Option Pricing and Estimation of Financial Models with R

6.2.3 Option pricing with R

```
R> call.price <- function(x = 1, t = 0, T = 1, r = 1, sigma = 1,
+      K = 1) {
+      d2 <- (log(x/K) + (r - 0.5 * sigma^2) * (T - t))/(sigma *
+          sqrt(T - t))
+      d1 <- d2 + sigma * sqrt(T - t)
+      x * pnorm(d1) - K * exp(-r * (T - t)) * pnorm(d2)
+ }
```

```
R> put.price <- function(x = 1, t = 0, T = 1, r = 1, sigma = 1,
+     K = 1) {
+     d2 <- (log(x/K) + (r - 0.5 * sigma^2) * (T - t))/(sigma *
+         sqrt(T - t))
+     d1 <- d2 + sigma * sqrt(T - t)
+     K * exp(-r * (T - t)) * pnorm(-d2) - x * pnorm(-d1)
+ }
```

We can now calculate the price of a constract with $S_0 = 100$, strike price $K = 110$, interest rate $r = 0.05$ with maturity 3 months. In this case $T = 1/4$, i.e. one fourth of the year, if we consider daily data. We assume a volatility of $\sigma = 0.25$.

```
R> S0 <- 100
R> K <- 110
R> r <- 0.05
R> T <- 1/4
R> sigma <- 0.25
R> C <- call.price(x = S0, t = 0, T = T, r = r, K = K,
   sigma = sigma)
R> C

[1] 1.980506
```

and for the price of the put

```
R> P <- put.price(x = S0, t = 0, T = T, r = r, K = K,
   sigma = sigma)
R> P
```

```
[1] 10.61406
```

and check the put-call parity formula

```
R> C - S0 + K * exp(-r * T)
```

```
[1] 10.61406
```

Another solution is to use the **fOptions** package from the **Rmetrics** suite (see Appendix B.1.1). We have to use the function `GBSOption` from the **fOptions** which calculates several exact formulas for options of the Generalized Black and Scholes model

```
R> require(fOptions)
```

For the call option we use the call price we need to write

```
R> GBSOption(TypeFlag = "c", S = S0, X = K, Time = T, r = r,
   b = r,
+      sigma = sigma)
```

```
Title:
 Black Scholes Option Valuation

Call:
 GBSOption(TypeFlag = "c", S = S0, X = K, Time = T, r = r, b = r,
     sigma = sigma)

Parameters:
          Value:
 TypeFlag c

 S        100
 X        110
 Time     0.25
 r        0.05
 b        0.05
 sigma    0.25

Option Price:
 1.980509
```

Notice that the function produces extensive output. If we just want a numeric value we need to access the slot `price` as follows:

```
R> GBSOption(TypeFlag = "c", S = S0, X = K, Time = T, r = r,
   b = r,
+      sigma = sigma)@price

[1] 1.980509
```

Note further that the generalized Black and Scholes formula includes an additional parameter `b` which is the *cost of carry*. In order to obtain the standard formulas one has to put `b = r`, as in our examples. For the put options, we need to change the argument `TypeFlag` from `c` to `p`

```
R> GBSOption(TypeFlag = "p", S = S0, X = K, Time = T, r = r,
   b = r,
+     sigma = sigma)@price

[1] 10.61407
```

6.6 Implied volatility and volatility smiles

If we look at the market price of e.g., a call option at a given time instant, we can compare it with the price predicted by the Black and Scholes formula. Let us denote by p the price observed on the market. Now, consider the Black and Scholes price of a call option at time $t = 0$

$$p_0 = S_0 \Phi(d_1) - e^{-rT} K \Phi(d_2)$$

with

$$d_1 = d_2 + \sigma\sqrt{T}, \qquad d_2 = \frac{\ln \frac{S_0}{K} + \left(r - \frac{1}{2}\sigma^2\right) T}{\sigma\sqrt{T}}.$$

Given the strike price K, the time to maturity T, the interest rate r, the current price of the asset S_0 and its volatility σ, we are able to calculate the predicted price p_0 by the above formula. We can compare this price p_0 with the market price p. The only delicate matter is which value of σ we should plug in the formula. One should think at taking the historical volatility estimated on the log-returns (see Section 5.1.2).

```
R> require(fImport)
R> S <- yahooSeries("ATL.MI", from = "2004-07-23",
   to = "2005-05-13")
R> head(S)

GMT
           ATL.MI.Open ATL.MI.High ATL.MI.Low ATL.MI.Close ATL.MI.Volume
2005-05-13       20.82       20.87      20.50        20.55       5944700
2005-05-12       20.88       21.10      20.63        20.77       3324700
2005-05-11       20.81       21.01      20.66        20.86       7415700
2005-05-10       20.65       20.98      20.61        20.80       2357700
2005-05-09       20.40       20.66      20.23        20.60       4171500
2005-05-06       20.30       20.68      20.08        20.50       3038800
           ATL.MI.Adj.Close
2005-05-13            16.96
2005-05-12            17.14
2005-05-11            17.22
2005-05-10            17.17
2005-05-09            17.00
2005-05-06            16.92

R> Close <- S[, "ATL.MI.Close"]
```

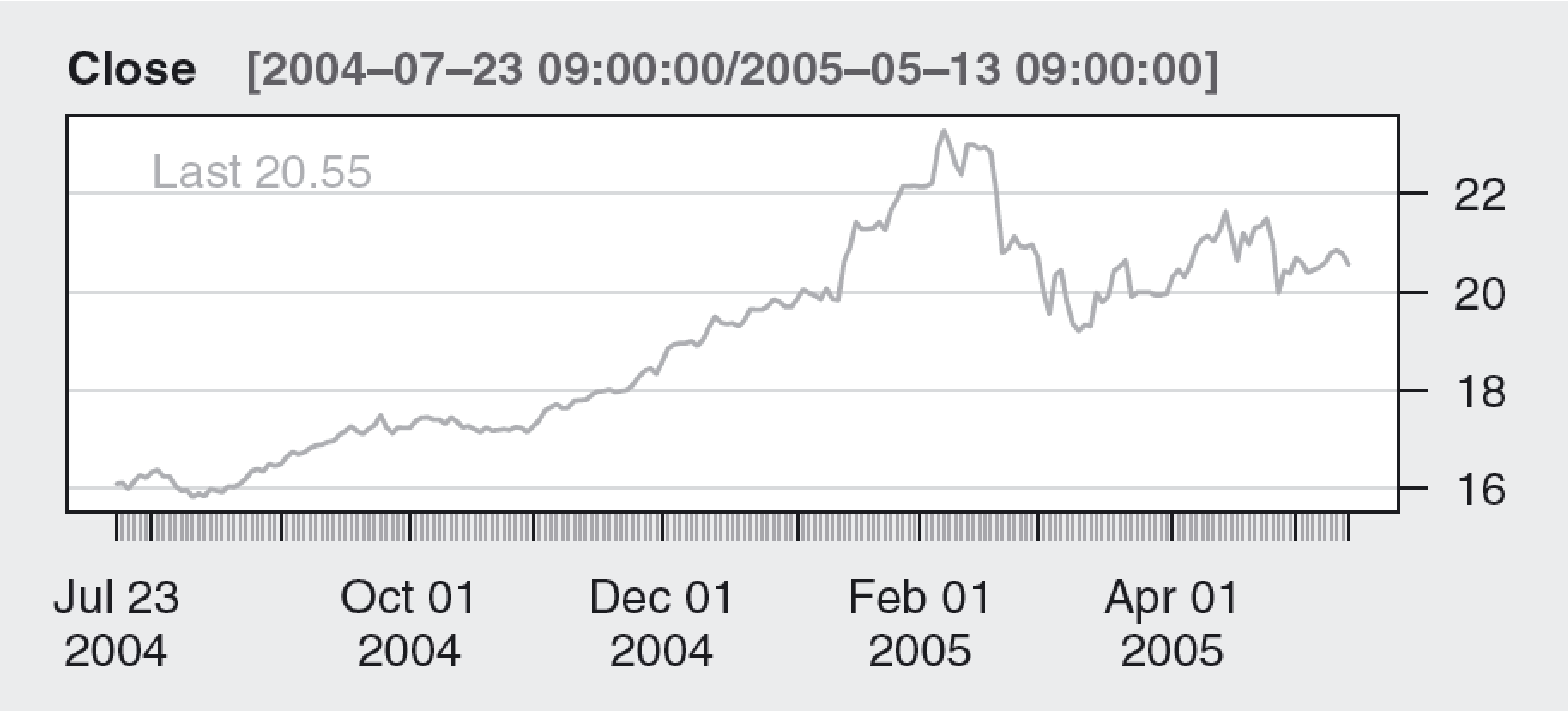

Figure 6.8 Close values of the ATL.MI asset.

```
R> require(quantmod)
R> chartSeries(Close, theme = "white")
```

We now calculate the variance of the log returns in order to obtain the historical volatility setting $\Delta = 1/252$ because we use daily data

```
R> X <- returns(Close)
R> Delta <- 1/252
R> sigma.hat <- sqrt(var(X)/Delta)[1, 1]
R> sigma.hat

[1] 0.1933289
```

We have used the R function `returns` from the **timeSeries** package. In order to use the Black and Scholes formula we need to identify all quantities. We consider a call option priced on 13 May 2005. The market price was $p = 0.0004$, the strike price $K = 23$, $S_0 = 20.55$. The expiry date was 3 June 2005 which corresponds to 15 days, thus we set $T = 15 \cdot \Delta$. The annual interest rate was $r = 0.02074$. On 13 May 2005, ATL.MI call option was priced

```
R> S0 <- Close[1]
R> K <- 23
R> T <- 15 * Delta
R> r <- 0.02074
R> sigma.hat <- as.numeric(sigma.hat)
R> require(fOptions)
R> p0 <- GBSOption("c", S = S0, X = K, Time = T, r = r, b = r,
   sigma = sigma.hat)@price
R> p0

[1] 0.003125474
```

One approach you can try that works is to go to
http://www.cboe.com/DelayedQuote/QuoteTableDownload.aspx

download the ticker chain you want and then read the downloaded file into R via:
tckr<-read.csv("C:/QuoteData.dat",skip=2)

This will read the chain into a data frame.

www.cboe.com/delayedquote/quote-table-download

Cboe ▸ Quotes ▸ Delayed Quotes

Quote Table Download

Search by Symbol

Download a list of all options and LEAPS for the underlying in comma delimited text format

Enter Stock or Index Symbols

Download Reset

Get free real-time spread order pricing with COBWeb[SM] - Cboe's Complex Order Book

Suggestions for Importing this data to Microsoft Excel

Some versions of Excel offer a built-in default formatting mechanism which can corrupt the raw data. These versions may express fractions as dates. Once downloaded, the conversion is not reversible.

We have devised a solution for preserving the price data. It saves the imported quotes as text, not numbers. The principal drawback is possible limitations to manipulating the data further within the spreadsheet program.

1. Enter the stock or index symbol in the dialog box. Click "Download."
2. Rename and save the file (QuoteTable.dat) to your hard drive.
3. Open Excel.
4. Click "File" - "Open".
5. Change "Files of Type" area to "All files" from the drop-down menu.
6. Double-click on your new .dat file. This activates the Excel Text Import Wizard.
7. Select "Delimited" - then "Next"
8. De-select "Tab" and select "Comma" - then "Next"
9. Under "Data Preview", select all columns by pressing and holding the shift key, and clicking each column header. (Please be sure to select all columns).
10. In the "Column Data Format" box, select "Text" - then "Finish"
11. You may now wish to readjust the appearance for easier reading. If desired, click the box at the intersection of the row and column headers to highlight the whole sheet, and select "Format" - "Column" - "AutoFit Selection" to adjust column sizes. Also, click the right-align button on the toolbar. (You may also wish to copy and move the date and time display into another area, or use it as a file title, to make the table narrower.)

We notice here that there is a difference in the theoretical price p_0 and the market price p. Apart from the fact that the market price is influenced by many factors (including the fact that most of the Black and Scholes hypotheses are not satisfied), one can interpret this saying that the market expectation on the exercise of this call is very small. From another point of view, one can instead consider the Black and Scholes formula replacing p_0 with p

$$p = S_0\Phi(d_1) - e^{-rT}K\Phi(d_2)$$

and solve it with respect to σ. The value of σ which satisfies the equality is called the *implied volatility*.

We can use the function `GBSVolatility` to solve this problem

```
R> p <- 4e-04
R> sigma.imp <- GBSVolatility(p, "c", S = S0, X = K, Time = T,
   r = r,
+      b = r)
R> sigma.imp

[1] 0.1557277
```

As we see, the implied volatility is lower than the historical volatility. This is interpreted again to mean that the market expects low probability of exercising the contract. The historical probability and the implied volatility rarely match. One reason is that the Black and Scholes model assumes a fixed volatility σ over time, while market actors know that volatility is far from being stable and try to predict its trend and levels. So, implied volatility incorporates the expectation of market actors on the options and the underlying assets.

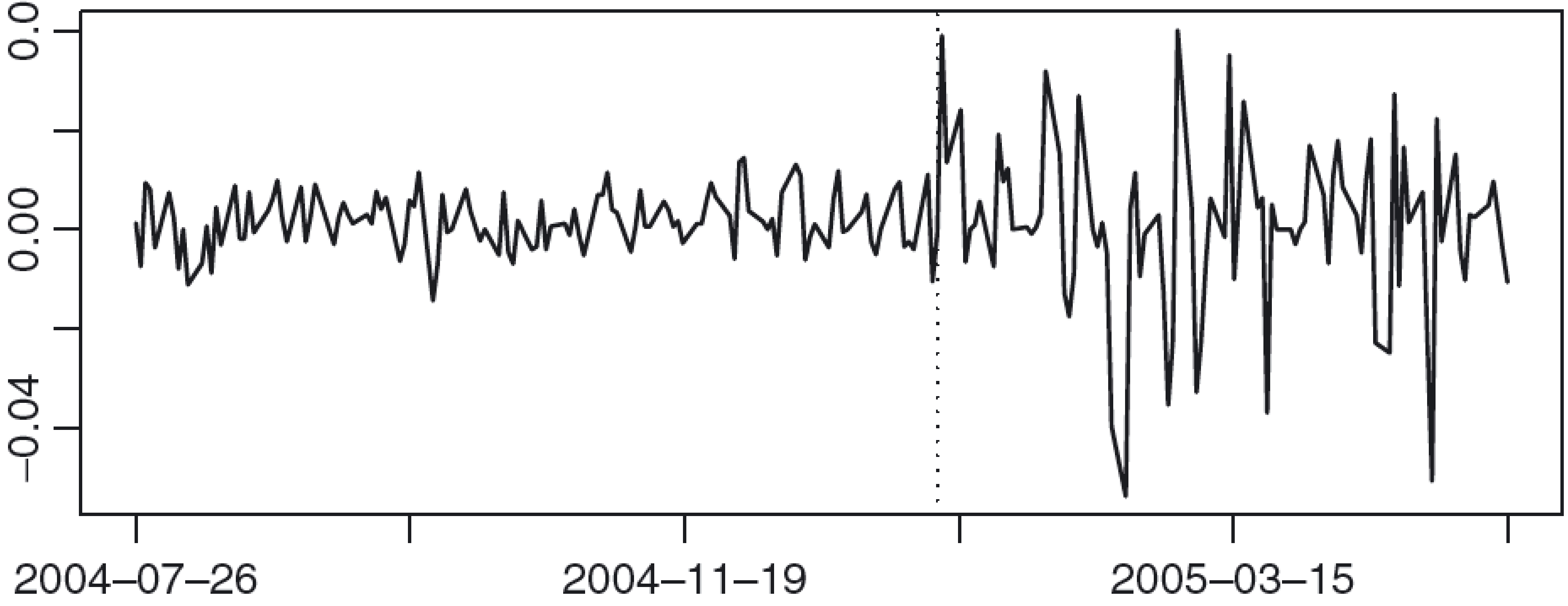

Figure 6.9 Returns of the ATL.MI asset with change point estimation.

`cpoint` function in the **sde** package. This function will allow for the discovery of a structural change point in the structure of the volatility of a generic stochastic differential equation following De Gregorio and Iacus (2008). In Section 9.1 we will discuss this in detail in a more general approach to the problem of change point in the volatility.

```
R> require(sde)
R> cp <- cpoint(as.ts(X))
R> cp
R> time(X)[cp$k0]
R> plot(X)
R> abline(v = time(X)[cp$k0], lty = 3)
```

6.6.1 Volatility smiles

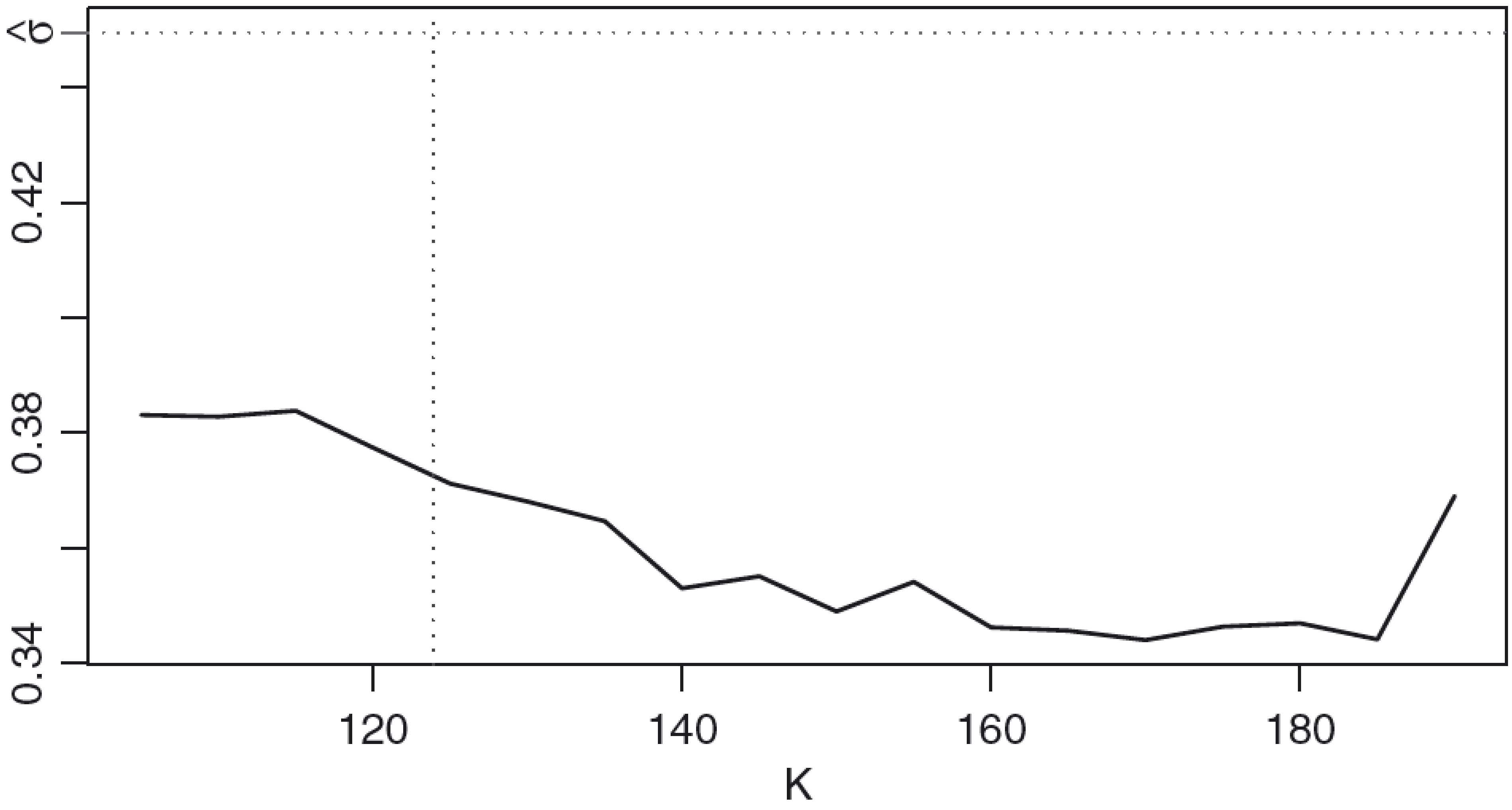

Figure 6.10 *Example of volatility smile: implied volatility as a function of the strike price K for given expiry date T. The vertical dotted line is the current price of the underlying asset and $\hat{\sigma}$ is the value of the historical volatility.*