Cat\_1: Computational Finance

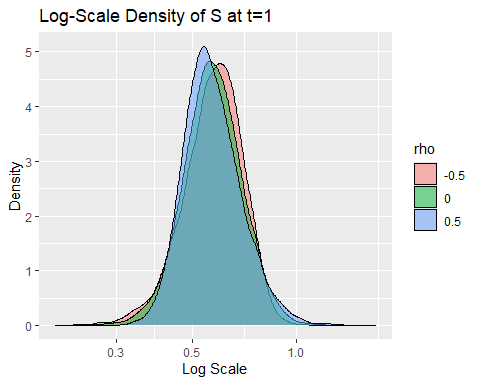
Joy Kendi

2025-09-10

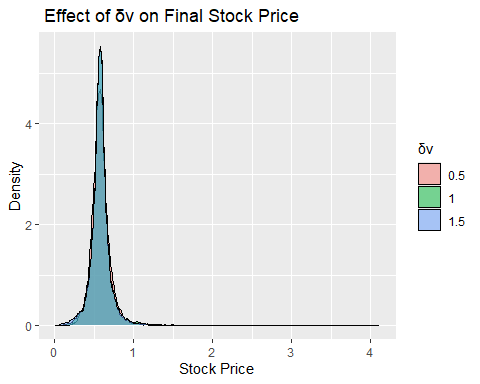
library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.4 ✔ readr 2.1.5  
## ✔ forcats 1.0.0 ✔ stringr 1.5.1  
## ✔ ggplot2 3.5.1 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.4 ✔ tidyr 1.3.1  
## ✔ purrr 1.0.4   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

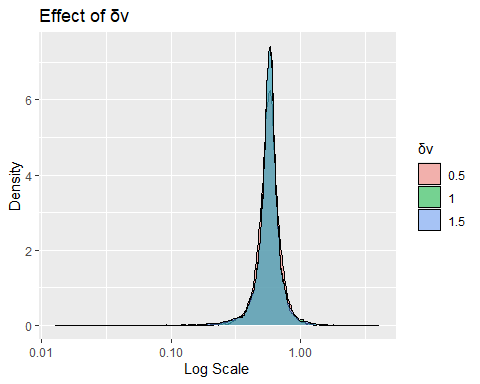
library(ggplot2)  
  
# Question 1  
mu <- 0.15  
V\_Bar <- 0.04  
delta\_v <- 0.2  
kappa\_v <- 1  
S0 <- 0.5  
V0 <- V\_Bar  
Time\_T <- 1  
Step\_Size <- 1000  
N\_Simulations <- 10000  
dt <- Time\_T/Step\_Size  
Corr <- c(-0.5, 0, 0.5)  
  
Sim\_SV <- function(rho, delta\_v, seed = 42){  
 set.seed(seed)  
   
 S <- matrix(NA\_real\_, nrow = N\_Simulations, ncol = Step\_Size +1 )  
 V <- matrix(NA\_real\_, nrow = N\_Simulations, ncol = Step\_Size +1 )  
   
 S[,1] <- S0  
 V[,1]<- V0  
   
 for (t in 1:Step\_Size) {  
 z1 <- rnorm(N\_Simulations)  
 z2 <- rho\* z1 + sqrt(1-rho^2) \* rnorm(N\_Simulations)  
   
 V\_T <- pmax(V[,t], 0)  
   
 V[, t+1] <- pmax(  
 V[, t] + kappa\_v \* (V\_Bar - V[, t]) \* dt +  
 delta\_v \* sqrt(V\_T) \* sqrt(dt) \* z2,  
 0  
 )  
  
 S[, t+1] <- S[, t] \* exp((mu - 0.5 \* V\_T) \* dt + sqrt(V\_T \* dt) \* z1)  
   
 }  
   
 tibble(  
 Final\_S = S[, Step\_Size + 1],  
 rho = factor(rho),  
 delta\_v = delta\_v  
 )  
}  
  
set.seed(123)  
  
RhoTest <- map\_dfr(Corr, ~Sim\_SV(rho = .x, delta\_v = 0.2))  
  
RhoTest |>  
 ggplot(aes(x = Final\_S, fill = rho))+  
 geom\_density(alpha = 0.5) +  
 scale\_x\_log10()+  
 labs(  
 title = "Log-Scale Density of S at t=1",  
 x = "Log Scale",  
 y = "Density"  
 )



Delta\_Array <- c(0.5, 1, 1.5)  
New\_RhoTest <- map\_dfr(Delta\_Array, ~Sim\_SV(rho = 0, delta\_v = .x))  
  
New\_RhoTest |>  
 ggplot(aes(x = Final\_S, fill = factor(delta\_v)))+  
 geom\_density(alpha = 0.5)+  
 labs(  
 title = " Effect of δv on Final Stock Price",  
 x = "Stock Price",  
 y = "Density",  
 fill = "δv"  
 )



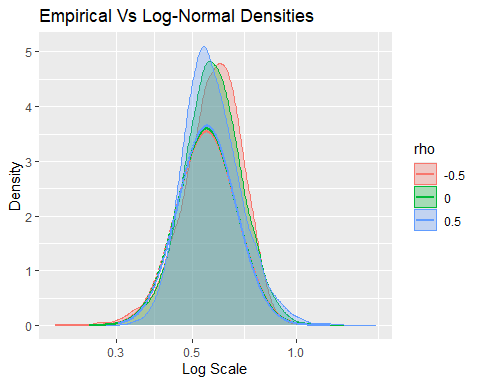
New\_RhoTest |>  
 ggplot(aes(x = Final\_S, fill = factor(delta\_v)))+  
 geom\_density( alpha = 0.5) +  
 scale\_x\_log10()+  
 labs(  
 title = "Effect of δv ",  
 x = "Log Scale",  
 y = "Density",  
 fill = "δv"  
 )



lognorm\_dist <- RhoTest |>  
 group\_by(rho) |>  
 summarise(  
 mu\_hat = mean(log(Final\_S)),  
 sigma\_hat = sd(log(Final\_S)),  
 .groups = "drop"  
 )  
  
lognorm\_dist

## # A tibble: 3 × 3  
## rho mu\_hat sigma\_hat  
## <fct> <dbl> <dbl>  
## 1 -0.5 -0.559 0.201  
## 2 0 -0.560 0.198  
## 3 0.5 -0.560 0.195

Difference <- RhoTest |>  
 group\_by(rho) |>  
 summarise(x = list(seq(min(Final\_S), quantile(Final\_S, 0.999), length.out = 500)))|>  
 unnest(x) |>  
 left\_join(lognorm\_dist, by = "rho") |>  
 mutate(dln = dlnorm(x, meanlog = mu\_hat, sdlog = sigma\_hat))  
   
ggplot(RhoTest, aes(x = Final\_S, color = rho)) +  
 geom\_density(aes(fill = rho), alpha = 0.3)+  
 geom\_line(data = Difference, aes(x=x, y = dln, color = rho),  
 linewidth = 1) +  
 scale\_x\_log10()+  
 labs(  
 title = "Empirical Vs Log-Normal Densities",  
 x="Log Scale",  
 y = "Density"  
 )



#Question 2  
# a) GBM Discretization  
S0 <- 100  
r <- 0.05  
sigma <- 0.2  
Time\_T <- 1  
N <- 50  
M <- 100000  
dt <- Time\_T/N  
Simm <- (1:N)\*dt  
  
set.seed(1)  
  
GBM <- function(M,N,dt,S0, r, sigma){  
 z <- matrix(rnorm(M\*N), nrow = M, ncol = N)  
 incr <- (r - 0.5\* sigma^2) \* dt + sigma \* sqrt(dt) \* z  
 log\_s <- cbind(log(S0), t(apply(incr, 1,cumsum)))  
 S <- exp(log\_s)  
 S  
}  
  
paths <- GBM(M,N,dt,S0,r,sigma)  
S\_grid <- paths[, -1, drop = FALSE]  
S\_T <- S\_grid[, N]  
S\_Bar <- rowMeans(S\_grid)  
  
payoff\_x <- pmax(S\_T - S\_Bar, 0)  
discrete\_x <- exp(-r \* Time\_T) \* payoff\_x  
  
Cmc <- mean(discrete\_x)  
SE\_Cmc <- sd(discrete\_x)/sqrt(M) #Standard Error  
CI\_Cmc <- Cmc + c(-1,1) \* 1.96 \* SE\_Cmc # Confidence Interval  
  
list(  
 CMC = Cmc,  
 CI95 = CI\_Cmc  
)

## $CMC  
## [1] 0.05777076  
##   
## $CI95  
## [1] 0.05725176 0.05828976

# b) Control Variate  
Asian\_Call <- function(S0, r, sigma, Time\_T, N){  
 ti <- (1:N)\*(Time\_T/N)  
 Summ\_t <- sum(ti)  
 Summ\_min <- sum(pmin(rep(ti, each =N), rep(ti, times = N)))  
   
 M\_G <- log(S0) + (r - 0.5 \* sigma^2) \* (Summ\_t/N)  
 V\_G <- (sigma^2 / N^2) \* Summ\_min  
   
 Var\_ST <- sigma^2 \* Time\_T  
 Cov <- (sigma^2/N) \* Summ\_t  
 Var\_ex <- Var\_ST + V\_G -2 \* Cov  
 sigma\_ex <- sqrt(Var\_ex/Time\_T)  
   
 S1 <- S0  
 S2 <- exp(M\_G + 0.5 \* V\_G - r \* Time\_T)  
   
 d1 <- (log(S1/S2)+0.5 \* sigma\_ex^2 \* Time\_T)/(sigma\_ex\* sqrt(Time\_T))  
 d2 <- d1 - sigma\_ex \* sqrt(Time\_T)  
   
 mu\_Y <- S1 \* pnorm(d1) - S2 \* pnorm(d2)  
 mu\_Y  
}  
  
mu\_Y <- Asian\_Call(S0, r, sigma, Time\_T, N)  
Geo\_S <- exp(rowMeans(log(S\_grid)))  
payoff\_y <- pmax(S\_T - Geo\_S, 0)  
discrete\_y <- exp(-r \* Time\_T) \* payoff\_y  
  
theta\_hat <-cov(discrete\_x, discrete\_y) / var(discrete\_y)  
z <- discrete\_x+ theta\_hat \* (mu\_Y - discrete\_y)  
  
Ccv <- mean(z)  
SE\_Ccv <- sd(z)/sqrt(M)  
CI\_Ccv <- Ccv + c(-1,1) \* 1.96 \* SE\_Ccv  
  
list(  
 CCV = Ccv,  
 CI95\_Ccv = CI\_Ccv,  
 theta\_hat = theta\_hat,  
 mu\_Y = mu\_Y  
)

## $CCV  
## [1] 5.784387  
##   
## $CI95\_Ccv  
## [1] 5.784374 5.784400  
##   
## $theta\_hat  
## [1] 0.9675969  
##   
## $mu\_Y  
## [1] 5.978259

# c)  
Var\_MC <- var(discrete\_x)  
Var\_CV <- var(z)  
VR <- 100\*(1 - Var\_CV/Var\_MC)  
  
tibble(  
 estimator = c("Plain MC", "With Control Variate"),  
 price = c(Cmc, Ccv),  
 SE = c(SE\_Cmc, SE\_Ccv),  
 CI\_Low = c(CI\_Cmc[1], CI\_Ccv[1]),  
 CI\_High = c(CI\_Cmc[2], CI\_Ccv[2])  
) |>  
 print()

## # A tibble: 2 × 5  
## estimator price SE CI\_Low CI\_High  
## <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 Plain MC 0.0578 0.000265 0.0573 0.0583  
## 2 With Control Variate 5.78 0.00000651 5.78 5.78

cat(sprintf("Variance Reduction: %.1f%%\n", VR)) # Variance Reduction: 99%, 0.1% of the variance is uncorrelated to the plain MC

## Variance Reduction: 99.9%

# d) CI   
CV\_Price <- function(M,N,sigma, seed=1){  
 set.seed(seed)  
 dt<- Time\_T/N  
 S\_Paths <- GBM(M,N,dt,S0,r,sigma)  
 S\_grid <- S\_Paths[,-1,drop = FALSE]  
 S\_T <- S\_grid[,N]  
 S\_Bar <- rowMeans(S\_grid)  
 X<- exp(-r \* Time\_T) \* pmax(S\_T -S\_Bar,0)  
   
 G <- exp(rowMeans(log(S\_grid)))  
 Y <- exp(-r \*Time\_T) \* pmax(S\_T - G, 0)  
 MuY <- Asian\_Call(S0, r, sigma, Time\_T, N)  
   
 theta <- cov(X,Y)/var(Y)  
 z <- X + theta \* (MuY - Y)  
   
 c(MC = mean(X), SE\_MC = sd(X)/sqrt(M),  
 CV = mean(z), SE\_Cv = sd(z)/sqrt(M)  
 )  
}  
  
Ms <- round(exp(seq(log(1e3), log(1e5), length.out = 6)))  
res\_m <- map\_dfr(Ms, \(m) {  
 out <- CV\_Price(M = m, N = N, sigma = sigma, seed = 123)  
 tibble(  
 M = m,  
 mc\_low = out["MC"] - 1.96 \* out["SE\_MC"],  
 mc\_high = out["MC"] + 1.96 \* out["SE\_MC"],  
 cv\_low = out["CV"] - 1.96 \* out["SE\_Cv"],  
 cv\_high = out["CV"] + 1.96 \* out["SE\_Cv"]  
 )  
})  
  
res\_m |>  
 mutate(  
 width\_mc = mc\_high - mc\_low,  
 width\_cv = cv\_high - cv\_low  
 ) |>  
 pivot\_longer(starts\_with("width\_"), names\_to = "method", values\_to = "width") |>  
 mutate(method = recode(method, width\_mc = "MC", width\_cv = "CV")) |>  
 ggplot(aes(x = M, y = width, color = method)) +  
 geom\_line() + geom\_point() +  
 scale\_x\_log10() +  
 labs(title = "95% CI width vs simulations M",  
 x = "log scale", y = "CI width") +  
 theme\_minimal()

