

Chapter 1

Ex 1.1

We shall use the **getSymbols** function in the quantmod package to retrieve financial data for General Electric (**GE**).

```
getSymbols("GE", src = "yahoo", from = "2000-01-01", to = "2009-12-30")
```

'getSymbols' currently uses auto.assign=TRUE by default, but will use auto.assign=FALSE in 0.5-0. You will still be able to use 'loadSymbols' to automatically load data. getOption("getSymbols.env") and getOption("getSymbols.auto.assign") will still be checked for alternate defaults.

This message is shown once per session and may be disabled by setting options("getSymbols.warning4.0"=FALSE). See ?getSymbols for details.

```
[1] "GE"
```

```
names(GE)
```

```
[1] "GE.Open"      "GE.High"      "GE.Low"       "GE.Close"     "GE.Volume"
[6] "GE.Adjusted"
```

```
GE["2000-01-01/2000-01-20"]
```

	GE.Open	GE.High	GE.Low	GE.Close	GE.Volume	GE.Adjusted
2000-01-03	49.03846	49.25882	47.81650	48.07692	22952500	26.34747
2000-01-04	47.19552	47.43590	46.15385	46.15385	23006200	25.29357
2000-01-05	46.07372	47.11538	45.69311	46.07372	28384500	25.24966
2000-01-06	45.87340	47.09535	45.71314	46.68970	20668100	25.58724
2000-01-07	47.43590	48.67788	47.11538	48.49760	20947000	26.57802
2000-01-10	48.93830	49.37900	48.43750	48.47757	15835500	26.56703
2000-01-11	48.39743	48.93830	48.27725	48.55769	15727900	26.61095
2000-01-12	48.41747	49.11859	48.25721	48.71795	19075900	26.69878
2000-01-13	49.07853	49.65945	49.03846	49.27885	15551600	27.00616
2000-01-14	49.15865	49.55930	47.93670	48.39743	19219500	26.52312
2000-01-18	47.95673	47.95673	47.03525	47.43590	19028500	25.99617
2000-01-19	46.95513	48.37740	46.87500	47.66627	15443600	26.12242
2000-01-20	47.77644	47.99680	45.71314	46.77484	31989300	25.63389

```
geAdj = GE$GE.Adjusted["2000-01-01-/2000-01-20"]; geAdj
```

```

      GE.Adjusted
2000-01-03    26.34747
2000-01-04    25.29357
2000-01-05    25.24966
2000-01-06    25.58724
2000-01-07    26.57802
2000-01-10    26.56703
2000-01-11    26.61095
2000-01-12    26.69878
2000-01-13    27.00616
2000-01-14    26.52312
2000-01-18    25.99617
2000-01-19    26.12242
2000-01-20    25.63389

```

```
max(geAdj); min(geAdj); mean(geAdj)
```

```
[1] 27.00616
```

```
[1] 25.24966
```

```
[1] 26.17034
```

```
chartSeries(GE)
```



```
chartSeries(GE, TA=NULL, subset='2001-01::2001-02')
```



```
saveRDS(GE, file = "GE.rds")
```

1.3.4

```
symbols <- c('^VLIC', 'GE', 'KO', 'AAPL', 'MCD')
getSymbols(symbols, src = "yahoo", from = "2012-02-01", to = "2013-02-01")
```

```
[1] "^VLIC" "GE"    "KO"    "AAPL"  "MCD"
```

```
# obtain Adjusted Close
```

```
VLICad <- VLIC$VLIC.Adjusted; GEad <- GE$GE.Adjusted
KOad <- KO$KO.Adjusted; AAPLad <- AAPL$AAPL.Adjusted
MCDad <- MCD$MCD.Adjusted
```

```
# compute cumulative sum (cumsum) of daily returns (Delt)
```

```
vl <- cumsum( (Delt(VLICad) * 100)[-1, ])
ge <- cumsum( (Delt(GEad) * 100)[-1, ])
ko <- cumsum( (Delt(KOad) * 100)[-1, ])
ap <- cumsum( (Delt(AAPLad) * 100)[-1, ])
```

```
md <- cumsum( (Delt(MCDad) * 100)[-1, ] )

### Range for the plot

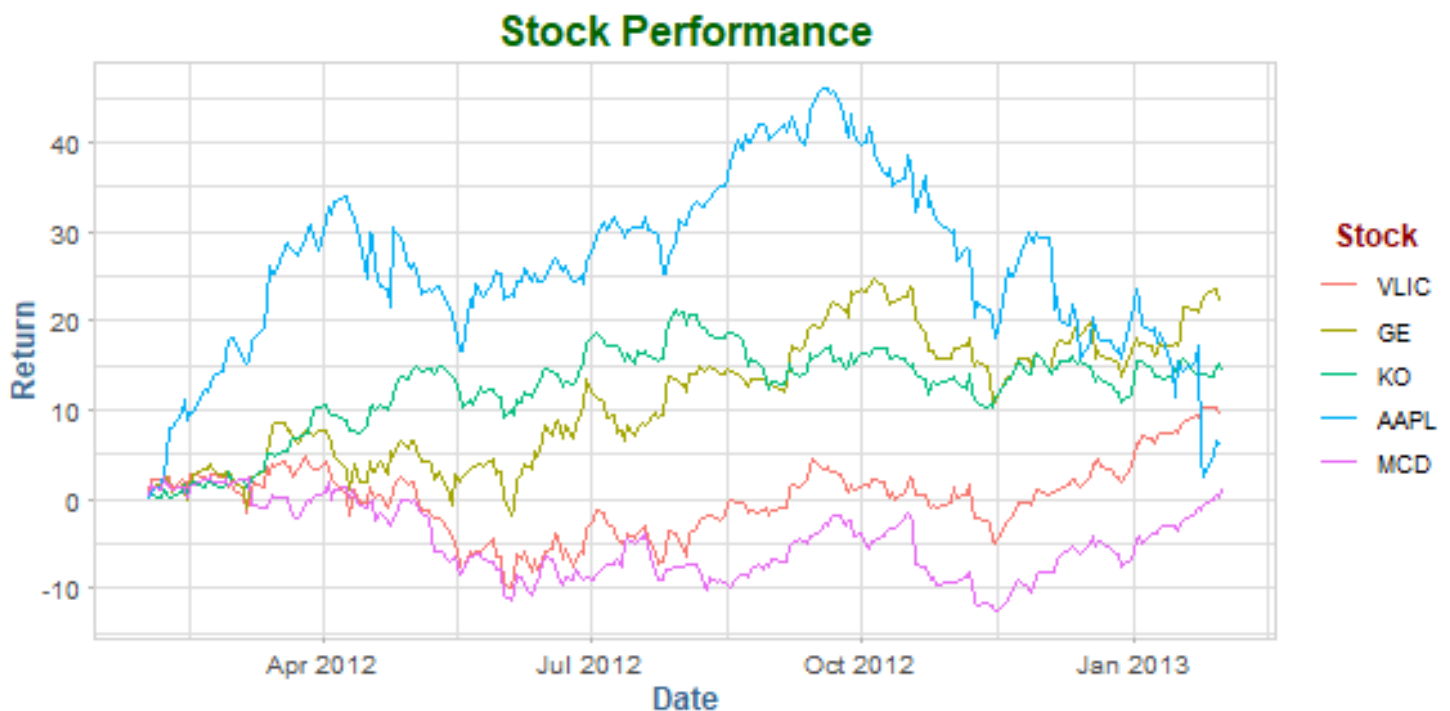
lim <- c(min(vl, ge, ko, ap, md), max(vl, ge, ko, ap, md))

### plot

stocks <- data.table( Date = index(vl),
                      VLIC = as.numeric(vl),
                      GE = as.numeric(ge),
                      KO = as.numeric(ko),
                      AAPL = as.numeric(ap),
                      MCD = as.numeric(md))

stocks.long <- melt(stocks,
                   id.vars = c("Date"),
                   variable.name = c("Stock"),
                   value.name = "Return")

ggplot(stocks.long, aes(Date, Return, col = Stock)) +
  geom_line() +
  labs(title = "Stock Performance")
```



1.3.6

Using no arbitrage arguments show that for options on stocks:

- i.) the stock's price is an upper bound for the price of a call;
- ii.) the strike price is an upper bound for the price of a put.

1.3.7

The following is a list of well-known investment strategies obtained by different combinations of put and call options on the same underlying asset. For each one of these strategies compute the payoff function and draw the profit graph.

Additionally, argue about the situations where the strategy is profitable.

```
plot.payoff <- function(payoff, strike) {
  breakeven <- payoff[min(which(payoff$Payoff > 0))]$Strike
  ggplot(payoff) +
    geom_point(aes(Strike, Payoff), col = ifelse(payoff$Payoff < 0, "darkred", "darkgreen"),
    geom_hline(yintercept = 0, col = "cornflowerblue", lwd = 1, alpha = .45) +
    labs(title = paste0("Option Payoff: K=$", strike, ", Break-Even: $", breakeven),
      x = "Strike", y = "Payoff") +
    scale_x_continuous(labels = dollar) +
    scale_y_continuous(labels = dollar)
}
```

Vanilla Call Payoff

```
call.payoff <- function( strike, initial.price, contracts = 1, contract.size = 100,
  lower = -.15, upper = .15) {

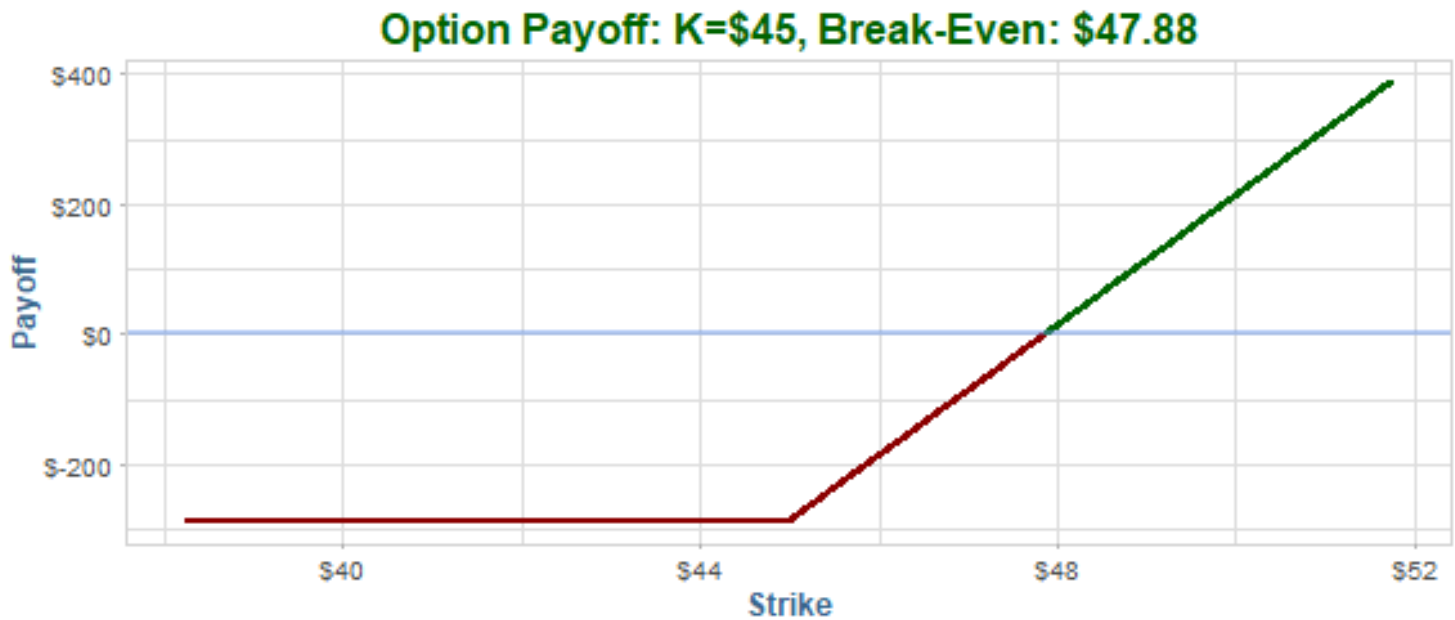
  cost <- -(initial.price * contract.size) * contracts

  x <- strike + seq(strike*lower, strike*upper, .01)
  y <- ( ( x - strike - initial.price ) * contract.size ) * contracts

  payoff <- ifelse(y < cost, cost, y)

  data.table(Strike = x, Payoff = payoff)
}

K <- 45
plot.payoff(call.payoff(K, 2.88), K)
```



Vanilla Put Payoff

```
put.payoff <- function( strike, initial.price, contracts = 1, contract.size = 100,
                        lower = -.15, upper = .15) {

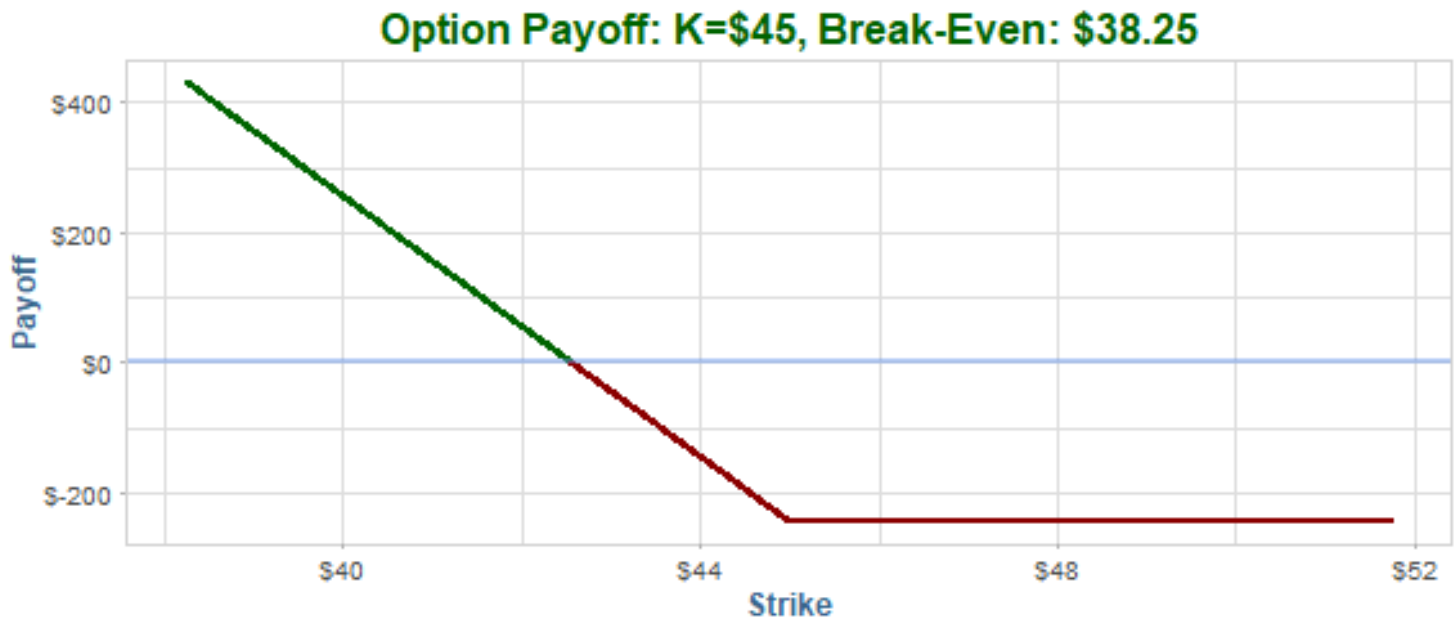
  cost <- -(initial.price * contract.size) * contracts

  x <- strike + seq(strike*lower, strike*upper, .01)
  y <- ( ( strike - x - initial.price ) * contract.size ) * contracts

  payoff <- ifelse(y < cost, cost, y)

  data.table(Strike = x, Payoff = payoff)
}

K <- 45
plot.payoff(put.payoff(K, 2.45), K)
```



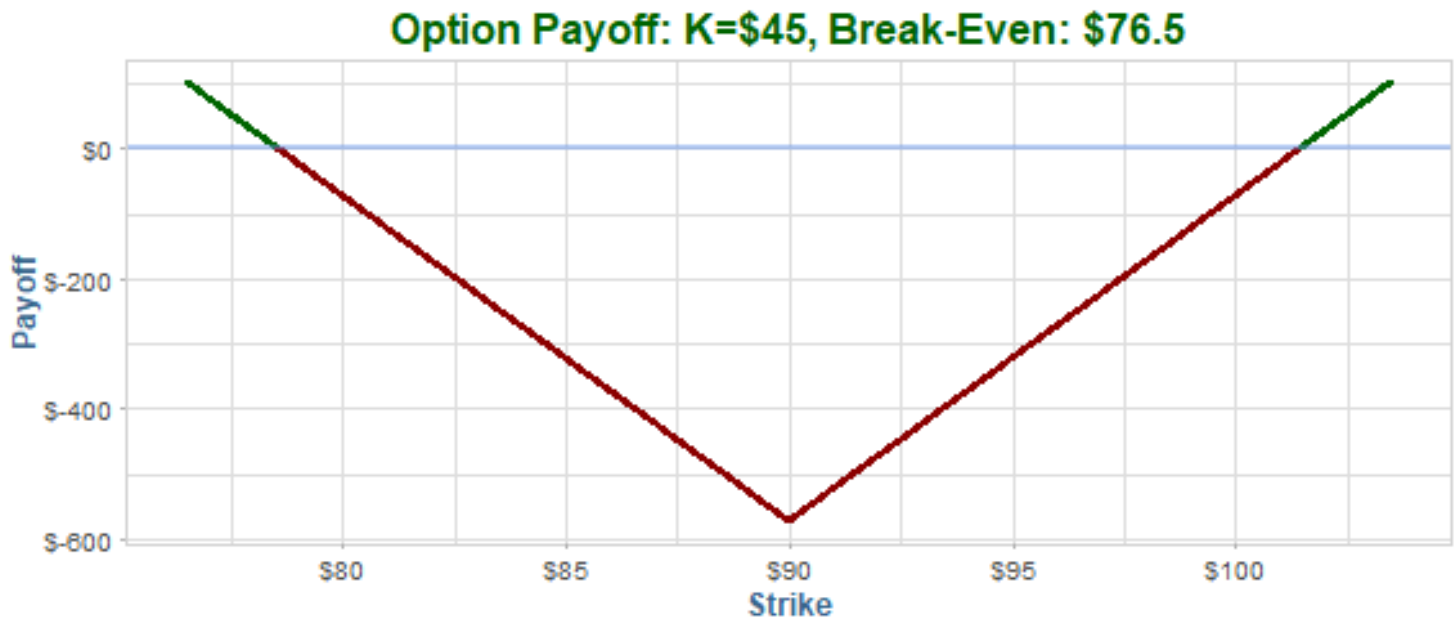
The Straddle:

The straddle strategy consists of two legs, one long and one short, both with the same strike and expiration date.

The main idea behind the straddle is that it is a non-directional long volatility strategy. It is generally suitable when you expect the underlying security to be very volatile and move a lot, but you are not sure whether the price move will be up or down. The position makes a profit when your expectation is correct and the underlying does make a big move to one side or the other. If the underlying price stays more or less the same, the trade makes a loss.

Straddle Payoff:

```
K <- 45
plot.payoff(call.payoff(K, 2.85) + put.payoff(K, 2.88), K)
```

**The strangle:**

Similar to the straddle. It consists on buying a put and a call with the same expiration date, but different strike prices. If K_c is the strike price for the call, and K_p is the strike price for the put, then the strategy requires $K_c > K_p$.

The strip:

This strategy consists of long positions in one call and two puts, all with the same strike price and expiration date.

The strap:

This one consists of long positions in two calls and one put, all with the same strike price and expiration date.

The butterfly spread:

This is made with options of the same type. Suppose we use calls and the underlying asset is a stock. Then a butterfly spread of calls consists on short selling two calls with strike price K_0 close to the current stock price, and buying two calls, one with strike price $K_0 - c$ and the other with strike price $K_0 + c$, where $2c > 0$ is the length of the spread chosen by the investor.