Monte Carlo Risk & Option Pricing Toolkit – Baseline Project: Step-by-Step Plan

PHASE 0 — Project Setup
---
1. Create project folder containing:
   - README.md
   - requirements.txt
   - .gitignore
   - directories: notebooks/, src/, data/, figures/, tests/

2. Install dependencies:
   pip install -r requirements.txt

3. Initialize Git repository:
   git init
   git add .
   git commit -m "Initial commit"

4. Create private GitHub repo (empty).

5. Add SSH remote:
   git remote add origin git@github.com:USERNAME/REPO.git
   git branch -M main

6. Push local repo (force if needed):
   git push --force origin main

PHASE 1 — GBM Simulation

– Implement simulate_gbm_paths in src/gbm.py using:
  $S(t+\Delta t) = S(t) * \exp[(\mu - 0.5\,\sigma^2)\Delta t + \sigma \sqrt{\Delta t}\, Z]$
– Generate N paths, M steps.
– Test and visualize in notebooks/MonteCarloToolkit.ipynb.
– Verify empirical mean/variance vs theoretical values.

PHASE 2 — Correlated Multi-Asset GBM

– Implement simulate_correlated_gbm_paths using Cholesky L:
  $W = L\,Z$
– Simulate 2 correlated assets for VaR.
– Plot correlation scatter and verify empirical correlation.

PHASE 3 — European Option Pricing (Monte Carlo)

– Implement payoff functions:
  call: max(S_T – K, 0)
  put: max(K – S_T, 0)

– Monte Carlo pricing under risk-neutral measure:
  price = exp(–rT) * mean(payoffs)

– Confidence intervals:
  CI = price ± 1.96 * std / sqrt(N)

– Compare with Black–Scholes closed-form price.

– Convergence analysis: price vs number of paths.

PHASE 4 — Portfolio VaR

– Simulate correlated returns for 2 assets.
– Build portfolio P&L;:
  R_p = w1 R1 + w2 R2
  PnL = V0 * R_p

– Compute:
  VaR_α: α-quantile of losses
  CVaR_α: mean loss beyond VaR

– Plot histogram with VaR markers.

PHASE 5 — Documentation & Polish

– Finalize notebook with clear sections.
– Save plots in figures/.
– Add unit tests in tests/.
– Clean up README with summary and usage.
– Ensure reproducibility with fixed seeds.

END OF BASELINE PROJECT PLAN