Finance Quantitative

"Exo: Formule de Breeden-Litzenberger"
Solution

Version: 14 mars 2024

On se propose de calculer la distribution empirique de S_T à partir de la volatilité implicite des options.

```
sigma <- .2
S <- 100
r <- .0
b <- 0.0
T <- 1
```

La courbe de volatilité est donnée par un polynome du second degré. La volatilité de "Black-Scholes" est la volatilité à l'argent, réputée indépendante du strike.

```
## quadratic smile coefficients
a1 <- -.80/10
a2 <- 1/10

## BS volatility function
bsVol <- function(K) {
  rep(sigma, length(K))
}

## Volatility with smile
smileVol <- function(K) {
  sigma + a1*log(K/S) + a2*log(K/S)^2
}
```

Densité de S_T

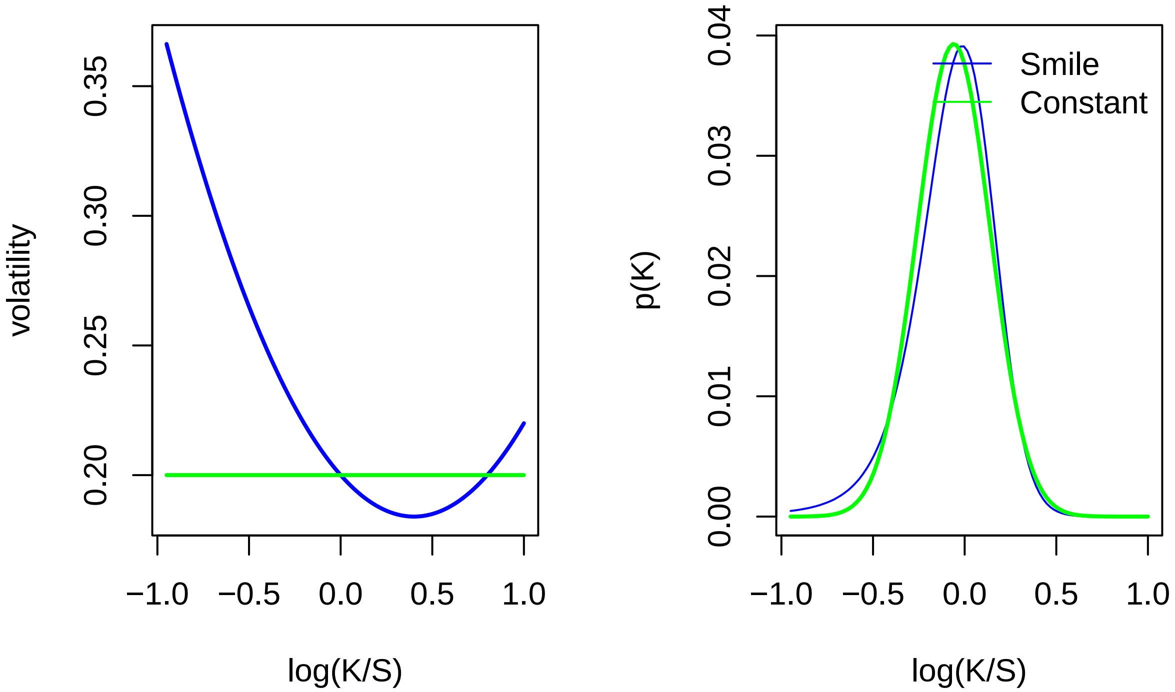
Calculer la densité $p(S_T)$ en utilisant la formule de Breeden-Litzenberger.

On calcule la dérivée seconde par différence finie, et on normalise les probabilités discrètes obtenues.

```
d2CdK2 <- function(vol, S, K, T, r, b) {
 dK <- 1.e-4
 c <- GBSOption('c', S, K, T, r, b, vol(K))@price
 cPlus <- GBSOption('c', S, K+dK, T, r, b, vol(K+dK))@price
 cMinus <- GBSOption('c', S, K-dK, T, r, b, vol(K-dK))@price
 (cPlus-2*c+cMinus)/(dK^2)
}

Ret <- seq(-.95, 1.00, length.out=100)
KRange <- S*exp(Ret*T)
nb <- length(KRange)
p <- matrix(nrow=nb, ncol=2)

i <- 1
```

```r
for(K in KRange) {
  p[i,1] <- d2CdK2(bsVol, S, K, T, r, b) * exp(r*T)
  p[i,2] <- d2CdK2(smileVol, S, K, T, r, b) * exp(r*T)
  i<- i+1
}
p[,1] = p[,1] / sum(p[,1])
p[,2] = p[,2] / sum(p[,2])
```

```r
par(mfrow=c(1,2))
plot(log(KRange/S), smileVol(KRange), xlab='log(K/S)', ylab='volatility', type='l', col='blue',
     lwd=2)
lines(log(KRange/S), bsVol(KRange), type='l', col='green', lwd=2)

plot(log(KRange/S), p[,2], type='l', xlab='log(K/S)', ylab='p(K)', col='blue', ylim=c(0, max(p)))
lines(log(KRange/S), y=p[,1], type='l', lwd=2, col='green')
legend('topright', c("Smile", "Constant"), lty=c(1,1), col=c('blue', 'green'), bty="n")
```




```r
par(mfrow=c(1,1))
```

Valorisation de call digitaux strike=105

On procède par intégration numérique de la fonction de densité.

```r
bs.pdf <- function(K) {
  d2CdK2(bsVol, S, K, T, r, b) * exp(r*T)
}

smile.pdf <- function(K) {
  d2CdK2(smileVol, S, K, T, r, b) * exp(r*T)
}

K <- 105
```

```r
d2 <- (log(S/K) + (r-sigma**2/2)*T)/(sigma*sqrt(T))
C.digital.bs <- exp(-r*T) * pnorm(d2)

smile.sigma <- smileVol(K)
d2 <- (log(S/K) + (r-smile.sigma**2/2)*T)/(smile.sigma*sqrt(T))
C.digital.smile <- exp(-r*T) * pnorm(d2)

# verify integration
Fwd.BS <- exp(-r*T) * integrate(Vectorize(bs.pdf), lower=10, upper=500, subdivisions = 5000,
                                rel.tol=1.e-2, abs.tol=1.e-3)$value

Call.BS <- exp(-r*T) * integrate(Vectorize(bs.pdf), lower=K, upper=500, subdivisions = 5000,
                                 rel.tol=1.e-2, abs.tol=1.e-3)$value
Call.Smile <- exp(-r*T) * integrate(Vectorize(smile.pdf), lower=K, upper=500, subdivisions = 5000,
                                    rel.tol=1.e-2, abs.tol=1.e-3)$value



print(paste("K: ", K,
"Prix (vol BS): ", round(Call.BS,3), "Prix (smile): ", round(Call.Smile,3)))
```

```
## [1] "K:  105 Prix (vol BS):  0.366 Prix (smile):  0.391"
```