Options dans le cadre Black-Scholes

TP-2: Pricing Vanna-Volga

Version: 22 mar 2022

The purpose of this problem set is to explore the Vanna-Volga pricing model. In this problem set, you will use the following functions:

GBSPrice: Price of a vanilla option:

$$P = f(\text{PutCall}, S, K, T, r, b, \sigma)$$

where:

PutCall 'c' for a call, 'p' for a put

b cost of carry: ridk free rate r less dividend yield d

r risk-free rate

```
GBSPrice <- function(PutCall, S, K, T, r, b, sigma) {
  d1 <- (log(S/K) + (b+sigma^2/2)*T)/(sigma*sqrt(T))
  d2 <- d1 - sigma*sqrt(T)

  if(PutCall == 'c')
    px <- S*exp((b-r)*T)*pnorm(d1) - K*exp(-r*T)*pnorm(d2)
  else
    px <- K*exp(-r*T)*pnorm(-d2) - S*exp((b-r)*T)*pnorm(-d1)

px
}
```

GBSVega: Vega ($\frac{\partial P}{\partial \sigma}$) of a Vanilla option:

```
GBSVega <- function(PutCall, S, K, T, r, b, sigma) {
  d1 <- (log(S/K) + (b+sigma^2/2)*T)/(sigma*sqrt(T))
  S*exp((b-r)*T) * dnorm(d1)
}
```

Volatility Interpolation

Given the implied volatility at three strikes, we will use the Vanna-Volga pricing method to interpolate the volatility curve. Assume $r = 0, b = 0, T = 1, \text{Spot} = 100$.

```r
# Benchmark data: (strike, volatility)
VolData <- list(c(80, .32), c(100, .30), c(120, .315))
```

Let's first define an array of pricing functions for the benchmark instruments:

```r
C1 <- function(vol=sigma, spot=Spot) GBSPrice(PutCall='c', S=spot, K=VolData[[1]][1], T=T, r=r, b=b, si

C2 <- function(vol=sigma, spot=Spot) GBSPrice(PutCall='c', S=spot, K=VolData[[2]][1], T=T, r=r, b=b, si

C3 <- function(vol=sigma, spot=Spot) GBSPrice(PutCall='c', S=spot, K=VolData[[3]][1], T=T, r=r, b=b, si

C <- c(C1, C2, C3)
```

1. Write a utility functions to compute the risk indicators, all by finite difference:

```r
Vega <- function(f, vol, spot=Spot) {
  d_vol <- 10e-5
  return( (f(vol+d_vol*vol, Spot)-f(vol-d_vol*vol, Spot))/(2*d_vol) )
}

Vanna <- function(f, vol, spot=Spot) {
  d_vol <- 10e-5
  d_spot <- 10e-5
  return( (f(vol+d_vol*vol, Spot+d_spot*Spot)+
           f(vol-d_vol*vol, Spot-d_spot*Spot)-
           f(vol+d_vol*vol, Spot-d_spot*Spot)-
           f(vol-d_vol*vol, Spot+d_spot*Spot))/(4*d_vol*d_spot) )
}

Volga <- function(f, vol, spot=Spot) {
  d_vol <- 10e-5
  return( (f(vol+d_vol*vol, Spot) - 2*f(vol, Spot) + f(vol-d_vol*vol, Spot))/d_vol**2 )
  }
```

Then, the calculation of vega for the three benchmark options may be performed by:

```r
r<-0
b<-0
T<-1
Spot <- 100
B.vega <- sapply(1:3, function(i) Vega(C[[i]], VolData[[i]][2]))
```

2. Compute vectors of vega, vanna, volga for the three hedge instruments

```r
r<-0
b<-0
T<-1
Spot <- 100
B.vega.benchmark <- sapply(1:3, function(i) Vega(C[[i]], VolData[[i]][2]))
B.vanna.benchmark <- sapply(1:3, function(i) Vanna(C[[i]], VolData[[i]][2]))
B.volga.benchmark <- sapply(1:3, function(i) Volga(C[[i]], VolData[[i]][2]))
print("B.vega.benchmark for C1, C2 and C3")
```

```
## [1] "B.vega.benchmark for C1, C2 and C3"
```

```
B.vega.benchmark
```

```
## [1]  8.840076 11.834380 11.499491
```

```
print("B.vanna.benchmark for C1, C2 and C3")
```

```
## [1] "B.vanna.benchmark for C1, C2 and C3"
```

```
B.vanna.benchmark
```

```
## [1] -14.84369   5.91719  26.87955
```

```
print("B.volga.benchmark for C1, C2 and C3")
```

```
## [1] "B.volga.benchmark for C1, C2 and C3"
```

```
B.volga.benchmark
```

```
## [1]  4.0722703 -0.2662745  3.5671601
```

3. Choose a new strike for which we want to compute the implied volatility. Let's choose $K = 110$.
4. Compute the risk indicators for a call option struck at that strike.

```
r<-0
b<-0
T<-1
Spot <- 100
K <- 110


VolData.ATM <- VolData[[2]][2]

f <- function(vol=VolData.ATM, spot=Spot){
  GBSPrice(PutCall='c', S=spot, K=K, T=T, r=r, b=b, sigma=vol)
}
B.vega <- Vega(f, VolData.ATM, Spot)
B.vanna <- Vanna(f, VolData.ATM, Spot)
B.volga <- Volga(f, VolData.ATM, Spot)
print("vega for K=110")
```

```
## [1] "vega for K=110"
```

```
B.vega
```

```
## [1] 11.80115
```

```r
print("vanna for K=110")
```

```
## [1] "vanna for K=110"
```

```r
B.vanna
```

```
## [1] 18.39802
```

```r
print("volga for K=110")
```

```
## [1] "volga for K=110"
```

```r
B.volga
```

```
## [1] 0.9256084
```

```r
b.risk <- c(B.vega, B.vanna, B.volga)
```

5. Compute the Vanna-Volga price adjustment and the corresponding implied volatility.

```r
A <- matrix(data = c(B.vega.benchmark, B.vanna.benchmark, B.volga.benchmark), nrow =3)
A <- t(A)
X <- solve(A, b.risk)
print("Matrice A =")
```

```
## [1] "Matrice A ="
```

```r
print(A)
```

```
##            [,1]       [,2]     [,3]
## [1,]   8.840076 11.8343799 11.49949
## [2,] -14.843691  5.9171899 26.87955
## [3,]   4.072270 -0.2662745  3.56716
```

```r
print("Risk indicators (b) = ")
```

```
## [1] "Risk indicators (b) = "
```

```r
print(b.risk)
```

```
## [1] 11.8011514 18.3980180  0.9256084
```

```r
print("Weights =")
```

```
## [1] "Weights ="
```

```r
print(X)
```

```
## [1] -0.1381243  0.6480079  0.4655345
```

```r
vol.K <- VolData[[1]][2]*X[1] + VolData[[2]][2]*X[2] + VolData[[3]][2]*X[3]
print("vol de K = 110")
```

```
## [1] "vol de K = 110"
```

```r
print(vol.K)
```

```
## [1] 0.2968459
```

```r
C1.M <- GBSPrice(PutCall='c', S=100, K=80, T=T, r=r, b=b, sigma=VolData.ATM)

C2.M <- GBSPrice(PutCall='c', S=100, K=100, T=T, r=r, b=b, sigma=VolData.ATM)

C3.M <- GBSPrice(PutCall='c', S=100, K=120, T=T, r=r, b=b, sigma=VolData.ATM)

C.M <- c(C1.M, C2.M, C3.M)

C1.BS <- GBSPrice(PutCall='c', S=100, K=80, T=T, r=r, b=b, sigma=VolData.ATM)

C2.BS <- GBSPrice(PutCall='c', S=100, K=100, T=T, r=r, b=b, sigma=VolData.ATM)

C3.BS <- GBSPrice(PutCall='c', S=100, K=120, T=T, r=r, b=b, sigma=VolData.ATM)

C.BS <- c(C1.BS, C2.BS, C3.BS)

O.BS <- GBSPrice(PutCall='c', S=100, K=110, T=T, r=r, b=b, sigma=VolData[[2]][2])


somme <- 0
for (i in 1:3) {
  somme <- somme + X[i]*(C.M[i]-C.BS[i])
}
O.M <- O.BS + somme
print("Price with BS")
```

```
## [1] "Price with BS"
```

```r
print(O.BS)
```

```
## [1] 8.141012
```

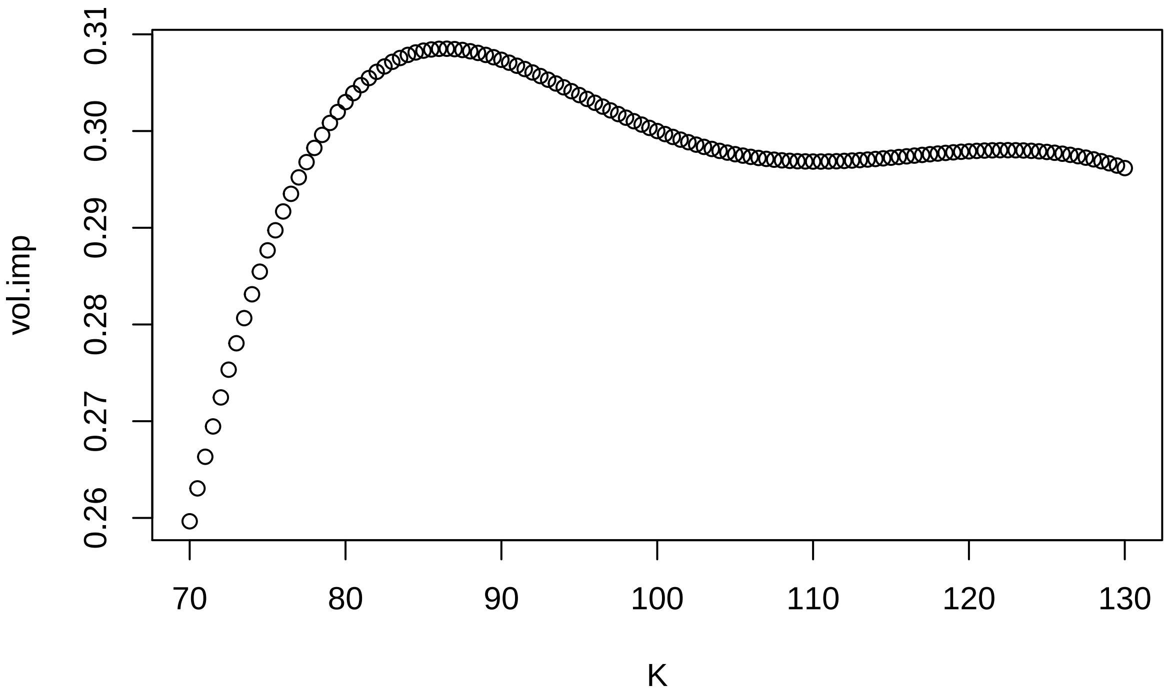
```r
print("Price ajusted")
```

```
## [1] "Price ajusted"
```

```
print(O.M)
```

```
## [1] 8.141012
```

6. Wrap the above logic in a function in order to interpolate/extrapolate the vol curve from $K = 70$ to $K = 130$



Pricing a digital call

Recall that a digital call with strike K pays one euro if $S_T \geq K$, and nothing otherwise.

Using the same logic as in the previous question, price a digital call, maturity $T = 1$, struck at $K = 105$.

```
r<-0
b<-0
T<-1
Spot <- 100
K <- 105

epsi <- 0.0001


VolData.ATM <- VolData[[2]][2]

f <- function(vol=VolData.ATM, spot=Spot){
  GBSPrice(PutCall='c', S=spot, K=K-epsi, T=T, r=r, b=b, sigma=vol)
}
B.vega <- Vega(f, VolData.ATM, Spot)
B.vanna <- Vanna(f, VolData.ATM, Spot)
```

```r
B.volga <- Volga(f, VolData.ATM, Spot)
print("vega for K=105 - espilon")
```

```
## [1] "vega for K=105 - espilon"
```

```r
B.vega
```

```
## [1] 11.96731
```

```r
print("vanna for K=105 - espilon")
```

```
## [1] "vanna for K=105 - espilon"
```

```r
B.vanna
```

```
## [1] 12.47117
```

```r
print("volga for K=105 - espilon")
```

```
## [1] "volga for K=105 - espilon"
```

```r
B.volga
```

```
## [1] 0.04725749
```

```r
b.risk.moins <- c(B.vega, B.vanna, B.volga)
X.moins <- solve(A, b.risk.moins)

O.BS.moins <- GBSPrice(PutCall='c', S=100, K=K-epsi, T=T, r=r, b=b, sigma=VolData[[2]][2])


somme <- 0
for (i in 1:3) {
  somme <- somme + X.moins[i]*(C.M[i]-C.BS[i])
}
O.M.moins <- O.BS.moins + somme

print("price with BS for K=105 - espilon")
```

```
## [1] "price with BS for K=105 - espilon"
```

```r
print(O.BS.moins)
```

```
## [1] 9.881693
```

```r
print("price ajusted for K=105 - espilon")
```

```
## [1] "price ajusted for K=105 - espilon"
```

```r
print(O.M.moins)
```

```
## [1] 9.881693

## [1] "vega for K=105 - espilon"

## [1] 11.96731

## [1] "vanna for K=105 - espilon"

## [1] 12.47142

## [1] "volga for K=105 - espilon"

## [1] 0.04727951

## [1] "price with BS for K=105 + espilon"

## [1] 9.881618

## [1] "price ajusted for K=105 + espilon"

## [1] 9.881618

## [1] "price for a digital call paying 2*epsi"

## [1] 7.545588e-05

## [1] "price for a digital call paying 1"

## [1] 0.3772794
```