

## Lecture 2

Ideas

0	0	0	C		
0	0	0			
0	0	0			
0	0	0			
0	0	0			



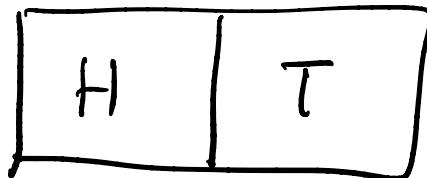
C					
	0	0			
		0			
			0		
				0	

Send python  
send poll

# Discrete probabilities

H, T

→

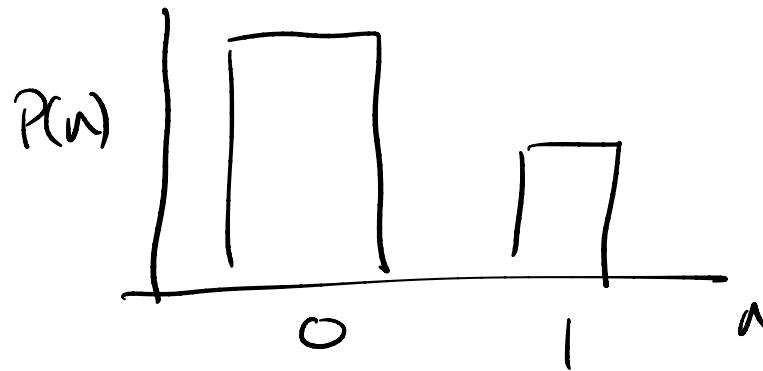
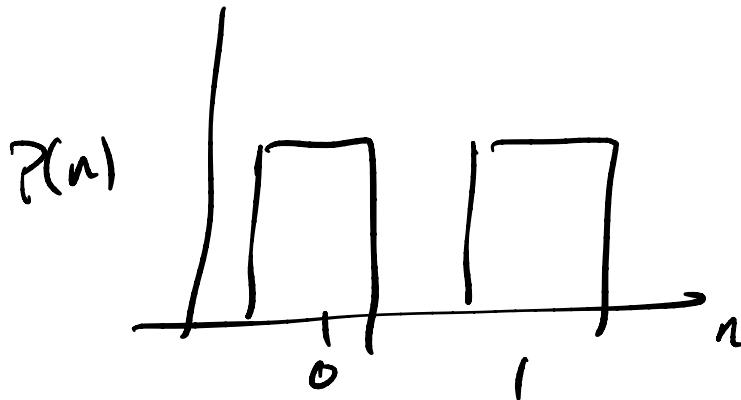


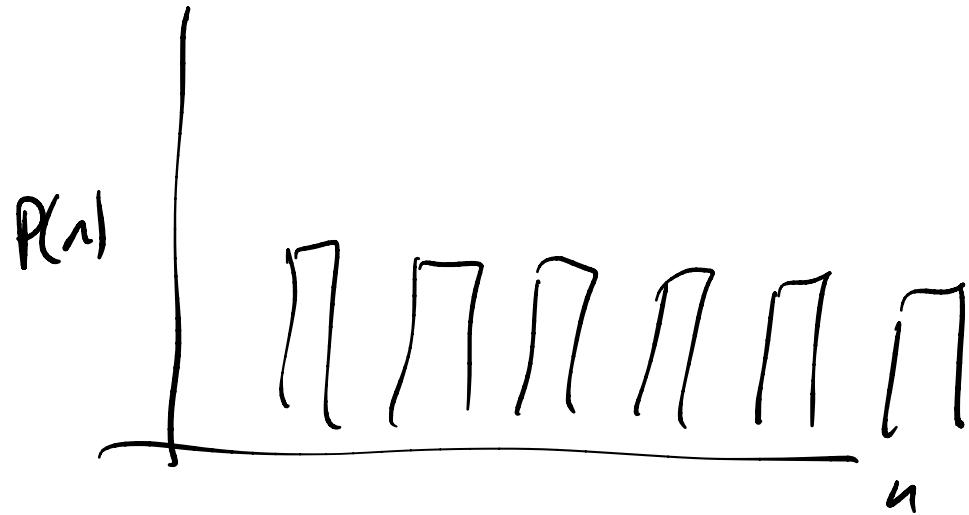
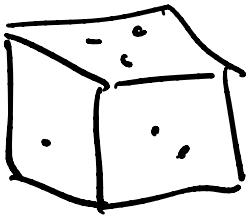
0

1

← n

$$P_H + P_T = 1$$





$$\sum_{n=1}^k P_n = 1 = P_1 + P_2 + P_3 + P_4 + P_5 + P_6$$

Boltzmann distribution

All configurations with the same energy are equally likely

$$P(E_n) = \frac{e^{-E_n/k_B T}}{Z}$$

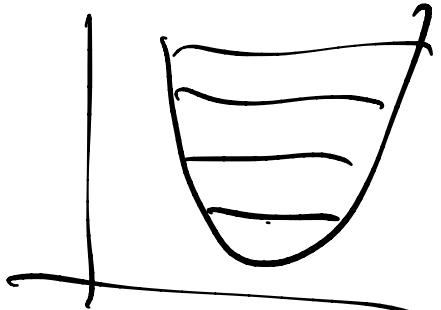
Particle in a box

$$E_n = \frac{\hbar^2 n^2}{8mL^2}$$

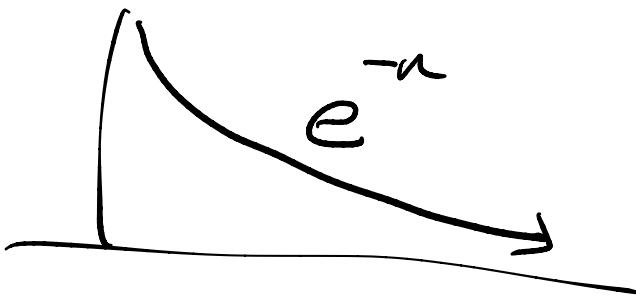
$P(n)$

$$\prod_{n=1}^{\infty} \frac{1}{n^2} e^{-n^2}$$

Harmonische Oszillatoren



$$E_n = \frac{1}{2} \hbar \omega (n + \frac{1}{2})$$



<del>1, 2, 3, 4, 5, 6</del>	3
4	5

$$P(n=1) = \frac{1}{6}$$

$$P(n=1 \cap n=2) = 0$$

↑  
and

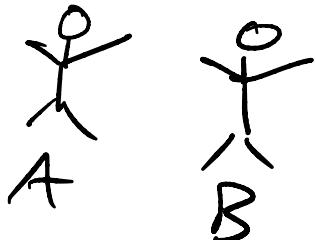
$$P(n=1 \cup n=2) = \frac{1}{6} + \frac{1}{6}$$

↑  
or

$$= \frac{1}{3}$$

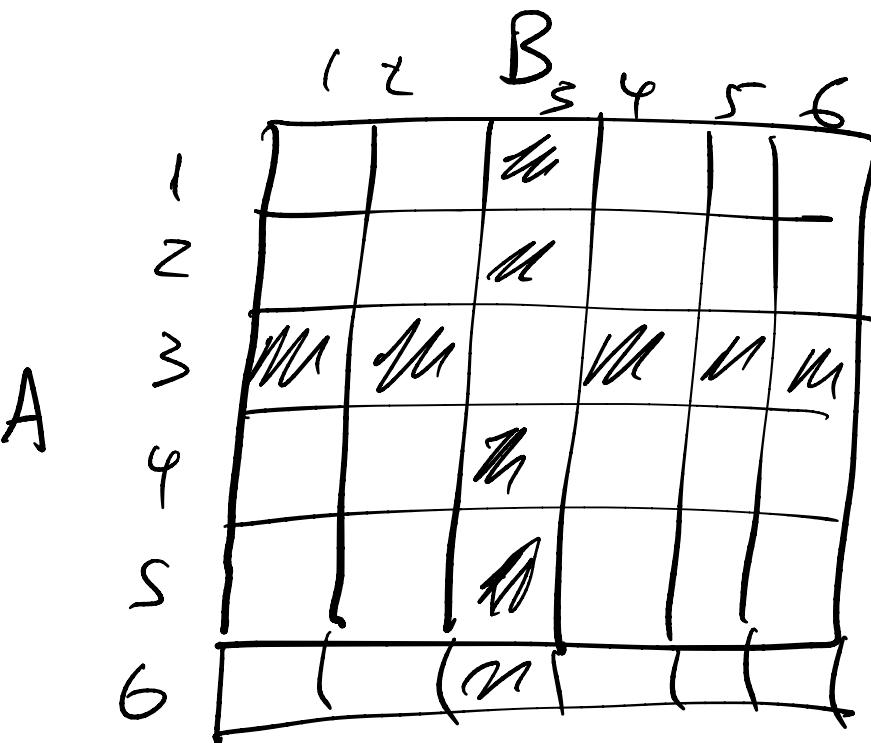
Independent events

$$P(A=1) \cap P(B=2)$$



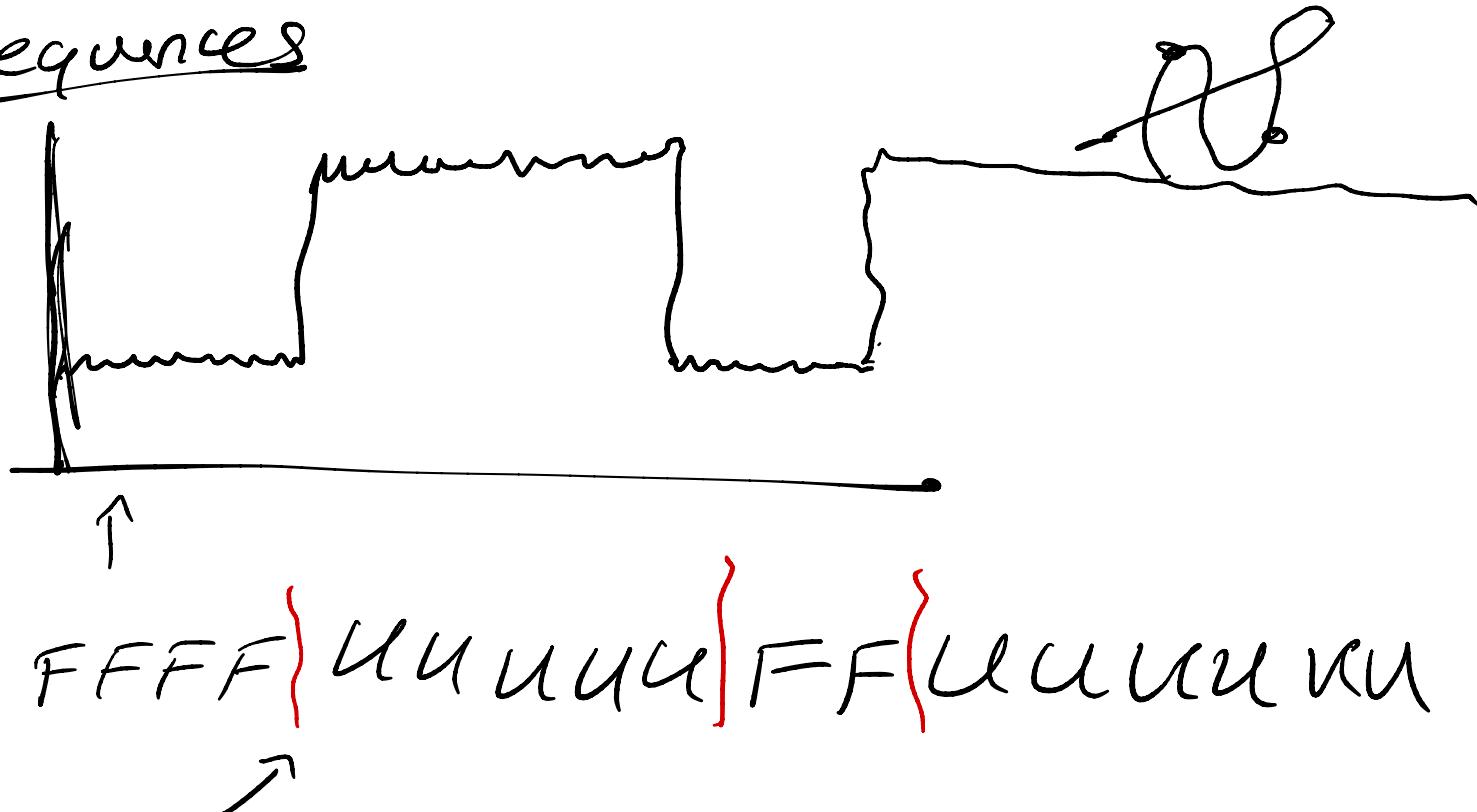
Independent events  $\rightarrow$  multiply

Probability that A & B roll a  
1 and 2

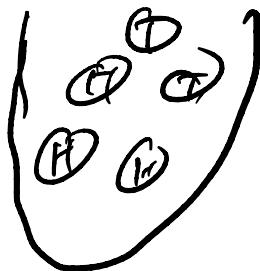


Prob A rolls 3  
or B rolls  
but not both  
= 1

## Sequences



for a given length , how many sequences  
are like this



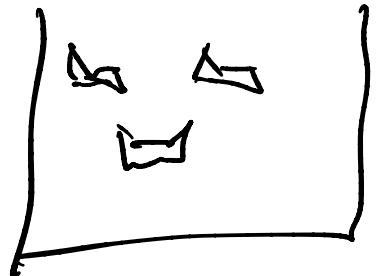
$$2 \times 2 \times 2 \times \dots \times \dots$$

$$\underbrace{H(A) H(\tau) H(T)}$$

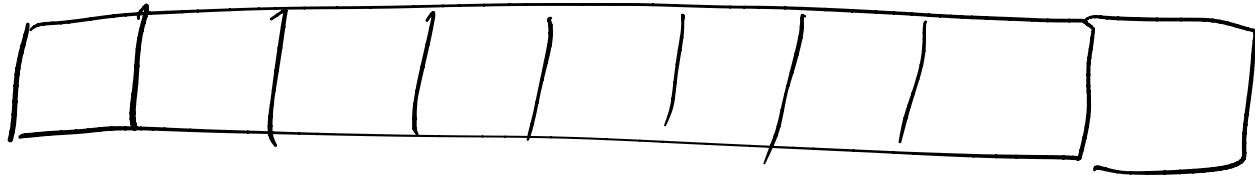
how many possible sequences

$N$  things with 2 states,

# sequences total is  $2^N = 2 \times 2 \dots$



C C C C C C . . . . . B . . . . . C C C



$$P(m, N) = \binom{N}{m} p^m (1-p)^{N-m}$$

Binomial distribution

$$\binom{N}{m} = \frac{N!}{m!(N-m)!} \quad \leftarrow \quad N(N-1)(N-2)\dots$$

hint

$$1 = \sum_{m=0}^N P(m, N) \quad \leftrightarrow$$

"Binomial"

$$(a+b)^N = \binom{N}{0} a^N + \binom{N}{1} a^{N-1} b$$

$$+ \dots + \binom{N}{N-1} a b^{N-1} + \binom{N}{N} b^N$$

$$(a+b) = a^2 + 2ab + b^2$$

Pascal's  
triangle

$$\begin{array}{r} 1 \\ 1 \quad 1 \end{array}$$

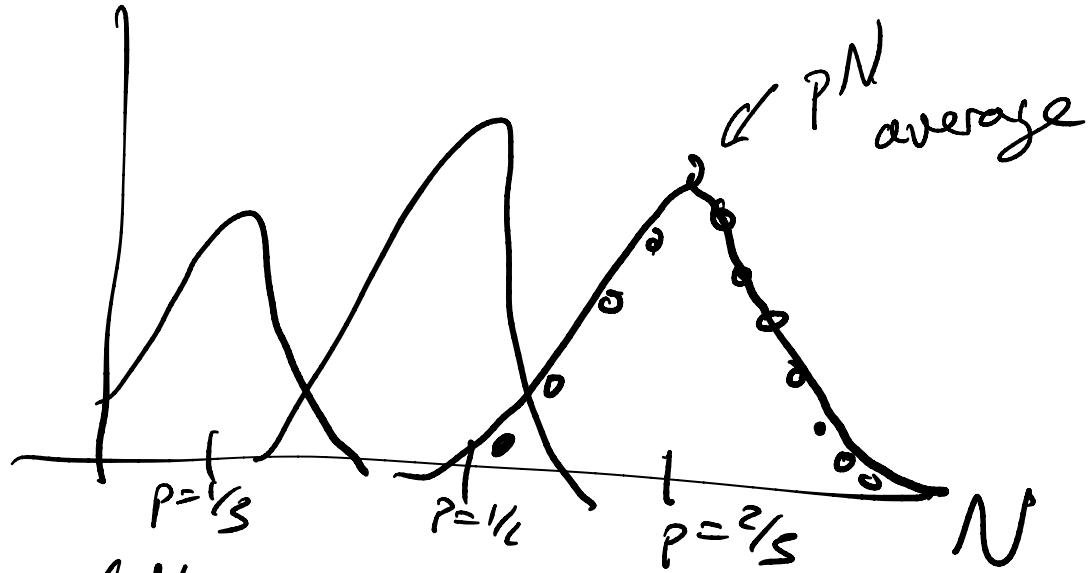
$$\begin{array}{r} 1 \quad 3 \quad 3 \quad 1 \end{array}$$

$$\begin{array}{r} 1 \quad 4 \quad 6 \quad 4 \quad 1 \end{array}$$

$$\begin{array}{r} 1 \quad 5 \quad 10 \quad 10 \quad 5 \quad 1 \end{array}$$

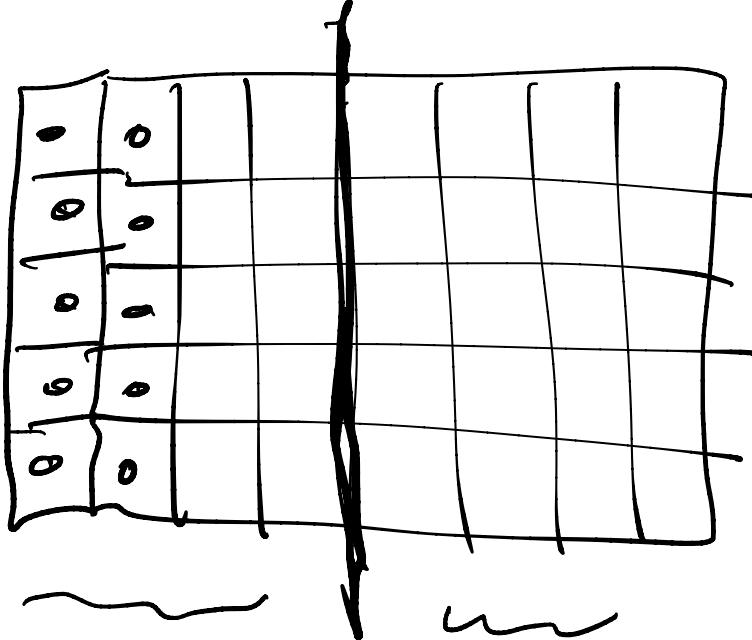
(5)  
(2)

Binomial  
distribution



$$P(m, N) = \binom{N}{m} p^m (1-p)^{N-m}$$

→ as  $N$  gets big → central limit theorem



diffusion

$m$  molecules

$$\binom{N}{m} \quad \binom{N}{0} \rightarrow \binom{N}{m/2} \quad \binom{N}{m/2}$$

# Particle Mixing Simulation Animation

January 25, 2024

This chat contains files or images produced by Advanced Data Analysis which are not yet visible in Shared Chats.

Anonymous

I want to write a python code that simulates mixing of two kinds of particles on an NxN lattice and makes an animation in a jupyter notebook. The representation will be 1s and 0s in a matrix. The initial state will be half full of 1s on the left and half full of 0s on the right. Then every step, two neighboring sites will swap. Images can be generated every N\_image steps

ChatGPT

To create a Python code that simulates the mixing of two kinds of particles on an NxN lattice and generates an animation in a Jupyter notebook, we can follow these steps:

1. **Create the Initial Lattice:** We'll represent the lattice as an NxN matrix filled with 1s on the left half and 0s on the right half.
2. **Simulate the Mixing Process:** At each step, we randomly select two neighboring sites and swap their values. Neighboring sites can be defined as those that are adjacent horizontally or vertically.
3. **Generate Images for the Animation:** Every N\_image steps, we'll capture the state of the lattice to generate an image.
4. **Create an Animation:** We'll compile these images into an animation to visualize the mixing process over time.

Let's start by writing the code for this simulation. We'll use libraries like '`numpy`' for matrix operations and '`matplotlib`' for visualization. We'll also use '`matplotlib.animation`' to create the animation.

Note: The animation part might not work directly in some environments due to limitations in displaying animations in certain Jupyter notebook configurations. However, the code will generate the necessary frames which can be used to create the animation externally if needed.

Let's begin by coding the initial setup and the mixing process. We'll add the animation part afterward.

The initial state of the lattice has been successfully created, with the left half filled with 1s and the right

Get started with ChatGPT

[Report content](#) | [Terms of use](#) | [Privacy policy](#) | [Manage cookies](#)

<https://chemua-652-spring.rcnyu.org/hub/login?next=%2Fhub%2F>

[https://www.dropbox.com/scl/fi/o0dn0nahcu0dzbm459nt9/particle\\_mixing\\_animation.gif?](https://www.dropbox.com/scl/fi/o0dn0nahcu0dzbm459nt9/particle_mixing_animation.gif?)

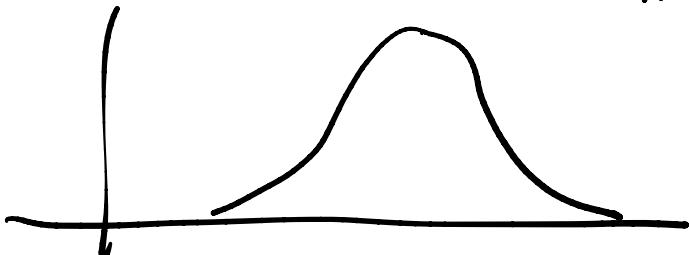
Average

$$\langle A \rangle = \sum_{n=1}^N A(n) P(n)$$

$$\langle 1 \rangle = \sum_{n=1}^N P(n) = 1$$

} average  
of distribution

if  $P(n) = \frac{1}{N}$        $\langle A \rangle = \frac{1}{N} \sum_{n=1}^N A(n)$



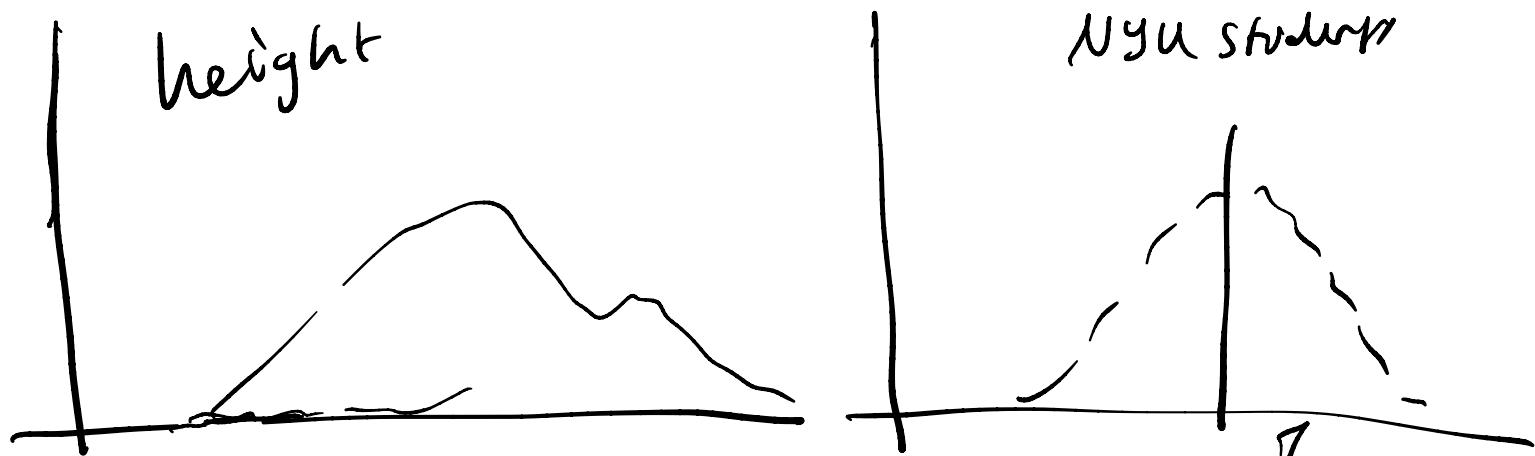
Sample  
Avg

$$\langle A \rangle = \frac{1}{N_{\text{samples}}} \sum_{i=1}^{N_{\text{samples}}} A(i)$$

mass



sample avg  $(m(1) + m(2) + \dots + m(1000000))_{1000000}$



$$(\mu_N - N)^2 \propto \frac{1}{N}$$

Standard deviation

$$\mu = \frac{1}{N} \sum_{i=1}^N A_i$$

$$\text{Variance} = \frac{1}{N} \sum_{i=1}^N (A_i - \mu)^2 = \sigma^2$$