2.8 Monte-Carlo Method

Idea: use random samplings to approximate expectation values

Ex.: binomial tree model for option price of European calls:

$$C = \sum_{j=0}^{n} b(j,n,p) \underbrace{e^{-rT} \max(S u^j d^{n-j} - K, 0)}_{= f(j,n)} = \mathbb{E}_{\text{bin.}}(f)$$

Monte-Carlo: take m samples $j_1, \dots, j_m$ from bin. distribution ($m \ll n$) and compute

$\frac{1}{m} \sum_{k=1}^{m} f(j_k, n)$, the empirical mean, or sample average.

The law of large numbers says that $\boxed{\frac{1}{m} \sum_{k=1}^{m} f(j_k, n) \xrightarrow{m \to \infty} \underbrace{\mathbb{E}(f)}_{\text{theoretical expectation}}.}$

(for very general classes of probability distributions)

Idea/hope of Monte-Carlo method:

- time efficient / fast, since only $m \ll n$ steps are necessary to compute a good approximation
- interesting idea: use randomness to approximate a deterministic quantity

Summary:

- skills:
  - git
  - python / scipy: basics, vector-based coding, timing, plotting, csv files

- finance:
  - cash flows, interest compounding
  - bonds, immunization
  - options (European and American calls and put)
  - option pricing with binomial trees (important concepts: $\underbrace{\text{no arbitrage}}_{\text{no risk-free profit}}$, replicating portfolio)
    - ↳ binomial tree implemented in python
    - ↳ explicit formula for European calls (possibly with Monte-Carlo)
    - ↳ Black-Scholes formula for European calls
    - ↳ put-call parity (to compute put price, given call price): $\underset{\text{call prize}}{C} - \underset{\text{put prize}}{P} = S - Ke^{-rT}$

- numerical methods:
  - root finding
  - how to find convergence rates
  - QQ plots
  - Monte-Carlo

- math:
  - Taylor expansion
  - binomial and normal distribution
  - CLT

Focus of second half of class:
- numerical methods
- option pricing with continuous stochastic processes

3. Continuous Time Models

3.1 Brownian Motion

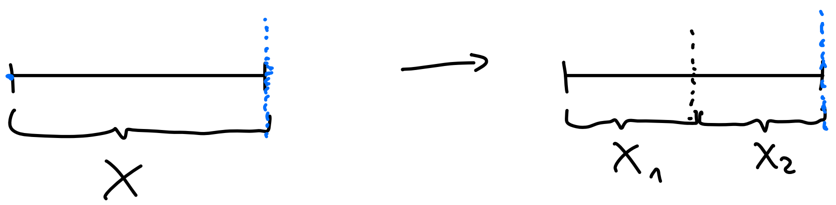
Motivation: Let us consider the normal distribution with mean 0 and variance 1:

$$\underbrace{\mathcal{N}(0,1)}_{\text{mean} \nearrow \;\; \nwarrow \text{standard deviation}} \qquad (\mathcal{N}(\mu,\sigma) \text{ for mean } \mu, \text{ std. deviation } \sigma)$$

Importance of the normal distribution comes from the Central Limit Theorem (CLT)!
(Which holds under very weak conditions on the underlying probability distribution; in particular, random variables need to be independent or only weakly dependent.)

Now consider a random variable X that is normally distributed: $X \underset{\text{is distributed according to}}{\sim} \mathcal{N}(0,1)$

Consider splitting it into two: $X = X_1 + X_2$, s.t. X_1 and X_2 independent and same distribution.



Note: $0 = \mathbb{E}(X) = \mathbb{E}(X_1 + X_2) = \mathbb{E}(X_1) + \mathbb{E}(X_2) \overset{X_1 \text{ and } X_2 \text{ same distribution}}{=} 2\,\mathbb{E}(X_1) \Rightarrow X_1$ has expectation 0

How does the variance behave?

$$1 = \mathrm{Var}(X) = \mathrm{Var}(X_1 + X_2) \overset{X_1 \text{ and } X_2 \text{ independent}}{=} \mathrm{Var}(X_1) + \mathrm{Var}(X_2) \overset{\text{same dist.}}{=} 2\,\mathrm{Var}(X_1) \Rightarrow \mathrm{Var}(X_1) = \frac{1}{2}$$

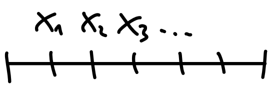
Computing higher moments shows that $X_1 + X_2$ is normally distributed.

$$\Rightarrow X_1 \text{ is distributed according to } \underbrace{\mathcal{N}\left(0, \tfrac{1}{\sqrt{2}}\right) = \tfrac{1}{\sqrt{2}}\,\mathcal{N}(0,1)}$$

$\leftarrow \operatorname{Var}\left(\frac{1}{\sqrt{2}} X\right) = \frac{1}{2} \operatorname{Var}(X)$

both are normal distributions with mean 0 and variance $\frac{1}{2}$, so they are the same

for n steps:



$$1 = \operatorname{Var}(X) = \operatorname{Var}\left(\sum_{i=1}^{n} X_i\right) = \sum_{i=1}^{n} \operatorname{Var}(X_i) = n \operatorname{Var}(X_1)$$

$$\Rightarrow \operatorname{Var}(X_1) = \frac{1}{n}$$

$$\Rightarrow \text{each } X_i \sim \mathcal{N}\left(0, \tfrac{1}{\sqrt{n}}\right) = \tfrac{1}{\sqrt{n}}\,\mathcal{N}(0,1)$$

or, calling $\frac{1}{n} = \Delta t$: $X_i \sim \sqrt{\Delta t}\,\mathcal{N}(0,1)$