From $\mathbb{P}$ to $\mathbb{Q}$: Girsanov vs NNS.rescale()

A production-ready alternative to change of measure

Fred Viole @OVVOLabs

November 21, 2025

One Rule to Price Them All

The Fundamental Pricing Identity

$$\boxed{\mathbb{E}^{\mathbb{Q}}[S_T] = S_0 e^{rT}}$$

Global Parameters Used: $S_0 = 100, r = 0.05, \mu = 0.10, \sigma = 0.2, T = 1$

Target forward: $S_0 e^{rT} = 105.1271$

- **Why?** Under $\mathbb{Q}$, the stock is a martingale after discounting.
- **Implication:** No arbitrage $\implies$ expected growth = risk-free rate.
- **Consequence:** All derivatives priced via risk-neutral expectation.
- **Key Insight:** You *do not need* μ to price — only r, σ, S_0.

The Two Worlds: $\mathbb{P}$ vs $\mathbb{Q}$

Real World $\mathbb{P}$

- Drift: $\mu = 0.1$ (investor belief)
- Volatility: $\sigma = 0.2$
- Use:
  - — Risk management
  - — VaR, stress testing
  - — P&L simulation
  - — Capital allocation

Risk-Neutral $\mathbb{Q}$

- Drift: $r = 0.05$ (by construction)
- Volatility: $\sigma = 0.2$ (unchanged)
- Use:
  - — Derivative pricing
  - — XVA (CVA, FVA)
  - — Hedging
  - — Model calibration

One model, two measures:

- $\mathbb{P}$: *What might happen*
- $\mathbb{Q}$: *What must be priced*

Discussion Point:
"Can you use $\mathbb{P}$-simulated paths to price? Yes — but only if you reweight or rescale to $\mathbb{Q}$."

Girsanov: Change of Measure

From Real Drift to Risk-Neutral Drift

$$\underbrace{dS_t = \mu S_t dt + \sigma S_t dW_t^{\mathbb{P}}}_{\text{Real world } \mathbb{P}} \xrightarrow{\text{Girsanov}} \underbrace{dS_t = r S_t dt + \sigma S_t dW_t^{\mathbb{Q}}}_{\text{Risk-neutral } \mathbb{Q}}$$

Radon-Nikodym derivative (weight):

$$\boxed{\frac{d\mathbb{Q}}{d\mathbb{P}} = \exp\left(-\lambda W_T^{\mathbb{P}} - \tfrac{1}{2}\lambda^2 T\right)}, \quad \lambda = \frac{\mu - r}{\sigma}$$

- **What is λ?** The market price of risk — how much extra return per unit volatility.
- **Intuition:** Paths with too much upside in $\mathbb{P}$ get downweighted in $\mathbb{Q}$.
- **Volatility unchanged:** σ is the same — only drift is adjusted.

Girsanov: Tiny Numeric (2 paths)

```r
ST_P <- c(95, 125)                          # Two simulated terminal prices under P
W_T <- (log(ST_P/S0) - (mu - 0.5*sigma^2)*T) / sigma  # Extract Brownian motion
lambda <- (mu - r)/sigma                    # Market price of risk
w <- exp(-lambda * W_T - 0.5*lambda^2*T)     # Radon-Nikodym weights
weighted.mean(ST_P, w)                       # Q-expectation of S_T

#> [1] 107.4521
```

Results:
Weights: 1.1421, 0.8104 → Weighted mean: 107.4521

S_T	Weight
95	1.1421
125	0.8104

- Path 95: upweighted
- Path 125: downweighted
- **Target:** $S_0 e^{rT} = 105.1271$

Key Takeaway: *"We don't change the paths — we change their importance."*

NNS.rescale: Direct Mean Enforcement

```r
dW <- rnorm(n, 0, sqrt(T))
ST_P <- S0 * exp((mu - 0.5*sigma^2)*T + sigma*dW)
ST_Q_nns <- NNS.rescale(ST_P, a = S0, b = r,
                        method = "riskneutral", T = T, type = "Terminal")
c(target = S0*exp(r*T), mean = mean(ST_Q_nns))

#>   target     mean
#> 105.1271 105.1271
```

One line: `NNS.rescale(P, ...)`

Monte Carlo: Shared Brownian Paths

```r
set.seed(1234); n <- 1e5; K <- 100
dW <- rnorm(n, 0, sqrt(T))
ST_Q_direct <- S0 * exp((r - 0.5*sigma^2)*T + sigma*dW)
ST_P <- S0 * exp((mu - 0.5*sigma^2)*T + sigma*dW)
```

Same $dW \rightarrow$ fair comparison

Monte Carlo: Pricing

```r
ST_Q_nns <- NNS.rescale(ST_P, a=S0, b=r, method="riskneutral",
                        T=T, type="Terminal")
W_T <- (log(ST_P/S0) - (mu - 0.5*sigma^2)*T)/sigma
lambda <- (mu-r)/sigma
w <- exp(-lambda*W_T - 0.5*lambda^2*T)
price_direct <- exp(-r*T)*mean(pmax(ST_Q_direct - K, 0))
price_nns <- exp(-r*T)*mean(pmax(ST_Q_nns - K, 0))
price_gir <- exp(-r*T)*weighted.mean(pmax(ST_P - K, 0), w)
ess_gir <- round((sum(w)^2) / sum(w^2))
c(direct = price_direct, nns = price_nns, gir = price_gir, ess = ess_gir)

#>      direct         nns         gir         ess
#>    10.48537    10.44869    10.49028 93962.00000
```

Results: Accuracy & Efficiency

	Direct $\mathbb{Q}$	NNS	Girsanov
Call price	10.4853712	10.4486858	10.4902804
Efficiency	100,000	100,000	$93,962$

Black-Scholes (analytic): 10.451

Verification

```r
cat(sprintf("Target: %.6f\n", S0*exp(r*T)))
#> Target: 105.127110
cat(sprintf("Direct Q: %.6f\n", mean(ST_Q_direct)))
#> Direct Q: 105.187609
cat(sprintf("NNS: %.6f\n", mean(ST_Q_nns)))
#> NNS: 105.127110
cat(sprintf("Girsanov: %.6f\n", weighted.mean(ST_P, w)))
#> Girsanov: 105.194594
```

All match target within MC error.

Constraint Families: From $\mathbb{P}$ to $\mathbb{Q}$

Two Valid Constraints

- **Terminal**: $\mathbb{E}[S_T^{\mathbb{Q}}] = S_0 e^{rT} \rightarrow$ Vanilla pricing
- **Discounted**: $\mathbb{E}[e^{-rt} S_t^{\mathbb{Q}}] = S_0 \rightarrow$ True martingale

Terminal at Grid Points

$\mathbb{E}[S_{t_k}^{\mathbb{Q}}] = S_0 e^{rt_k}$

$\rightarrow$ Valid for multi-maturity vanillas

$\rightarrow$ Not a martingale

Discounted at Grid Points

$\mathbb{E}[e^{-rt_k} S_{t_k}^{\mathbb{Q}}] = S_0$

$\rightarrow$ True discrete martingale

$\rightarrow$ Required for path-dependent

Dynamic Rescaling Options:

- `type = "Terminal"` at each $t_k \rightarrow$ correct forwards
- `type = "Discounted"` at each $t_k \rightarrow$ correct martingale

The NNS.rescale Mechanism: Distributional Transformation

From $\mathbb{P}$-Paths to $\mathbb{Q}$-Distribution

$$\boxed{S_T^{\mathbb{Q}} = e^{\theta} \cdot S_T^{\mathbb{P}}, \quad \theta = \log\left(\frac{S_0 e^{rT}}{\mathbb{E}^{\mathbb{P}}[S_T^{\mathbb{P}}]}\right)}$$

For GBM: Achieves Exact $\mathbb{Q}$-Distribution

$$S_T^{\mathbb{Q}} \overset{d}{=} S_0 \exp\left(\left(r - \frac{\sigma^2}{2}\right) T + \sigma W_T^{\mathbb{Q}}\right)$$

Why This Works

- **Distributionally Exact** for GBM
- **No-Arbitrage by Design**: $\mathbb{E}^{\mathbb{Q}}[S_T] = S_0 e^{rT}$
- **Empirical Convergence**: Sample mean $\to S_0 e^{\mu T}$
- **Zero Variance Loss**: Deterministic, full ESS

GBM Exactness: Why `NNS.rescale` Works Perfectly

Distributional Equivalence Theorem

Scaling $\mathbb{P}$-paths by $e^{(r-\mu)T}$ yields the exact $\mathbb{Q}$-distribution:

$$\boxed{S_T^{\mathbb{P}} \cdot e^{(r-\mu)T} \stackrel{d}{=} S_0 \exp\left(\left(r - \frac{\sigma^2}{2}\right) T + \sigma W_T^{\mathbb{Q}}\right)}$$

`NNS.rescale` Implements This

$$\theta \to (r-\mu)T \quad \text{as } n \to \infty \quad \Rightarrow \quad e^{\theta} \to e^{(r-\mu)T}$$

Key Insight

- **Empirical** $\rightarrow$ **Theoretical** scaling
- **Same $\mathbb{Q}$-distribution** without measure theory
- **Perfect pricing** up to MC error

Distributional Equivalence: Correct Proof

Theorem

For GBM:

$$\boxed{S_T^{\mathbb{P}} \cdot e^{(r-\mu)T} \stackrel{d}{=} S_T^{\mathbb{Q}}}$$

Proof via Distribution Matching

$$S_T^{\mathbb{P}} = S_0 \exp\left(\left(\mu - \frac{\sigma^2}{2}\right) T + \sigma W_T^{\mathbb{P}}\right)$$

$$S_T^{\mathbb{P}} \cdot e^{(r-\mu)T} = S_0 \exp\left(\left(r - \frac{\sigma^2}{2}\right) T + \sigma W_T^{\mathbb{P}}\right)$$

$$S_T^{\mathbb{Q}} = S_0 \exp\left(\left(r - \frac{\sigma^2}{2}\right) T + \sigma W_T^{\mathbb{Q}}\right)$$

Since $W_T^{\mathbb{P}} \stackrel{d}{=} W_T^{\mathbb{Q}} \sim \mathcal{N}(0, T)$, $\Rightarrow$ distributions are identical. □

GBM Exactness: Two Paths, One Distribution

Equivalent Constructions

Direct $\mathbb{Q}$-Simulation	**NNS.rescale**
$S_0 \exp\left(\left(r - \frac{\sigma^2}{2}\right)T + \sigma W_T^{\mathbb{Q}}\right)$	$S_T^{\mathbb{P}} \cdot e^{(r-\mu)T}$
Same distribution	Same distribution

The Truth

- **Distributional equivalence** under GBM
- **Empirical convergence** to correct scaling
- **Computational edge**: No re-simulation, full efficiency
- **Conceptual clarity**: Transform outcomes, not measures

Discrete-Grid Martingale via Dynamic Rescaling

Construction: At each t_k:

$$S^{\mathbb{Q}}_{t_k} \leftarrow \texttt{NNS.rescale}(\text{discounted } S, \text{type="Discounted"})$$

Enforces discounted ensemble mean $= S_0$.

```r
n_steps <- 100; dt <- T/n_steps; n_paths <- 10000
paths <- matrix(NA, n_steps+1, n_paths); paths[1,] <- S0
drift <- (r - 0.5*sigma^2)*dt; vol <- sigma*sqrt(dt)
for(i in 1:n_steps){
  inc <- rnorm(n_paths, drift, vol)
  next_p <- paths[i,] * exp(inc)
  t_i <- i*dt
  disc <- next_p * exp(-r*t_i)
  disc_rescaled <- NNS.rescale(disc, a=S0, b=r, method="riskneutral",
                               T=t_i, type="Discounted")
  paths[i+1,] <- disc_rescaled * exp(r*t_i)
}
disc_means <- rowMeans(exp(-r*seq(0,T,by=dt)) * paths)
c(head=disc_means[1], mid=disc_means[51], tail=disc_means[101])

#> head  mid tail
#>  100  100  100
```

Dynamic Rescaling: Ensemble Means

Time	Theoretical	Standard	Rescaled
0	100	100	100
0.25	101.2578	101.131	101.2578
0.5	102.5315	102.4051	102.5315
0.75	103.8212	103.8043	103.8212
1	105.1271	105.0349	105.1271

Dynamic Rescaling: Statistics

Metric	Value
Mean Volatility (Normal)	0.199539
Mean Volatility (Rescaled)	0.199529
Terminal Mean (Normal)	105.034938
Terminal Mean (Rescaled)	105.12711
Terminal Variance (Normal)	450.182022
Terminal Variance (Rescaled)	450.972466

Takeaway: Three Roads from $\mathbb{P}$ to $\mathbb{Q}$

All Roads Lead to the Same Price

Direct $\mathbb{Q}$	**Girsanov**	**NNS.rescale**
Simulate with drift r	Reweight $\mathbb{P}$-paths	Rescale $\mathbb{P}$-paths
Simple	*Elegant*	*Exact + Efficient*
MC noise: $\sigma/\sqrt{n}$	ESS $\downarrow$ as $\lvert\mu - r\rvert \uparrow$	**Full ESS, zero bias**

Mathematical Equivalence (GBM)

$$\boxed{\underbrace{S_T^{\mathbb{P}} \cdot e^{(r-\mu)T}}_{\text{Exact multiplier}} \overset{d}{=} \underbrace{S_0 \exp\left(\left(r - \frac{\sigma^2}{2}\right)T + \sigma W_T^{\mathbb{Q}}\right)}_{\text{Direct } \mathbb{Q}\text{-simulation}}}$$

NNS.rescale $\rightarrow$ empirically discovers $e^{(r-\mu)T}$

Decision Guide: Choose Your Weapon

Use `NNS.rescale` When...

- Vanilla pricing under GBM
- You want **maximum MC efficiency**
- Stability and speed matter (production)
- You already simulate under $\mathbb{P}$ (risk systems)

Use Dynamic Rescaling When...

- Path-dependent exotics (Asians, barriers)
- Multi-maturity calibration
- Need true discrete martingale: `type = "Discounted"`

Raw .Rnw File Download