Monte Carlo Method/Simulations

It’s all about me…..

Prof. Mark Whitehorn
Emeritus Professor of Analytics
Computing
University of Dundee
Created first UK masters in Data Science

Consultant
Writer (author)
m.a.f.whitehorn@dundee.ac.uk

Monte Carlo simulations

Think of a Monte Carlo simulation as a way to mimic some aspect of real life in a computer model. These models make use of random numbers.

3

Named by.....

- Stanislaw Ulam
- 1946
- Physicist at Los Alamos Scientific Laboratory
- In hospital, playing Canfield Solitaire
- Wondered how often it came out
- Worked on combinatorial calculations for a while

then thought it might be faster/easier to simply play a large number of games and measure the answer

4

Monte Carlo Quotes

“The first thoughts and attempts I made to practice [the Monte Carlo Method] were suggested by a question which occurred to me in 1946 as I was convalescing from an illness and playing solitaires. The question was what are the chances that a Canfield solitaire laid out with 52 cards will come out successfully? After spending a lot of time trying to estimate them by pure combinatorial calculations, I wondered whether a more practical method than ‘abstract thinking’ might not be to lay it out say one hundred times and simply observe and count the number of successful plays.”

5

Monte Carlo Quotes

“This was already possible to envisage with the beginning of the new era of fast computers, and I immediately thought of problems of neutron diffusion and other questions of mathematical physics, and more generally how to change processes described by certain differential equations into an equivalent form interpretable as a succession of random operations. Later [in 1946], I described the idea to John von Neumann, and we began to plan actual calculations.”

6

JOURNAL OF THE AMERICAN STATISTICAL ASSOCIATION

Number 247 SEPTEMBER 1949 *Volume 44*

THE MONTE CARLO METHOD

NICHOLAS METROPOLIS AND S. ULAM
Los Alamos Laboratory

We shall present here the motivation and a general description of a method dealing with a class of problems in mathematical physics. The method is, essentially, a statistical approach to the study of differential equations, or more generally, of integro-differential equations that occur in various branches of the natural sciences.

ALREADY in the nineteenth century a sharp distinction began to appear between two different mathematical methods of treating physical phenomena. Problems involving only a few particles were

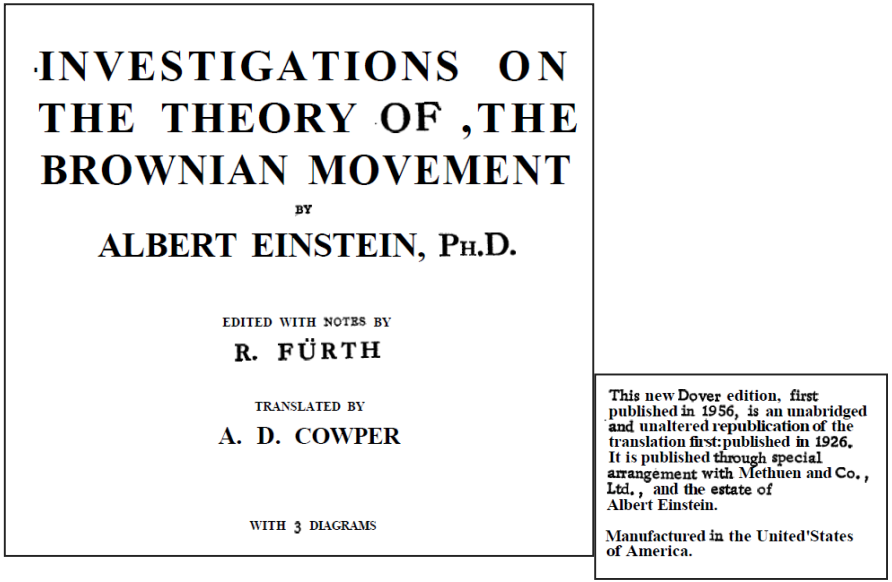
First Example - random walks

Some people, including Einstein, were/are fascinated by ‘random walks’

What is a random walk?

Why are they so fascinating?

Random walks are very easy to investigate using a Monte Carlo model.



INVESTIGATIONS ON THE THEORY OF THE BROWNIAN MOVEMENT

BY

ALBERT EINSTEIN, PH.D.

EDITED WITH NOTES BY

R. FÜRTH

TRANSLATED BY

A. D. COWPER

WITH 3 DIAGRAMS

This new Dover edition, first published in 1956, is an unabridged and unaltered republication of the translation first published in 1926. It is published through special arrangement with Methuen and Co., Ltd., and the estate of Albert Einstein.

Manufactured in the United States of America.

8

Monte Carlo method

As a general rule you need to:

- Define a domain of possible inputs
- Generate a set of inputs
  - These should be randomly chosen correctly from the domain
- Perform some computation on the inputs
- Aggregate the results
- Draw a conclusion
- Perform reality check

9

Monte Carlo – random walks

For example:
How far, on average, will a random walk take you after n steps?

- What is the domain?
- How will you draw values from the domain?
- What calculation will you perform?
- How will you aggregate?
- What is your conclusion?
- Does it sound reasonable?

10

Monte Carlo – random walks

For example:
How far, on average, will a random walk take you after n steps?

- What is the domain? (N,E,S,W)
- How will you draw values from the domain? (0.25 for each)
- What calculation will you perform? (measure distance from origin)
- How will you aggregate? (Perform multiple simulations)
- What is your conclusion? (Let's see)
- Does it sound reasonable?

11

Code in R

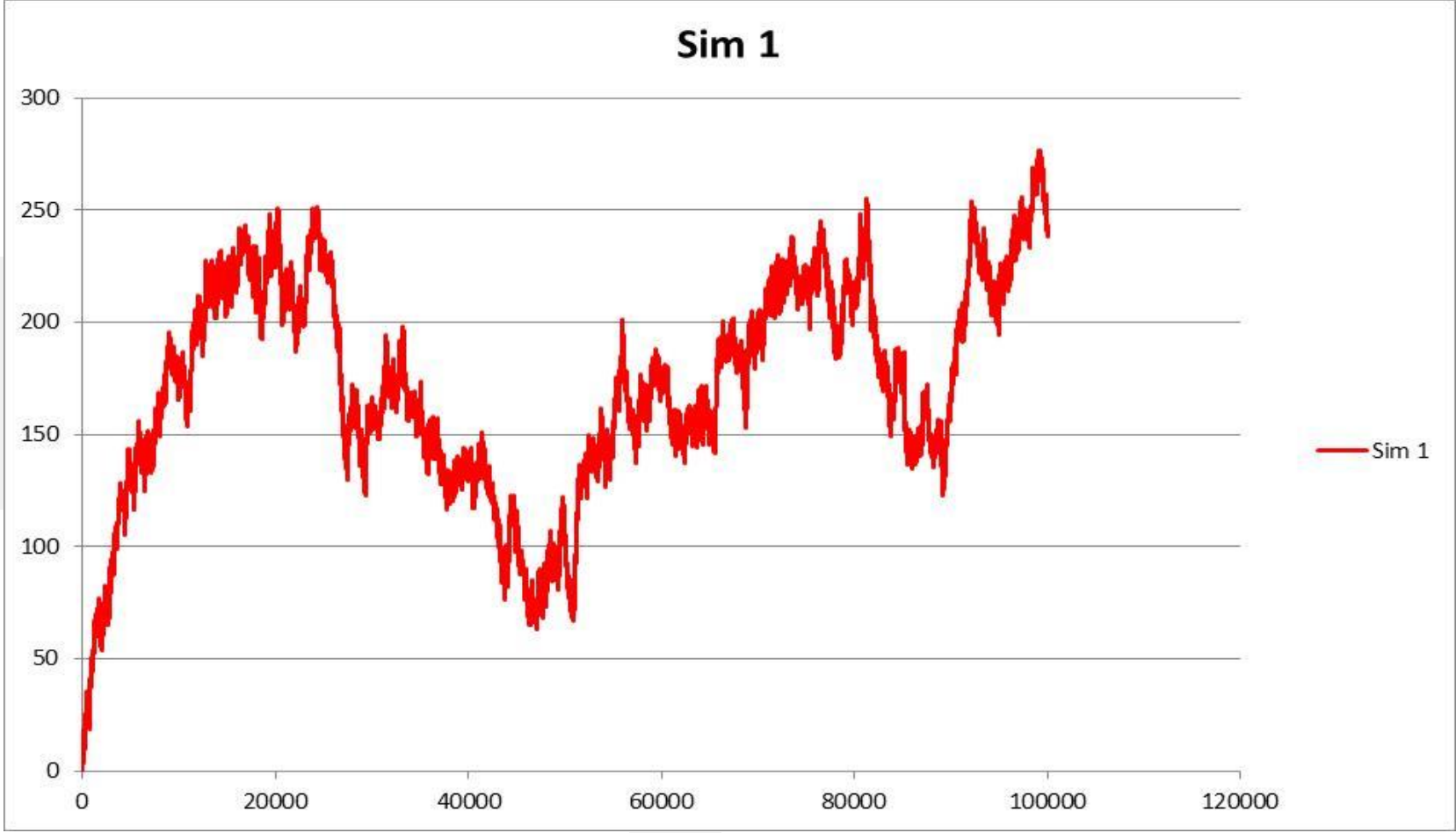
```
############################################################
# Monte Carlo simulation of random walk linear distance
############################################################
StartingSet=c('N','E','S','W')
Steps=100
Steps=Steps+1
Runs=50000 # up to 50,000 takes about 3 seconds
X = 1:Runs
LinDist= rep(0,Runs)
for(i in X) {
  Walk=sample(StartingSet,Steps, replace=TRUE)
  Walk[1] = "Null"
  XCord = sum((Walk=="N")-(Walk=="S"))
  YCord = sum((Walk=="E")-(Walk=="W"))
  LinDist[i] = sqrt(XCord^2 + YCord^2)
}
LinDistAvg = mean(LinDist)
LinDistAvg
```

Code in R

```
StartingSet=c('N','E','S','W')
Steps=10000 #10,000
Steps=Steps+1
X = 1:Steps
XX = 2:Steps #Steps=Steps+1
AllX = rep(0,Steps)
AllY = rep(0,Steps)
RunningX = rep(0,Steps)
RunningY = rep(0,Steps)

Walk=sample(StartingSet,Steps, replace=TRUE)
Walk[1] = "Null"
```

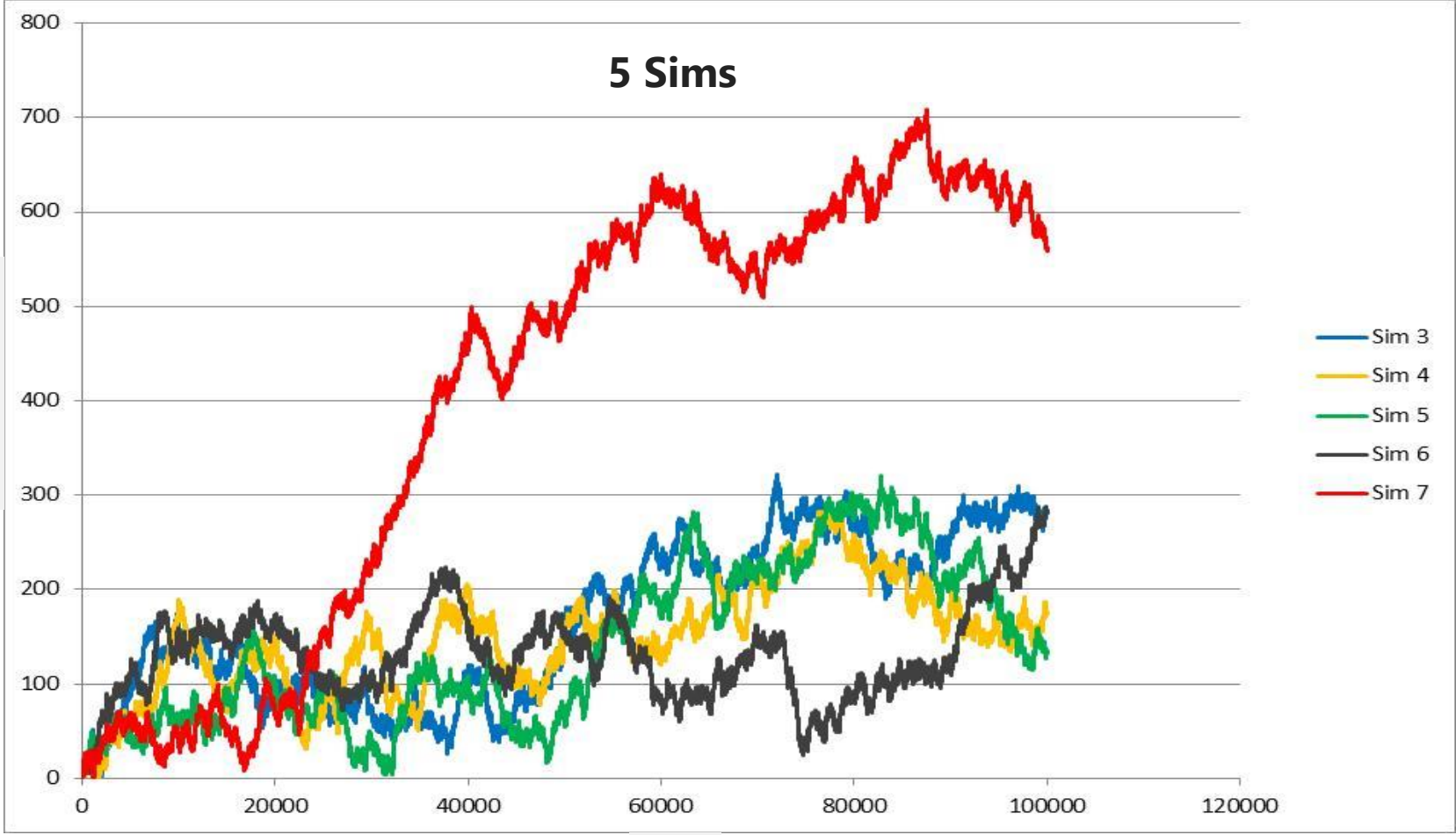
```
for(i in X) {
  if (Walk[i] == "N") AllY[i] = 1
  if (Walk[i] == "S") AllY[i] = -1
  if (Walk[i] == "E") AllX[i] = 1
  if (Walk[i] == "W") AllX[i] = -1
}
for (i in XX) {
  RunningX[i] = RunningX[i-1] + AllX[i]
  RunningY[i] = RunningY[i-1] + AllY[i]
}
plot(RunningX,RunningY,"l")

B = matrix(c(0, RunningX[Steps], 0,
RunningY[Steps]), nrow=2,   ncol=2)
lines(B, col="red")
points(0,0, col="red") #Mark End

XCord = sum((Walk=="N")-(Walk=="S"))
YCord = sum((Walk=="E")-(Walk=="W"))
LinDist = sqrt(XCord^2 + YCord^2)

LinDist
```

Distance From Origin

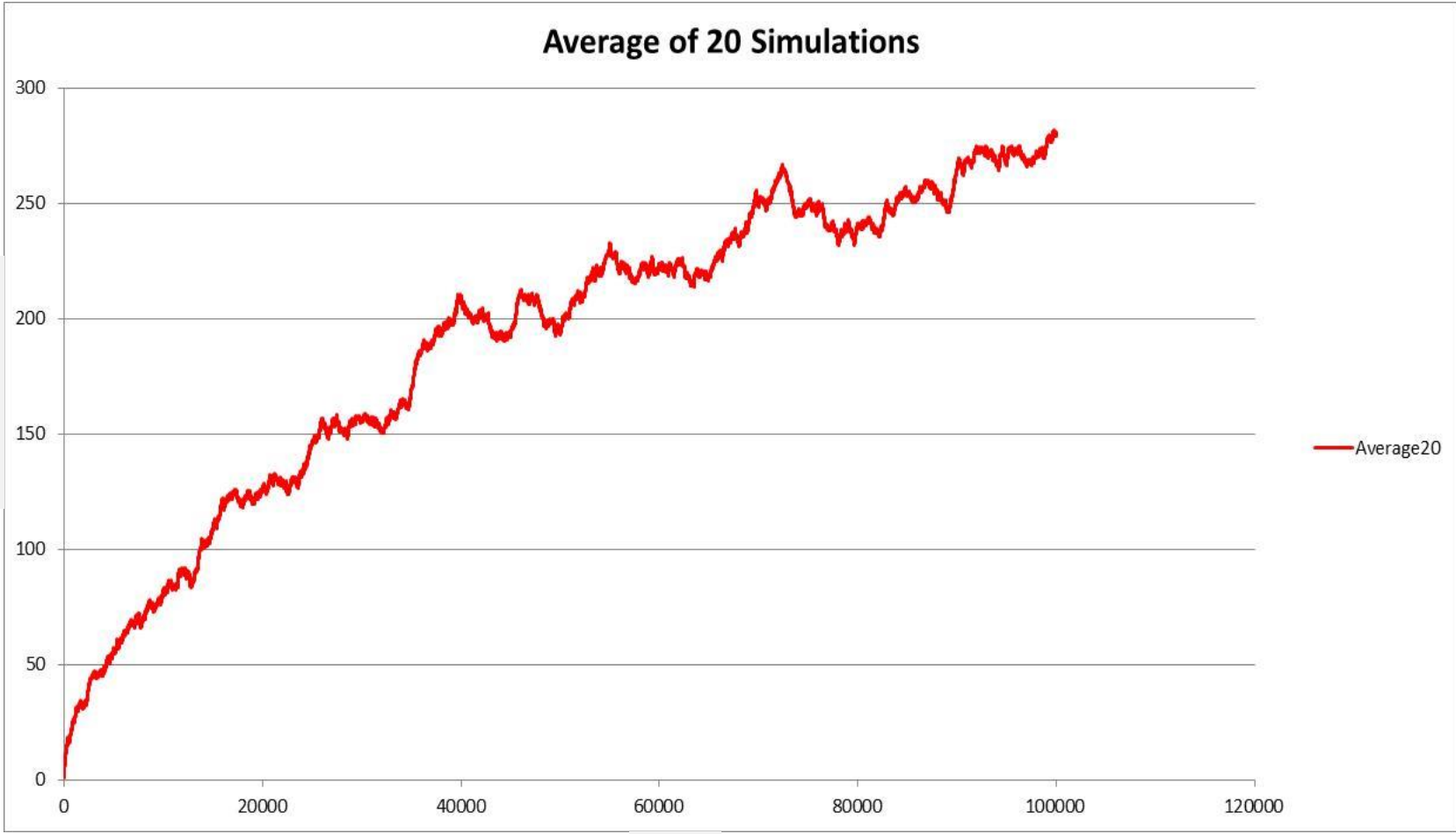


n

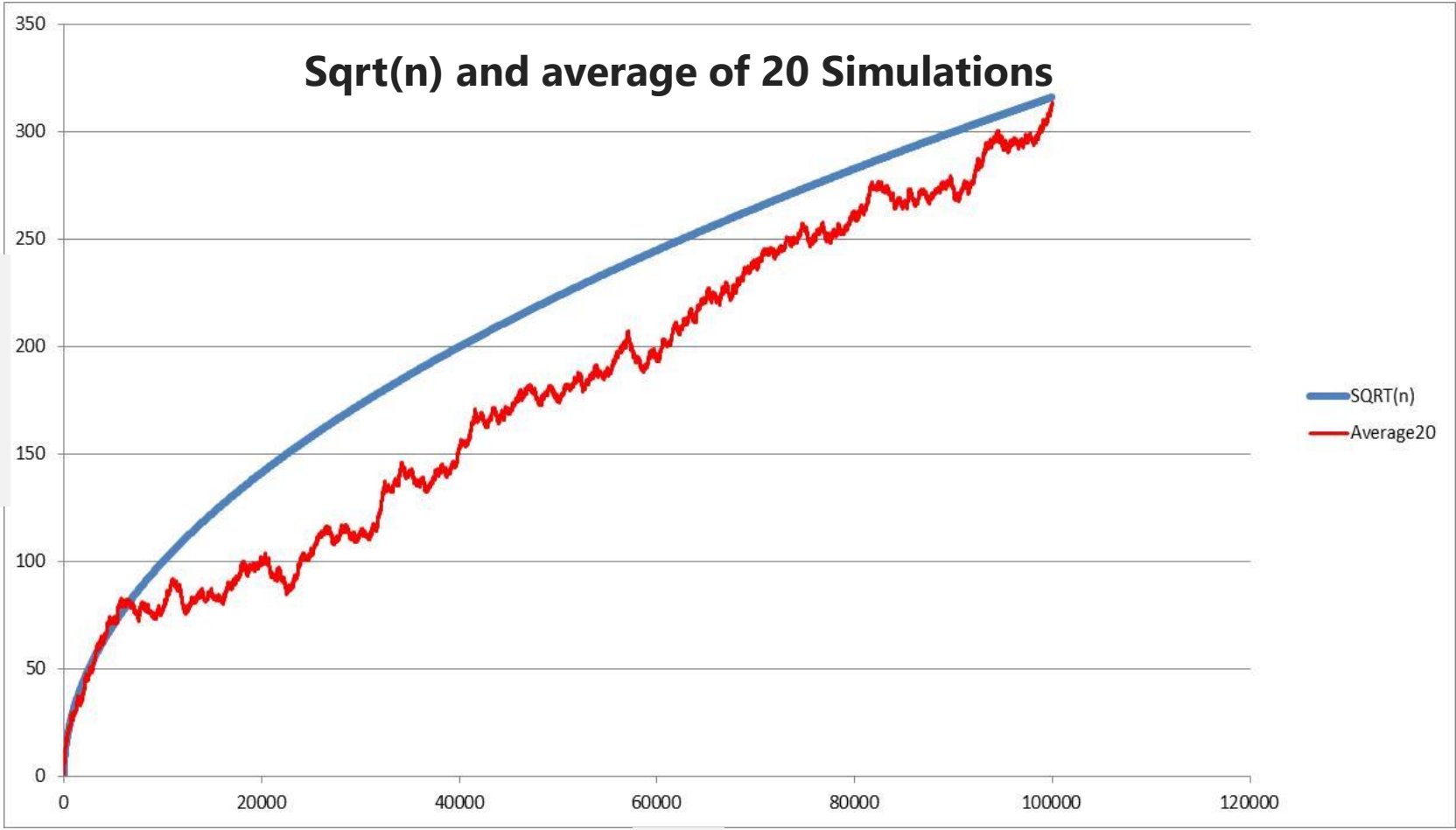
Distance From Origin



n

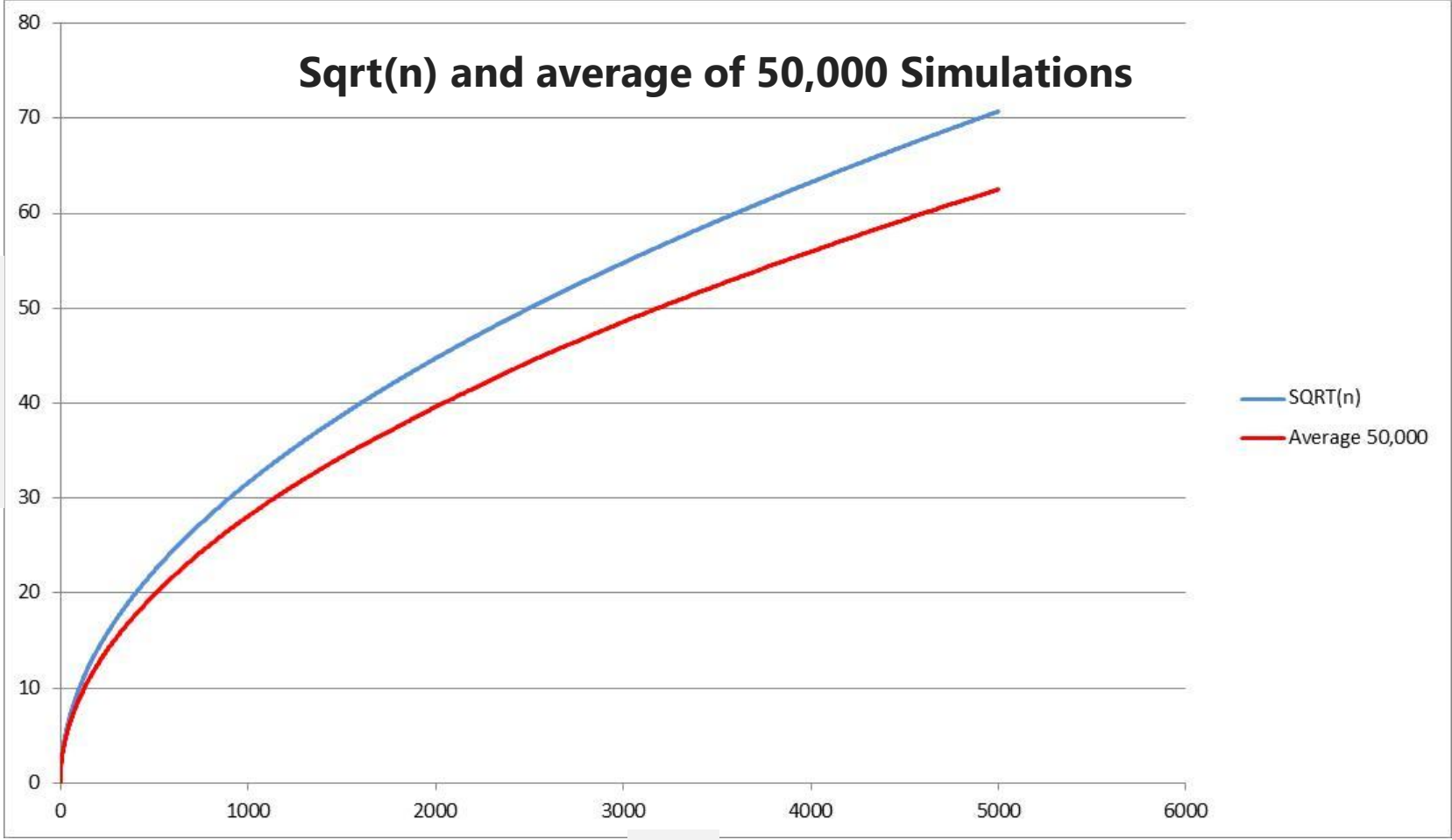


Distance From Origin



n

Distance From Origin



n

Monte Carlo – random walks

How far, on average, will a random walk take you after n steps?

- Does it sound reasonable?
- It looks convincing to me but:
  - It is NOT a proof
  - It is empirical (based on experience not pure logic)
  - It might be a better match to SQRT(n-1) or any other variant
  - But it may be close enough for our purposes

19

Coins in the pocket

MCSs are highly applicable.

For example, the “Coins in the Pocket” investigation (thanks to Paddy Blackwill for bring this problem to my attention!).

https://missingpersons.police.uk/en-gb/case/06-026129

A body was found in 1990. In a pocket was 53 pence (value) in coins (possibly five coins?) with some dated from 1967 & 1975.

We can use probabilities to deal with certain levels of complexity

Assuming no coins were planted on the body, can we make estimates as to when the body came to be lying there?

Yes, and the more information we have, the better we can make the estimate.

We can use probabilities to deal with certain levels of complexity

For example, if we assume that 10 million coins are minted each year (unrealistic*, but we are starting simply) since decimalisation (1971).

The body was discovered in 1990.

What are the chances in, say, 1980 of choosing 5 coins from the circulating pool of 100 million coins, none of which date later than 1975?

* https://www.royalmint.com/corporate/circulating-coin/uk-currency/mintages/

Monte Carlo methodology

Repeat this MCS for each year from 1975 to 1990:

- Define a domain of possible inputs (Pool of coins for a given year, say, 1980
- Generate a set of inputs
  - These should be randomly chosen correctly from the domain (say, draw five coins from the pool)
- Perform some computation on the inputs (are any coins later than 1975? Yes/No)
- Aggregate the results (Calculate % Yes from 100,000 draws)
- Draw a conclusion (Is that likely to be the year of death?)
- Perform reality check

23

Coin problem

An MCS is how Paddy approached this problem and he is kind enough to have made the code available.

https://colab.research.google.com/drive/11mLNLNWzUkVsPVjqmLS0UZHBBvJcZe8G

Paddy's code

```
sample_loose_change <- function(circulation, circulation_dist, trials=1E5) {
    # what are the chances that the highest dated coin is 1975
    # from 53p in loose change?

    success <- 0
    # we assume that "loose change" is 4-15 coins
    coin_counts <- sample(4:15, size = trials, replace = TRUE)
    for (coin_count in coin_counts) {
        # sample coins from those in circulation
        coin_dates <- sample(
            circulation,
            size = coin_count,
            replace = TRUE,
            prob = circulation_dist)
        if (max(coin_dates) == '1975') {
            success <- success + 1
        }
    }
    # estimated probability:
    return(success / trials)
}
```

So, those are a couple of use cases.

Reverse Engineering with MCS

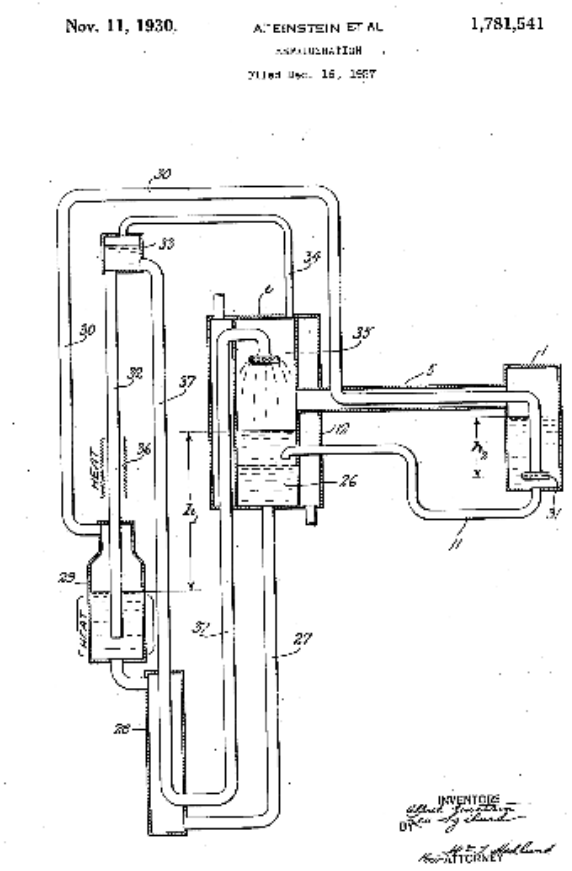
This is a really important facet of MCS that is often not appreciated (imho).

You don’t have to be Einstein to answer this one

And now, an apparent non-sequitur:

“Why is a gas-powered fridge like a random walk?”

REFRIGERATION Filed Dec. 16 1927 Patented Nov. 11, 1930 UNITED STATES PATENT OFFICE ALBERT EINSTEIN, OF BERLIN, AND LEO SZILARD, OF BERLIN-WILMERSDORF, GERMANY



Galton's quincunx as a MCS

Youtube is your friend for this one!

30

Expected Payoff

We work for a company that sells insurance by ‘phone. Our sales people talk to the customer and we run real time analytics based on the customer’s:

- answers to the questions
- word usage
- voice pattern

Expected Payoff

Our analytical system says:

- Product A Net value £5 – Probability of acceptance 0.14
- Product B Net value £90 – Probability of acceptance 0.54
- Product C Net value £200 – Probability of acceptance 0.32

Obviously we don't offer the customer product A, but should we offer B or C?

Expected Payoff

Probabilities tell us:

A £5 * 0.14 = £0.7

B £90 * 0.54 = £48.6

C £200 * 0.32 = £64.0

The best option is C

Gamblers call this the ‘expectation’ or ‘expected payoff’; the value of the prize times the probability of winning it. Blaise Pascal famously applied this logic to belief in a deity.

Deterministic

Deterministic systems/processes have a known (and repeatable) outcome from a given starting point.

Probabilities tell us:

A £5 * 0.14 = £0.7
B £90 * 0.54 = £48.6
C £200 * 0.32 = £64.0

The best option is always C

A = B + C
If B = 6 and C=4 then A=10
every single time

Stochastic (as opposed to deterministic)

Stochastic systems/processes display a level of indeterminacy; from the same starting point it is possible to reach a number (possibly infinite) of outcomes.

Monte Carlo Simulations are stochastic, in practice we provide that by introducing random numbers.