點變成無體積時，更加接近真實 π 值 ($\pi \approx 3.14159$)。

數據摘要：

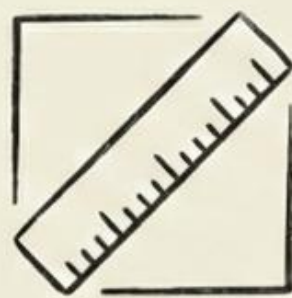
- 模擬寬鬆 $\pi \approx 3.18$ (誤差 2.66%)
 - 模擬嚴格 $\pi \approx 3.06$ (誤差 2.67%)
- 結論：實驗變數設定 (邊界定義) 對結果的影響，比樣本數大小更關鍵。

最終結論：大數法則與定義的藝術



大數法則 (Law of Large Numbers)

無論是丟橡皮擦還是跑程式，樣本數 (N) 越大，數據越穩定。初期的劇烈波動最終都會歸於平靜。



定義決定結果 (Definition Matters)

在現實世界中，物體都有體積。如何定義「邊界」(Loose vs Strict) 直接決定了計算的準確度。寬鬆的定義往往能修正物理世界的誤差。



任務成功 (Mission Accomplished)

我們成功利用紙箱與橡皮擦，驗證了蒙地卡羅法的有效性，並算出接近 3.14 的數值！

旅程的收穫：從「算數學」到「玩數學」



1. 克服恐懼

π 從一個課本上的冰冷符號，變成了我們親手「丟」出來的戰利品。

2. 掌握工具

從不懂 Excel 到學會 Python 模擬，我們學會用現代科技解決數學問題。

3. 團隊合作

「一人投擲，一人紀錄。」數學不再是孤獨的考試，而是充滿歡笑的探索遊戲。

結語：

「數學不只是課本上的數字，它是探索世界的遊戲。」