THE COMPLETE

MONTE CARLO SIMULATION

PROMPT GUIDE FOR CLAUDE AI

20 Ready-to-Copy Prompts · GBM / VaR / Options · 3D Dashboards · Live Market Data
NIFTY 50 · S&P 500 · Stocks · Crypto · Portfolios

20 Prompts	6 Levels	Copy & Use	Mar 2026

Built with Claude AI · Designed for quant finance enthusiasts, traders & content creators.

Paste any prompt directly into Claude. Swap the [BRACKETS] for your own instrument, timeframe, or data.

CONTENTS

What's Inside

Navigate to any section quickly

SECTION 1

Monte Carlo Foundations

What it is, how it works, why it matters for markets

What Is a Monte Carlo Simulation?

A Monte Carlo simulation runs thousands of possible future price paths for a financial asset, each driven by random chance within historical statistical bounds. The result isn't a single prediction — it's a probability distribution of outcomes.

Think of it like this: instead of asking 'where will NIFTY be in 30 days?', you ask 'across 10,000 possible futures, what does the range of NIFTY prices look like?' That range tells you far more than any single forecast.

The Core Formula — Geometric Brownian Motion (GBM)

∑ GBM DAILY PRICE STEP

$$\text{Price}(t+1) = \text{Price}(t) \times \exp\left((\mu - 0.5\sigma^2)\cdot\Delta t \;+\; \sigma\cdot\sqrt{\Delta t}\cdot Z \right)$$

SYMBOL	MEANING
μ (mu)	Mean daily return (drift) — estimated from historical log returns
σ (sigma)	Daily volatility — std dev of historical log returns
Δt	Time step = 1 (one trading day)
Z	Random draw from Standard Normal distribution N(0,1)
exp(…)	Exponential function — ensures price can never go below zero
P(t+1)	Simulated price tomorrow given today's price P(t)

The Three Core Parameters

📌 **NOTE**

Before writing any Monte Carlo prompt, you need three inputs: a starting price, a drift (μ), and a volatility (σ). Everything else flows from these.

PARAMETER	WHERE TO GET IT
Starting Price	Latest closing price from your data or a live web search
Daily Drift (μ)	Mean of log returns = mean(ln(P_t / P_{t-1})) from historical data
Daily Volatility (σ)	Std dev of log returns from historical data — OR use VIX/100/√252 for implied vol

SECTION 2

Starter Prompts

5 beginner-friendly prompts — copy, paste, get results

PROMPT 01 [BEGINNER] Minimal Monte Carlo Simulation

```
You are a Python quant analyst. Run a Monte Carlo simulation for [NIFTY 50 / S&P 500 /
TICKER].

Parameters:
- Current price: [24,158]
- Annual volatility: [22%]
- Annual drift: [9%]
- Simulations: 10,000
- Horizon: 30 trading days

Use the GBM formula:
  Price(t+1) = Price(t) * exp((mu - 0.5*sigma^2)*dt + sigma*sqrt(dt)*Z)

Convert annual to daily: mu_daily = mu/252, sigma_daily = sigma/sqrt(252)

Print: Expected price, Median, P5, P95, P(gain), P(loss)
Then generate a clean HTML page showing a paths chart and results table.
```

💡 **CUSTOMISE:** Replace [24,158] with today's price. Replace [22%] / [9%] with your instrument's real volatility and expected annual return.

PROMPT 02 [BEGINNER] Simulation with Histogram & Distribution

```
You are a quantitative finance analyst. Run a Monte Carlo simulation for [INSTRUMENT]:
- Starting price: [PRICE]
- Historical annual volatility: [VOL%]
- Annual drift: [DRIFT%]
- Simulations: 10,000 paths, Horizon: 30 days

After simulating, produce an HTML page with TWO Plotly charts:

Chart 1 — PATHS: Plot 200 sample paths as thin translucent lines.
  Color paths green if final price > start, red if below.
  Draw the mean path as a thick white line.

Chart 2 — HISTOGRAM: Distribution of final prices at day 30.
  Color bars: red if price < current-5%, blue if within 5%, green if > +5%
  Add vertical dashed lines for: Current, Expected, P5, P95
  Label each line.

Style: Dark background (#050A0F), monospace font, no axis borders.
```

💡 **CUSTOMISE:** The two-chart combo is the most shareable format. Great for social media posts.

PROMPT 03 [BEGINNER] Confidence Bands Chart

```
Build a confidence interval band chart for a Monte Carlo simulation of [INSTRUMENT].

Run 10,000 GBM paths. Starting price [PRICE], daily mu = [MU], daily sigma = [SIGMA].
Horizon: 30 trading days.

Compute at each time step: P5, P25, Mean, P75, P95 paths.

Generate HTML with a single Plotly chart:
- Shaded area between P5 and P95 (colour: rgba(0,180,255,0.08))
- Shaded area between P25 and P75 (colour: rgba(0,180,255,0.15))
- Mean path: bright cyan, width 3
- P5 line: dashed red, P95 line: dashed green
- Horizontal dashed white line at current price
- X-axis: 'Trading Day', Y-axis: 'Price (₹)'
- Dark theme: paper_bgcolor='#050A0F', plot_bgcolor='#050A0F'
- Add annotation showing expected price at day 30
```

💡 **CUSTOMISE:** This is the clearest chart for showing 'where will price be'. Use it for explainer content.

PROMPT 04 [BEGINNER] Probability Summary Table

```
Run a Monte Carlo simulation for [INSTRUMENT] with:
- Start price: [PRICE], Annual vol: [VOL%], Annual drift: [DRIFT%]
- 10,000 simulations, 30 trading days

After simulating, calculate ALL of these probabilities:
- P(Price increases from start)
- P(Price decreases from start)
- P(+2% gain)     P(-2% loss)
- P(+5% gain)     P(-5% loss)
- P(+10% gain)    P(-10% loss)
- P(above [KEY LEVEL 1])  e.g. above 25,000
- P(below [KEY LEVEL 2])  e.g. below 23,000

Display as an HTML table with:
- Left column: Scenario name
- Middle column: Target price
- Right column: Probability % (green for bullish, red for bearish)
Dark theme, bold font for probabilities, hover highlighting on rows.
```

💡 **CUSTOMISE:** Replace [KEY LEVEL 1] and [KEY LEVEL 2] with psychologically important price levels: round numbers, 52-week highs/lows, support/resistance.

PROMPT 05 [BEGINNER] Full Beginner Dashboard (All-in-One)

```
You are a quant analyst and Python developer. Build a complete Monte Carlo dashboard for
[NIFTY 50].

Data: Historical monthly prices (provide your data or describe columns here)
Steps:
1. Load data → sort by date → compute log returns
2. Estimate: mu (mean daily return), sigma (daily std dev)
```

```
3. Run 10,000 GBM simulations over 30 trading days
4. Calculate: Expected price, Median, P5/P25/P75/P95, all probabilities

Build a single HTML file with FOUR sections:
SECTION A — Market Summary: Current price, Annual vol %, Annual drift %
SECTION B — Simulation Results: A table of all percentile prices + changes from current
SECTION C — Probability Grid: Colour-coded cards showing key probabilities
SECTION D — Charts: Paths chart + Histogram + Confidence bands (all Plotly, dark theme)

Include a short 3-sentence interpretation at the bottom.
Make it look like a Bloomberg terminal. Output complete HTML only.
```

💡 **CUSTOMISE:** This is the prompt that built the NIFTY dashboard you saw. Start here, then add 3D charts using Section 4 prompts.

SECTION 3

Intermediate Prompts

Blended volatility, VIX regimes, realistic parameter estimation

PROMPT 06 [INTERMEDIATE] Blended Volatility (VIX + Historical)

```
Build a Monte Carlo simulation for [NIFTY 50] using blended volatility:

Inputs:
- Current price: [24,158]
- India VIX (implied annual vol): [23.36%]
- Historical annual vol from 18yr data: [20.14%]
- Blend ratio: 70% implied + 30% historical
- Blended sigma_annual = 0.70 * (VIX/100) + 0.30 * hist_vol

Convert to daily:
- mu_daily = annual_drift / 252
- sigma_daily = blended_sigma_annual / sqrt(252)

Run 10,000 GBM simulations over [15] trading days.

In the dashboard, show a side-by-side comparison:
- Historical vol simulation results vs Blended vol results
- Which gives wider confidence interval and why
Explain: 'Current VIX of 23.36% signals elevated fear — the market is pricing in more risk
than the 18-year average.'
```

💡 **CUSTOMISE:** Always use blended vol when VIX is elevated (>20). Historical vol alone understates current market risk during stressed periods.

PROMPT 07 [INTERMEDIATE] Volatility Regime-Aware Simulation

```
Build a regime-aware Monte Carlo simulation for [INSTRUMENT].

Step 1 — Identify volatility regimes from historical data:
  LOW regime:    Annual vol < 15%  (calm markets)
  MEDIUM regime: Annual vol 15-25% (normal markets)
  HIGH regime:   Annual vol > 25%  (stressed markets)
  CRISIS regime: Annual vol > 35%  (crash / GFC-like)

Step 2 — Estimate separate mu and sigma for each regime from historical data.

Step 3 — Current regime: [HIGH — VIX at 23.36]
  Run simulation using HIGH-regime parameters.
  Also run with MEDIUM-regime parameters as a 'recovery scenario'.

Step 4 — Dashboard shows:
  - Regime statistics table (4 regimes × avg return, vol, P(gain) in each)
  - Two side-by-side fan charts: current regime vs recovery scenario
  - Regime history chart: shaded bands showing which regime was active each year
  - Interpretation: 'In HIGH volatility regimes historically, NIFTY has...'
```

💡 **CUSTOMISE:** This is the most realistic simulation approach. It stops you from using calm-period parameters during a crisis.

PROMPT 08 [INTERMEDIATE] Parameter Sensitivity Analysis

```
Build a Monte Carlo sensitivity dashboard for [INSTRUMENT].

Base case: Start = [24,158], mu = [0.04%/day], sigma = [1.41%/day]

Run the simulation 9 times varying parameters:
  Sigma scenarios: Low vol [0.8%], Base [1.41%], High vol [2.2%]
  Mu scenarios:    Bear [-0.05%/day], Neutral [0.04%], Bull [0.12%/day]

Create a 3x3 grid of mini-charts (Plotly subplots) showing the confidence bands
for each parameter combination. Title each: e.g. "Low Vol + Bear Drift"

Also build a sensitivity table:
  Rows = Sigma scenarios, Columns = Mu scenarios
  Each cell shows: Expected Price | P(Gain) | P5 | P95

Highlight the BASE CASE cell in cyan. Color cells green if P(gain)>55%, red if <45%.
```

💡 **CUSTOMISE:** This prompt is great for content — the 3x3 grid shows your audience how vol and drift interact. Very visual, very shareable.

PROMPT 09 [INTERMEDIATE] Rolling Monte Carlo (Backtest Accuracy)

```
You are a quant analyst. Backtest the accuracy of Monte Carlo simulations using historical
data for [NIFTY 50].

Method — Rolling window backtest:
1. For each month from Jan 2010 to Oct 2024:
   a. Use the previous 36 months of data to estimate mu and sigma
   b. Run a Monte Carlo 30-day simulation from that month's start price
   c. Record: Predicted P5, Mean, P95 at month-end
   d. Record: Actual price at month-end

2. Calculate coverage statistics:
   - How often did actual price fall within P5-P95? (should be ~90%)
   - Average error: |Predicted mean - Actual| / Actual * 100
   - Bias: Was simulation systematically too high or too low?

3. Dashboard shows:
   - Time series of predicted range vs actual price
   - Coverage ratio chart (rolling 12-month)
   - Error distribution histogram
   - Summary: 'The GBM model captured [X]% of outcomes within the 90% CI'
```

💡 **CUSTOMISE:** This is the most rigorous prompt in the guide. It proves — or disproves — how accurate Monte Carlo is on your specific instrument.

SECTION 4

Advanced 3D Interactive Charts

3D Monte Carlo visualisations using Plotly — drag, rotate, zoom

📌 **NOTE**
All 3D charts below use Plotly's scatter3d and surface trace types. They are fully interactive in a browser — drag to rotate, scroll to zoom, hover for values.

PROMPT 10 [ADVANCED] 3D Monte Carlo Path Tornado

```
Build a 3D Monte Carlo path chart for [INSTRUMENT] — the 'tornado' view.

Run 10,000 GBM simulations. Sample 500 paths for 3D rendering.

Create a Plotly scatter3d chart:
  X-axis = Trading Day (0 to 30)
  Y-axis = Simulation index (0 to 499)
  Z-axis = Simulated Price

Path colouring:
  - Final price > current: colour = rgba(0,255,100, 0.06) — faint green
  - Final price < current: colour = rgba(255,50,50, 0.06)  — faint red
  - Path opacity: 0.08, line width: 0.8

Special overlays:
  - Mean path: cyan, width = 6, plotted at Y = 250 (midpoint)
  - P95 path: dashed green, width = 3, Y = 490
  - P5 path: dashed red, width = 3, Y = 10

Scene: dark background #030610, cyan grid lines, camera eye at x=1.8 y=1.2 z=0.9
Title: 'NIFTY 50 — 10,000 GBM Simulations · 30 Trading Days'
Add subtitle annotation: 'Drag to rotate · Scroll to zoom'
```

💡 **CUSTOMISE:** The tornado chart is the most impressive visualisation for social media. The coloured paths create a stunning fan effect.

PROMPT 11 [ADVANCED] 3D Probability Density Surface

```
Build a 3D probability density surface — showing HOW LIKELY each price is at each future
date.

Run 10,000 GBM simulations for [INSTRUMENT].

At each time step t = 0, 1, 2, ... 30:
  Compute a histogram (40 bins) of simulated prices at that step.
  Normalise to probability density (density=True in numpy).

Build a Plotly surface3d chart:
  X-axis = Trading Day (0 to 30)
```

```
  Y-axis = Price bins (P1 to P99 of final prices, 40 bins)
  Z-axis = Probability density

Colourscale (dark to bright as density increases):
  [0, '#030610'], [0.2, '#003060'], [0.5, '#006090'],
  [0.75, '#00C0D0'], [0.9, '#00FF88'], [1.0, '#FFFF00']

Enable contour projection on the Z floor.
Add a horizontal plane at Z=0 (transparent base).
Camera: eye at x=1.8, y=-1.5, z=1.2 for best wave view.
The chart should look like an ocean wave — narrow at day 0, spreading wide by day 30.
```

💡 **CUSTOMISE:** This is the 'wow' chart. The expanding wave shows uncertainty growing over time. Perfect for explaining why forecast ranges widen.

PROMPT 12 [ADVANCED] 3D Scatter Cloud (Outcome Space)

```
Create a 3D scatter cloud visualising the full Monte Carlo outcome space for [INSTRUMENT].

Run 10,000 simulations. Sample 500 paths × every 3rd day = ~2,500 data points.

Each point represents (Simulation #, Day, Price at that day).

Plotly scatter3d with marker:
  X = Simulation index, Y = Day, Z = Price
  Size = 1.5 (small dots for density effect)
  Colour = Price value (Z), using diverging colourscale:
    Low prices  → red  (#FF2D55)
    Mid prices  → gold (#FFCC00)
    High prices → cyan (#00F5FF)
    Very high   → green (#00FF88)
  Opacity = 0.65

The cloud should be dense in the middle (near current price) and thin at extremes.
Add 3 vertical reference planes at P5, Mean, P95 prices using mesh3d.
Dark background. Camera at x=1.6 y=1.6 z=1.0.
```

💡 **CUSTOMISE:** Combine this with the density surface in a tabbed dashboard for maximum impact. The contrast between the two 3D charts is stunning.

PROMPT 13 [ADVANCED] 2D Heatmap (Price-Time Density)

```
Build a 2D heatmap of simulation density for [INSTRUMENT].

Run 10,000 GBM simulations over 30 days.

At each time step, bin prices into 50 buckets between P1 and P99.
Create a 2D density matrix: rows = price bins, cols = time steps.
Normalise each column so the sum = 1 (conditional density at each day).

Plotly heatmap:
  X = Day (0 to 30), Y = Price bins
  Z = Density values
```

```
  Colourscale: dark blue (low) → cyan → green → yellow (high density)
  No gap between cells.

Overlay three lines on top of the heatmap:
  - White dashed: Current price (horizontal)
  - Cyan solid: Mean path
  - Red dashed: P5 path, Green dashed: P95 path

Add annotation at day 30 showing expected price.
Use dark background. This chart looks like a thermal imaging scan.
```

💡 **CUSTOMISE:** The heatmap is the most 'data-dense' chart — every pixel carries information. Great for advanced/technical audiences.

SECTION 5

Live Market Data Prompts

Prompts that search the web for real-time data before simulating

⚠ **WARNING**
These prompts ask Claude to search the web for live data. Always verify any fetched prices and statistics before using them for real decisions.

PROMPT 14 [LIVE DATA] Live NIFTY Simulation (Web Search)

```
You are a quant analyst. Before running the simulation, SEARCH THE WEB for current data on
NIFTY 50:

Search for:
1. "NIFTY 50 current price today"    → get current price
2. "India VIX today"                 → get implied volatility
3. "FII DII flows March 2026"        → get institutional flow data
4. "USD INR rate today"              → get Rupee exchange rate
5. "NIFTY 52 week high low 2026"     → get range context

Once you have live data, run a Monte Carlo simulation:
- Starting price: [fetched live price]
- Blended vol: 70% * (India VIX/100) + 30% * 0.2014 (historical)
- Daily drift: 0.000356 (from 18-year historical average)
- 10,000 simulations, 15 trading days (rest of current month)

Build HTML dashboard with:
1. Live data ticker tape (all fetched values)
2. Simulation results with current market context callout box
3. Monte Carlo paths + distribution charts (Plotly, dark theme)
4. FII vs DII context: 'FII selling [amount] but DII buying [amount]...'
```

💡 **CUSTOMISE:** Add 'Search as of [today's date]' in your prompt to help Claude find the most recent data.

PROMPT 15 [LIVE DATA] Any Global Index — Auto-Fetch & Simulate

```
Search the web for current data on [S&P 500 / NASDAQ / DAX / FTSE / NIKKEI]:

Fetch:
- Current index level and today's % change
- Current CBOE VIX (or equivalent fear gauge)
- Year-to-date return
- 52-week range

Then build a Monte Carlo simulation:
- Start: fetched current level
- Sigma: (fetched VIX / 100) / sqrt(252) for daily vol
- Mu: based on 20-year historical average annual return for this index
  (S&P 500: ~10%, NASDAQ: ~12%, DAX: ~9%, FTSE: ~7%, NIKKEI: ~8%)
- 10,000 simulations, 21 trading days (1 month)
```

```
Dashboard must show:
- Flag/icon for the index country
- All fetched live statistics
- Simulation fan chart with confidence bands
- Key probabilities: P(index above [round number] by month-end)
- Brief macro commentary based on current VIX level
```

💡 **CUSTOMISE:** Works for any major index. Change the historical return assumption to match the index's long-run average.

SECTION 6

Simulation Variations

Portfolio, options, crypto, intraday — extend the core model

PROMPT 16 [PORTFOLIO] Correlated Multi-Asset Portfolio

```
Build a correlated Monte Carlo portfolio simulation for a portfolio of:
  - NIFTY 50:    [50%] weight
  - Gold:        [20%] weight
  - US Bonds:    [15%] weight
  - USD/INR FX:  [15%] weight

Using historical monthly data provided:
1. Calculate the 4x4 correlation matrix between log returns
2. Use Cholesky decomposition to generate correlated random shocks:
   Z_correlated = Cholesky(Correlation_Matrix) × Z_independent
3. Simulate each asset's daily price path using GBM with correlated Z
4. Compute daily portfolio value = sum of weighted asset values

Run 10,000 portfolio paths over 30 days. Calculate:
- Portfolio VaR (95%) and CVaR
- Portfolio Expected Return and Std Dev
- Probability of portfolio loss > [5%]
- Diversification benefit: compare portfolio vol vs weighted-average vol

Dashboard: correlation heatmap, portfolio paths fan chart, risk metrics table.
```

💡 **CUSTOMISE:** Cholesky decomposition ensures your simulated assets move together realistically. Without it, assets are independent — unrealistic.

PROMPT 17 [OPTIONS] Monte Carlo for Options Pricing

```
Use Monte Carlo simulation to price a European [CALL/PUT] option on [NIFTY 50]:

Option details:
  - Current underlying price S: [24,158]
  - Strike price K: [25,000]
  - Time to expiry T: [21 trading days / 252]
  - Risk-free rate r: [6.5% / 252] per day (India 10Y bond)
  - Implied volatility sigma: [23.36% / sqrt(252)] per day

Simulation steps (risk-neutral pricing):
  Use drift = r (risk-free rate), NOT historical drift
  Run 100,000 paths (options pricing needs more paths for accuracy)
  At expiry compute payoff:
    CALL: max(S_T - K, 0)
    PUT:  max(K - S_T, 0)
  Discount: Option_Price = mean(payoffs) × exp(-r×T)

Compare with Black-Scholes analytical price.
Show: MC price, BS price, difference, convergence chart (price vs N simulations),
```

```
Greeks estimated numerically from MC. HTML output, dark theme.
```

💡 **CUSTOMISE:** Always use risk-neutral drift (r) not historical drift for options pricing. Using historical drift gives wrong prices.

PROMPT 18 [CRYPTO] Crypto Monte Carlo (Fat Tails)

```
Build a Monte Carlo simulation for [BITCOIN / ETH / any crypto] accounting for fat tails.

Problem with standard GBM for crypto: Crypto has extreme moves (>5 sigma) far more often
than Normal distribution predicts.

Solution — use Student-t distribution instead of Normal:
  Z_fat = t.rvs(df=3) / sqrt(3/(3-2))    # normalise so std dev = 1
  (degrees of freedom = 3 gives heavy tails matching crypto behaviour)

Inputs: Current BTC price [$[PRICE]], daily vol sigma = [3.5%], daily drift = [0.05%]

Run 10,000 simulations for [30] days using fat-tail Z.
Also run with Normal Z. Compare results side-by-side:
  - Fat-tail P5 will be MUCH lower than Normal P5
  - Fat-tail P95 will be MUCH higher than Normal P95

Dashboard shows both simulations, with commentary:
  'Using Student-t (df=3): P5 = $X  vs  Normal: P5 = $Y — a [N]% difference in tail risk'
```

💡 **CUSTOMISE:** For stocks use Normal. For crypto, small-caps, or during crises — always use fat-tail distributions. The difference is enormous at the tails.

PROMPT 19 [INTRADAY] Intraday Monte Carlo (5-min data)

```
Build an intraday Monte Carlo simulation using [NIFTY / STOCK] 5-minute data.

Data: 5-minute OHLCV data for the last [20 trading days]
Steps:
1. Calculate 5-minute log returns
2. Estimate: mu_5min (mean 5-min return), sigma_5min (std dev of 5-min returns)
3. Note: intraday has U-shaped volatility (higher at open and close)
   Segment the day into: Open (9:15-10:15), Midday (10:15-14:30), Close (14:30-15:30)
   Estimate separate sigma for each segment.

Run 5,000 intraday GBM simulations for rest of today ([N] remaining 5-min bars).
Use the appropriate segment sigma at each time step.

Show:
- Intraday path fan chart (X = time, not day number)
- Probability of hitting [key intraday level] before close
- Expected closing price with confidence range
- Compare: flat-vol GBM vs segmented-vol GBM
```

💡 **CUSTOMISE:** Intraday MC requires much more data (5-min bars). Without U-shape vol adjustment, the simulation is unrealistic at market open and close.

PROMPT 20 [LONG-TERM] Monthly Return Distribution for Year-End Forecast

```
Build a year-end Monte Carlo forecast for [NIFTY 50] using monthly returns.

Why monthly? For 12-month forecasts, monthly returns are less noisy than daily.

From 18-year historical monthly data:
1. Compute monthly log returns
2. Estimate: mu_monthly, sigma_monthly
3. Convert: Annual drift = mu_monthly * 12, Annual vol = sigma_monthly * sqrt(12)

Run 10,000 simulations of 12 monthly steps (not 252 daily steps).

Additional analysis:
- What are the chances of ending December [2026] above [28,000]?
- What are the chances of a 20%+ correction at any point during the year?
  (For this, track the monthly minimum across each path, not just the final value)
- Show the distribution of calendar-year returns (histogram of annual % change)
- Compare: How does current starting level compare to historical valuation?

Dashboard: yearly fan chart, annual return distribution, key level probabilities.
```

💡 **CUSTOMISE:** For year-end forecasts, track intermediate highs and lows — not just the final price. An investor cares if the portfolio drops 20% during the year, even if it recovers.

SECTION 7

Customisation Guide

How to adapt every prompt to your instrument and style

The [BRACKET] Variables — Full Cheat Sheet

BRACKET	WHAT TO REPLACE IT WITH
[INSTRUMENT]	NIFTY 50 · Bank Nifty · Reliance · TCS · S&P 500 · Bitcoin · EUR/USD
[PRICE]	Latest closing price. Get from NSE website, TradingView, or ask Claude to search.
[VOL%]	Annual historical volatility. Rule of thumb: Nifty ~20%, S&P 500 ~18%, BTC ~70%
[DRIFT%]	Expected annual return. Use 8-10% for large-cap indices, adjust for growth stocks.
[VIX VALUE]	India VIX for Indian markets. CBOE VIX for US. Use today's value for live accuracy.
[SIMULATIONS]	1,000 = fast preview. 10,000 = standard. 50,000 = high precision options pricing.
[DAYS]	30 = 1 month. 21 = 1 month of trading days. 63 = 1 quarter. 252 = 1 year.
[KEY LEVEL]	Round numbers: 25,000. 52-week high/low. Previous ATH. Strong support/resistance.
[THEME]	Dark Bloomberg / Cyberpunk Dark / Clean White / Print-Friendly / Neon Terminal
[CHART TYPE]	Plotly (interactive) / Matplotlib (static PNG) / Chart.js (web embed)

Style & Theme Guide

THEME	PROMPT PHRASE TO ADD
Dark Bloomberg	Use dark background #050A0F, cyan accents #00D4FF, monospace fonts (DM Mono), minimal borders
Cyberpunk Neon	Use black background, neon green and pink glows, scanline overlay, Orbitron or Share Tech Mono font
Clean White	Use white background, navy text, grey borders, professional sans-serif font, no glow effects
Print-Friendly	White background, all text black or dark grey, no Plotly (use Matplotlib static PNG instead)
Social Media Card	Fixed 1200×630px container, large numbers, bold typography, strong contrast, screenshot-ready

Weak vs Strong Prompt Examples

❌ WEAK	✅ STRONG
"Simulate the stock price"	"Run 10,000 GBM simulations for NIFTY 50 starting at ₹24,158 with daily sigma = 1.41% and daily mu = 0.036%, over 30 trading days"
"Make a nice chart"	"Plotly scatter3d: 500 paths, X=day, Y=sim index, Z=price. Colour green if final > start else red. Mean path in cyan width=5. Dark bg #030610"
"Add volatility"	"Use blended vol: sigma = 0.70*(India_VIX/100/sqrt(252)) + 0.30*(0.2014/sqrt(252)). Show both implied-only and blended results side by side"

💡 **PRO TIP**

The more specific your numbers, the better the output. Always provide: exact starting price, exact sigma and mu values, exact number of simulations, exact horizon in trading days, and exact chart specification including colours.

SECTION 8

Formula Reference

Every equation you need — copy into any prompt

Core Monte Carlo Formulas

∑ GBM — Daily Price Step

$P(t+1) = P(t) \times \exp((\mu - 0.5\sigma^2)\cdot 1 + \sigma\cdot Z)$ where $Z \sim N(0,1)$

∑ Log Return (always use this, not simple return)

$r_t = \ln(P_t / P_{t-1})$ → daily log return

∑ Parameter Estimation from Historical Data

$\mu_{daily} = \text{mean}(r_t)$ | $\sigma_{daily} = \text{std}(r_t)$

∑ Annualisation

$\sigma_{annual} = \sigma_{daily} \times \sqrt{252}$ | $\mu_{annual} = \mu_{daily} \times 252$

∑ Blended Volatility (VIX + Historical)

$\sigma_{blend} = 0.70 \times (VIX/100 \div \sqrt{252}) + 0.30 \times \sigma_{historical\_daily}$

∑ Value at Risk (95%)

$VaR_{95} = -\text{percentile}(final\_prices, 5\%)$ [expressed as loss from start]

∑ Conditional VaR / Expected Shortfall

$CVaR_{95} = -\text{mean}(final\_prices[final\_prices < VaR_{95}])$

∑ Sharpe Ratio

$\text{Sharpe} = (\text{Portfolio\_Return} - \text{Risk\_Free\_Rate}) \div \text{Portfolio\_StdDev}$

∑ Black-Scholes Call Price (for comparison vs MC)

$\text{Call} = S\cdot N(d1) - K\cdot e^{(-rT)}\cdot N(d2)$$d1 = [\ln(S/K) + (r + 0.5\sigma^2)T] \div (\sigma\sqrt{T})$$d2 = d1 - \sigma\sqrt{T}$

∑ Cholesky Decomposition (Correlated Returns)
L = cholesky(Σ) → Z_corr = L × Z_indepwhere Σ is the correlation matrix between assets

Monte Carlo Parameter Quick Reference

INSTRUMENT	TYPICAL PARAMETERS (approximate starting values)
NIFTY 50	Annual vol ~20%, Annual drift ~9-10%, VIX typically 12-18 (elevated: >20)
Bank Nifty	Annual vol ~25-28%, Higher beta to NIFTY, more volatile — use VIX × 1.3
S&P 500	Annual vol ~16-18%, Annual drift ~10%, VIX typically 15-20 (elevated: >25)
NASDAQ 100	Annual vol ~22-25%, Annual drift ~12-14%, Higher tech premium
Gold (XAUUSD)	Annual vol ~12-15%, Annual drift ~6-8%, Negative correlation to equities
Crude Oil (WTI)	Annual vol ~30-40%, High drift uncertainty — use shorter horizons
Bitcoin (BTC)	Annual vol ~60-80%, No reliable drift — use short horizons only
EUR/USD	Annual vol ~7-9%, Near-zero drift for FX pairs, use interest rate diff

You now have 20 prompts to build anything.

Start with Prompt 01. Build the basics. Then layer in 3D charts, live data, and portfolio features.

Share your dashboards, tag the source, and help your followers discover quantitative finance.

⚠ DISCLAIMER: Educational use only. Not financial advice. All investments carry risk.