# Random walks

PHYS 246 class 8

https://lkwagner.github.io/IntroductionToComputationalPhysics/intro.html

# Announcements/notes

```
from google.colab import drive
drive.mount('/content/drive')
!cp /content/drive/MyDrive/Colab\ Notebooks/Dynamics.ipynb ./
!jupyter nbconvert --to HTML "Dynamics.ipynb"
```

- 'Classifying galaxies' is due tonight.
- I updated the 'Random walks' notebook this morning with improved instructions.
- Please note that we grade based on what was submitted. The PDF needs to be readable and all work needs to be on the PDF. If you have trouble doing this, ask for help!

# Stochastic processes

Anything with some randomness in its dynamics.

Grains of pollen in a liquid (original Brownian motion)

Financial markets*

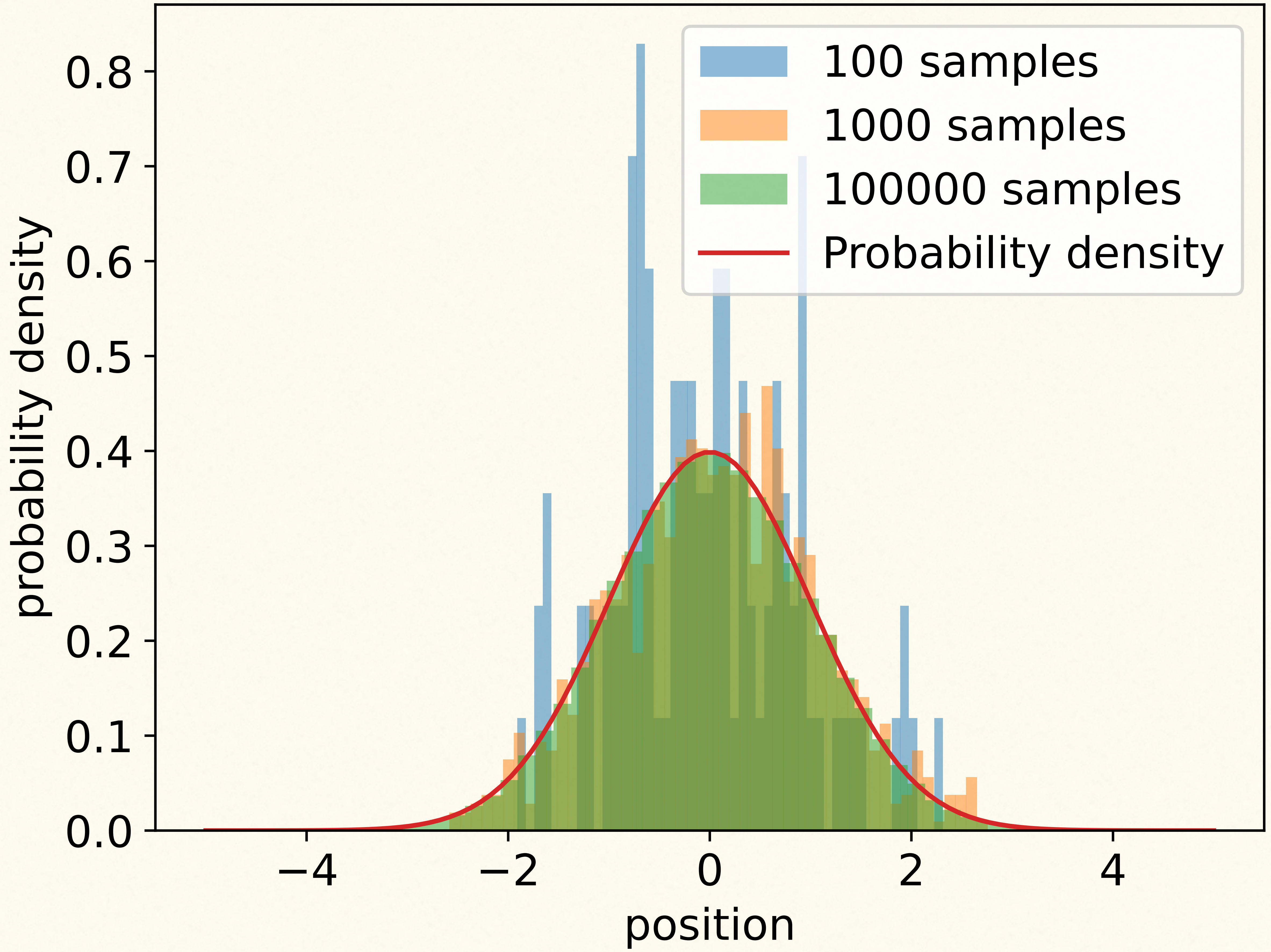
Monte Carlo methods (statistical physics, quantum mechanics, particle physics, ...)

Training machine learning models

You've done some of these!

*it's not uncommon for physics PhDs to end up in finance...

# Histograms and probability densities




```python
for nsample in [100, 1000, 100000]:
    samples = np.random.randn(nsample)
    plt.hist(samples, bins=50, density=True, alpha = 0.5, label=f'{nsample} samples')
x = np.linspace(-5, 5, 100)
y = np.exp(-x**2/2)/np.sqrt(2*np.pi)
plt.plot(x, y, label="Probability density")
plt.xlabel("position")
plt.ylabel("probability density")
plt.legend()

plt.savefig("hist_vs_density.pdf", bbox_inches='tight', transparent = True)
```

Probability density:

$\rho(x) \geq 0$

$$\int \rho(x)dx = 1$$

$\int_a^b \rho(x)dx$ is the proportion of samples on average that lie in the range [a,b]

A histogram plots the proportion of samples in the range [a,b]

# Numerical integration

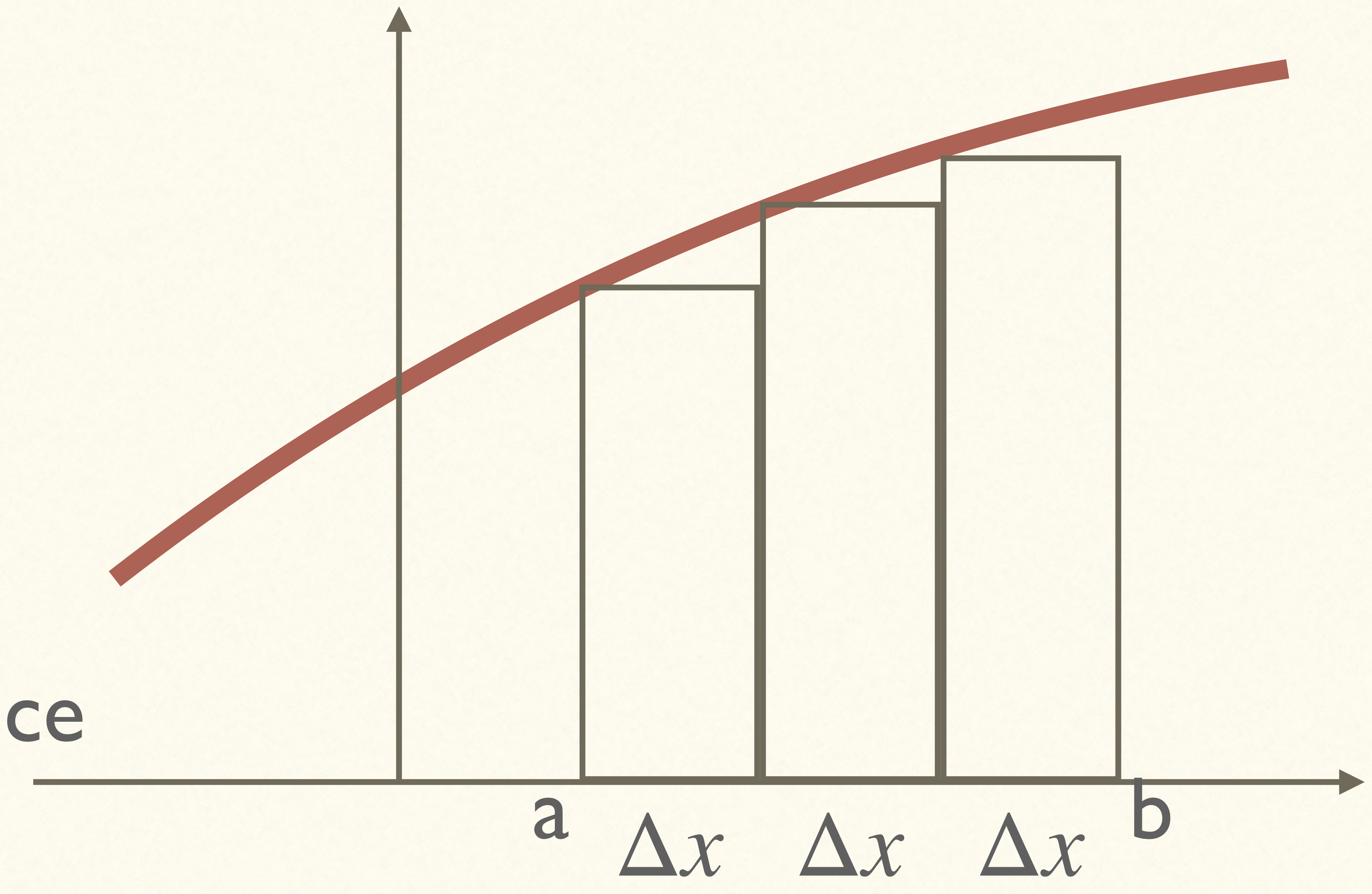
Riemann sums

$$\int_a^b \rho(x)dx \simeq = \sum_i \rho(x_i)\Delta x$$

Sample variance should match the histogram variance

$$\frac{1}{N}\sum x_i^2 \simeq \int x^2\rho(x)dx$$

You should be able to confirm this!



# Diffusion equation

Tells us how the probability density changes in time.

$$\frac{\partial \rho}{\partial t} = D\frac{\partial^2 \rho}{\partial x^2}$$

You can solve this analytically but it's pretty easy to do numerically too.

$$\rho(x, t + \Delta t) = \rho(x) + \Delta t\frac{\partial \rho(x,t)}{\partial t} = \rho(x) + \Delta t D\frac{\partial^2 \rho(x,t)}{\partial x^2}$$

Very similar to Euler integration but you propagate an entire function instead of just a position.

The density computed this way should match the histograms of the simulations you did!

# hints

For the large sample sizes (100,000, 10,000), make sure to test for small numbers of walkers first.

np.roll will move the probability density +/- one grid point -> makes it easy to take second derivative.

Generating walks per sample

```
walks = np.array([randomWalk(1000, 6) for i in range(10)])
```

memory: 1001*nwalkers

Inverting the loop

```
for i in range(t):
    walkers += 6*np.random.randn(N)
```

memory: nwalkers