COMPUTATIONAL FINANCE & RISK MANAGEMENT
UNIVERSITY of WASHINGTON
Department of Applied Mathematics

Options in R

CFRM 425 (020)

R Programming for Quantitative Finance

Lecture References and Topics

- Peter James
  - Option Theory (Wiley, 2003)
    - General background
    - Excellent reference book to have in your arsenal
- The fOptions R package
  - https://www.rmetrics.org/

RMetrics

- Based in Zurich
- RMetrics package – Contains fOptions and other financial R packages

Rmetrics Foundation

- Based in Zurich
- Founded in 2007
- Financially supports
  - Summer Schools and Workshops
  - Student Internships
  - Open Source Software Development
  - Publications of eBooks
  - R Programming, R Studio, and R Shiny trainings and seminars

fOptions

- fOptions: Rmetrics - Pricing and Evaluating Basic Options
- Pricing models

00fOptions-package	Basic Option Valuation
BasicAmericanOptions	Valuation of Basic American Options
BinomialTreeOptions	Binomial Tree Option Model
HestonNandiGarchFit	Heston-Nandi Garch(1,1) Modelling
HestonNandiOptions	Option Price for the Heston-Nandi Garch Option Model
LowDiscrepancy	Low Discrepancy Sequences
MonteCarloOptions	Monte Carlo Valuation of Options
PlainVanillaOptions	Valuation of Plain Vanilla Options

Plain Vanilla Option

- Pricing models

```
GBS*                      the generalized Black-Scholes option
BlackScholesOption        a synonyme for the GBSOption
Black76Option             options on Futures
MiltersenSchwartzOption   options on commodity futures
```

- View Results

```
print                     print method for Options
summary                   summary method for Options
```

Plain Vanilla Option

- First Example: European Call Option:

```
GBSOption(TypeFlag, S, X, Time, r, b, sigma)

      TypeFlag = "c", "p" (call/put)
      S = Market share price
      X = Strike price
      Time = time to expiration,as a year fraction
      r = risk-free interest rate
      b = annual dividend rate
      sigma = return volatility
```

- First Example: European Call Option:

```
TypeFlag<-"c"
S<-530.32
X<-550
Time<-283/365
r<-0.02
b<-0.05
sigma<-0.24

res.GBS <- GBSOption(TypeFlag, S, X, Time, r, b, sigma)
(summ <- summary(res.GBS))
```

```
Call:
 GBSOption(TypeFlag = TypeFlag, S = S, X = X, Time = Time, r = r,
     b = b, sigma = sigma)

Parameters:
           Value:
 TypeFlag c
 S        530.32
 X        550
 Time     0.775342465753425
 r        0.000703
 b        0
 sigma    0.2442

Option Price:
 37.04709

Description:
 Sat Feb 29 17:55:58 2020
```

Plain Vanilla Option

- First Example: European Call Option – We can also get results from *slots* on the return object

```
summ@price
summ@parameters
summ@parameters$x
```

```
> summ@price
[1] 37.04709
```

```
> summ@parameters
$TypeFlag
[1] "c"

$s
[1] 530.32

$x
[1] 550

$Time
[1] 0.7753425

$r
[1] 0.000703

$b
[1] 0

$sigma
[1] 0.2442
```

```
> summ@parameters$x
[1] 550
```

- In addition, we can calculate the option price along with the associated "Greeks", by calling the function **`GBSCharacteristics(.)`**
- Same input data as before:

```
TypeFlag<-"c"
S<-530.32
X<-550
Time<-283/365
r<-0.02
b<-0.05
sigma<-0.24
```

- Now:

```
chars.GBS <- GBSCharacteristics(TypeFlag, S, X, Time, r, b, sigma)
class(chars.GBS)   # list
names(chars.GBS)  # "premium" "delta"   "theta"   "vega"    "rho"     "lambda"  "gamma"
# Note: lambda is the same thing as vega
```

```
(resultSet <- c(chars.GBS$premium, chars.GBS$delta, chars.GBS$theta,
                chars.GBS$vega, chars.GBS$rho, chars.GBS$lambda, chars.GBS$gamma))
# 46.258450018   0.559305500 -43.216176904 189.382574311 194.108880452   6.412036998   0.003618755
```

Black-Scholes Closed Form: European Options

- Let's look at a series of put options
  - Just before expiration
  - Three months to expiration
  - Series of underlying prices

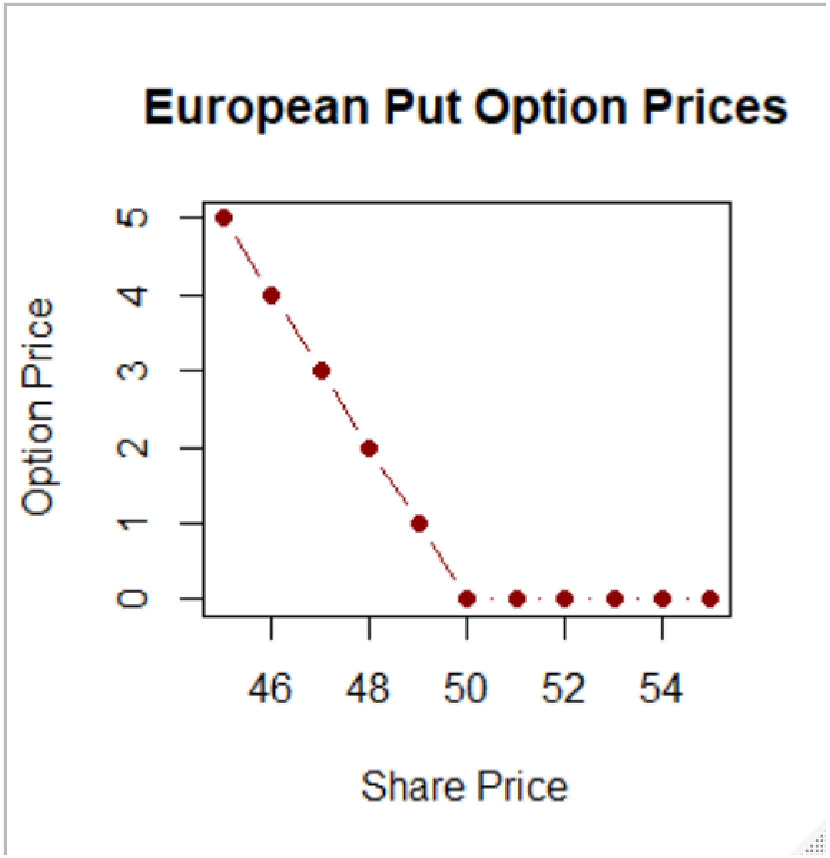
```
TypeFlag<-"p"
SVec <- seq(from = 45, to = 55, by = 1)
X<-50
Time <- 0.001/365   # Very close to the bell
r<-0.01
b<-0.0     # No dividend
sigma<-0.25

puts.GBS.Maturity <- GBSOption(TypeFlag, SVec, X, Time, r, b, sigma)
summPuts.Mat <- summary(puts.GBS.Maturity)
matPrices <- summPuts.Mat@price
```

```
> matPrices
 [1] 4.99999986 3.99999989 2.99999992 1.99999995
 [5] 0.99999997 0.00825428 0.00000000 0.00000000
 [9] 0.00000000 0.00000000 0.00000000
```

Black-Scholes Closed Form: European Options

- Now, let's look at a plot

```
plot(x = sVec, y = matPrices, type = "b", col = 'darkred',
     main = 'European Put Option Prices Close to Mty',
     xlab = 'Strike Price', ylab = 'Option Price', lwd = 1.5, pch = 16)
```



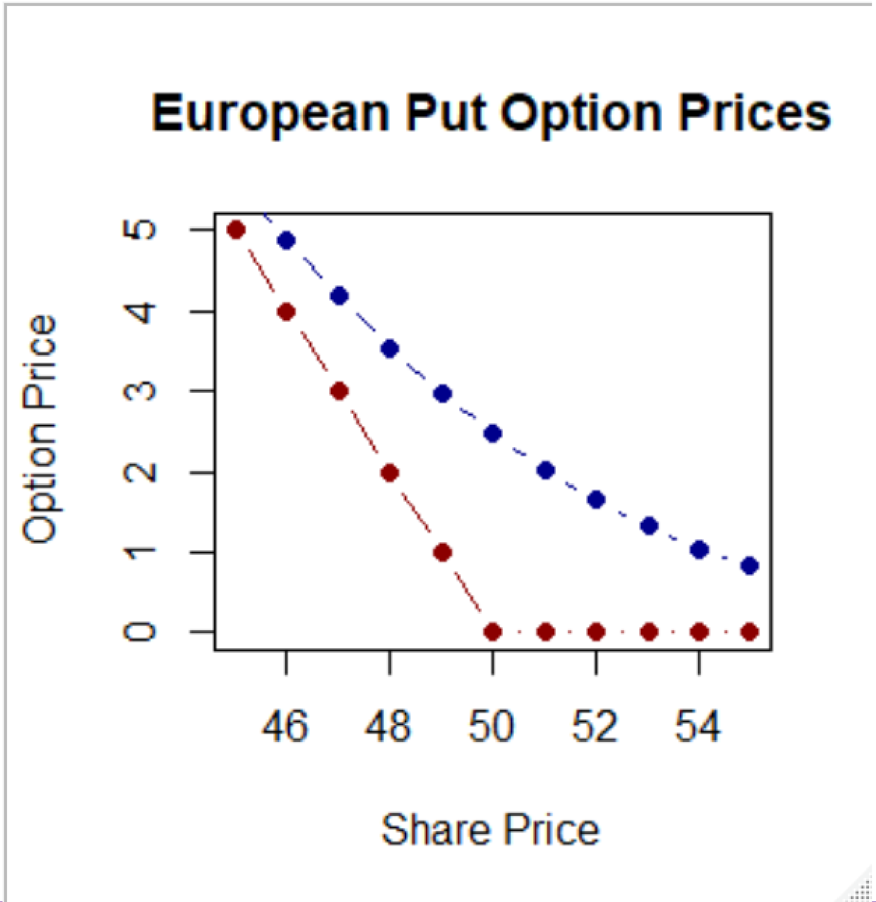
- Note that the option prices are equal to the intrinsic values
- Time value is essentially zero

Black-Scholes Closed Form: European Options

- Next, let's look at option prices at three months to expiration
- All other parameters remain the same

```
Time <- 90/365
puts.GBS.3M <- GBSOption(TypeFlag, SVec, X, Time, r, b, sigma)
summPuts.3M <- summary(puts.GBS.3M)
threeMthPrices <- summPuts.3M@price
```

- Now, overlay these results on the previous plot
- Time value is now positive
- Adds to intrinsic value
- Higher option prices



American Options

- Recall: American options may be exercised anytime up to and including the contract expiration date
- In fOptions, we will use the following:
  - Barone-Adesi & Whaley (1987): Closed form approximation
    - https://deriscope.com/docs/Barone_Adesi_Whaley_1987.pdf
  - Bjerksund and Stensland (2002): Closed form approximation
    - https://core.ac.uk/download/pdf/30824897.pdf
  - Jarrow-Rudd Binomial Tree: Numerical approximation
    - See the James book
- Time-permitting, we will look at Monte Carlo pricing outside of fOptions – maybe roll your own

- Example 1: Price a single American option (Barone-Adesi & Whaley):

```
# Quick single option contract example:
TypeFlag<-"p"
X<-50
r<-0.01
b<-0.0    # No dividend
sigma<-0.25
```

```
put.BAW <- BAWAmericanApproxOption(TypeFlag, S = 55,
                                   X, Time = 90/365, r, b, sigma)
put.BAW@price
# Option price = 0.8253149

class(put.BAW)  # foptions
```

- Example 2: Same input data with same vector (as before) of underlying equity prices **sVec** (Barone-Adesi & Whaley):

```
puts.BAW.3M <- BAWAmericanApproxOption(TypeFlag, S = sVec,
                                       X, Time = 90/365, r, b, sigma)

summBAW <- summary(puts.BAW.3M)
(BAW.Prices <- summBAW@price)
```

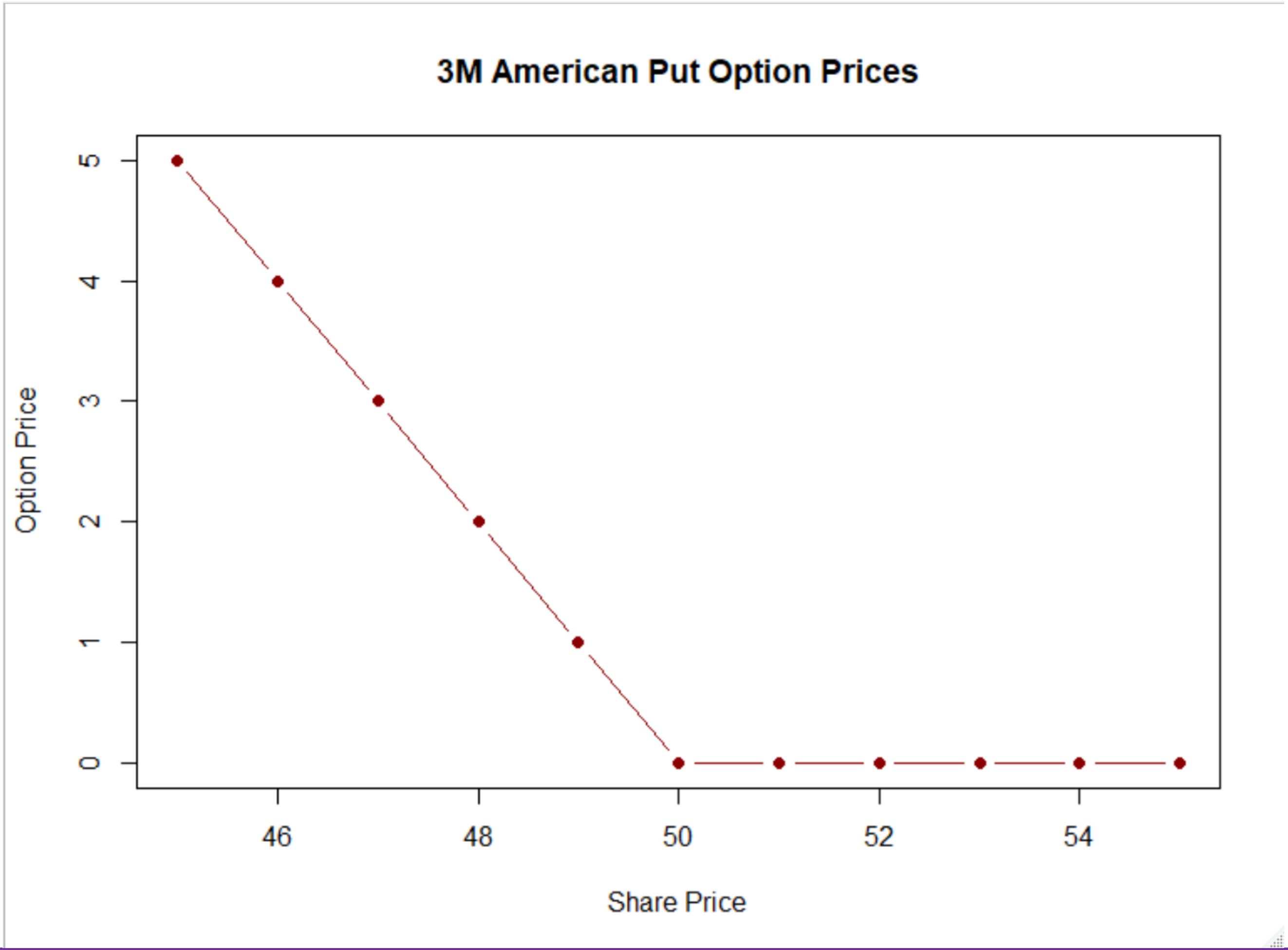
- Result:

```
[1] 5.6368572 4.8811371 4.1832868 3.5473923 2.9758651 2.4693538 2.0267202
[8] 1.6453608 1.3213433 1.0498102 0.8253149
```

- Next, let's plot this against the intrinsic values only at expiration

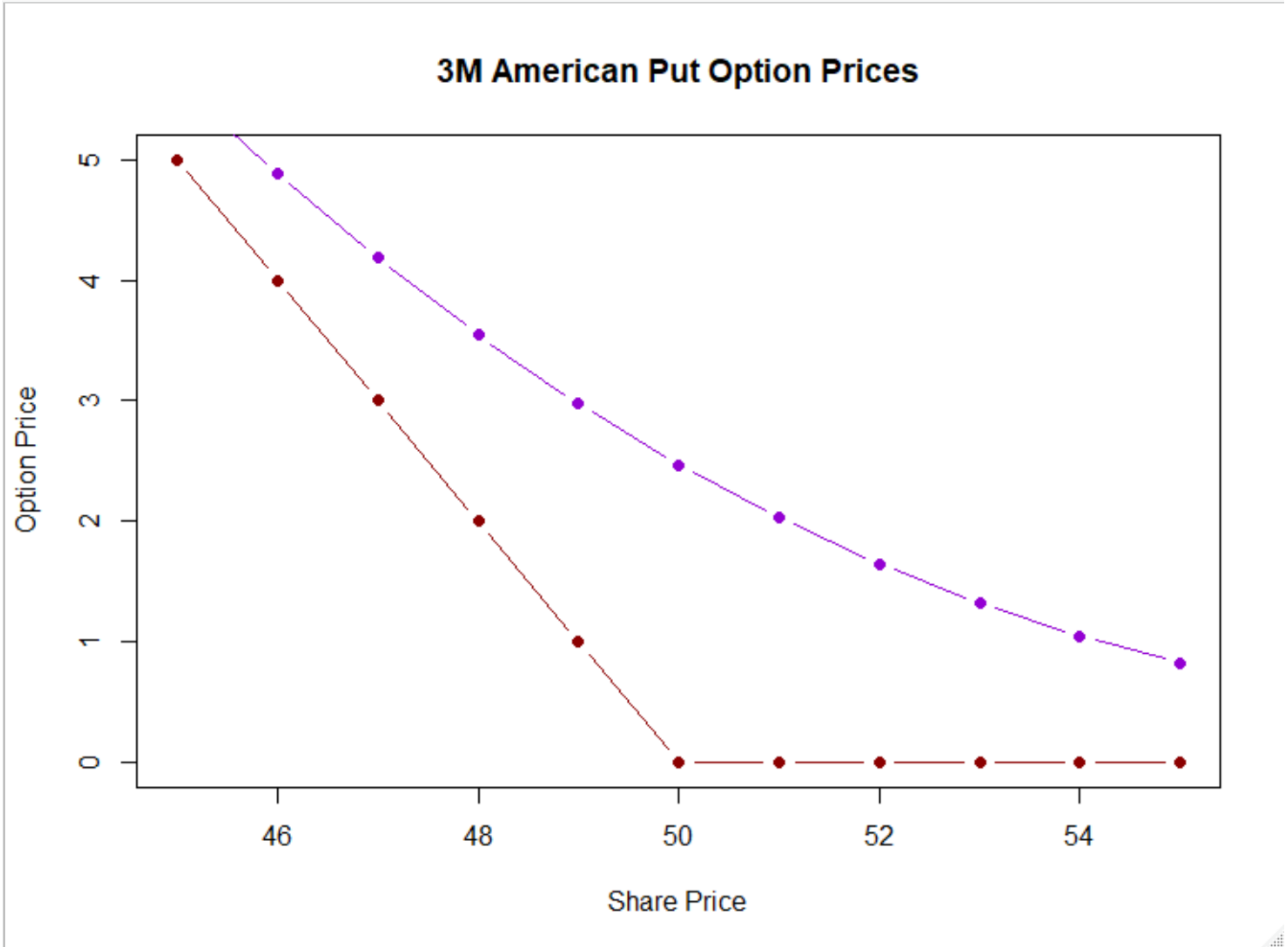
- Intrinsic values (from the European case):

```
# Put intrinsic value plot again first:
plot(x = sVec, y = matPrices, type = "b", col = 'darkred',
     main = '3M American Put Option Prices',
     xlab = 'Share Price', ylab = 'Option Price', lwd = 1.5, pch = 16)
```



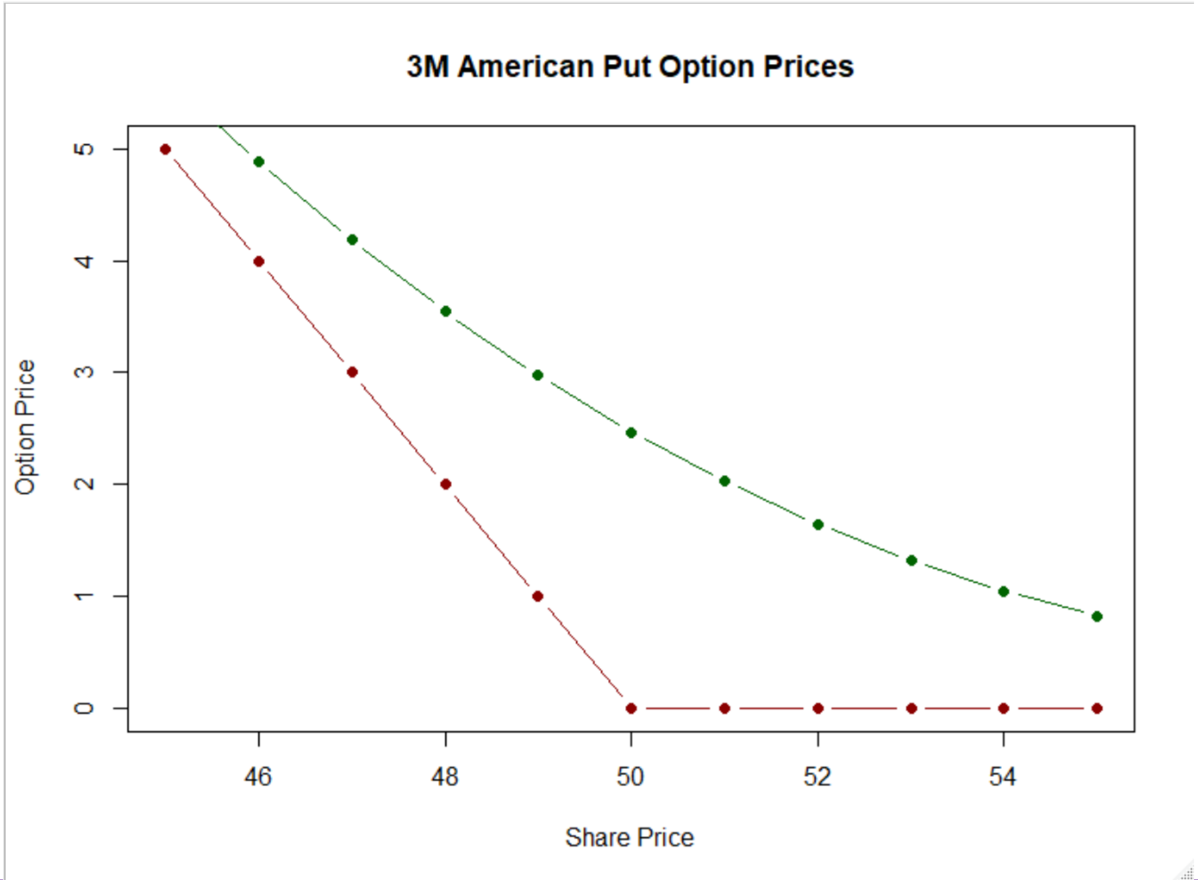
- Overlay the BAW American option prices

```
lines(x = sVec, y = BAW.Prices, type = "b", col = 'darkviolet',
      lwd = 1.5, pch = 16)
```



- Let's do the same with the Bjerksund and Stensland Approximation:

```
puts.BjSt.3M <- BSAmericanApproxOption(TypeFlag, S = SVec, X,
                                       Time, r, b, sigma)

summBjSt <- summary(puts.BjSt.3M)
BjSt.Prices <- summBjSt@price

lines(x = SVec, y = BjSt.Prices, type = "b", col = 'darkgreen',
      lwd = 1.5, pch = 16)
```



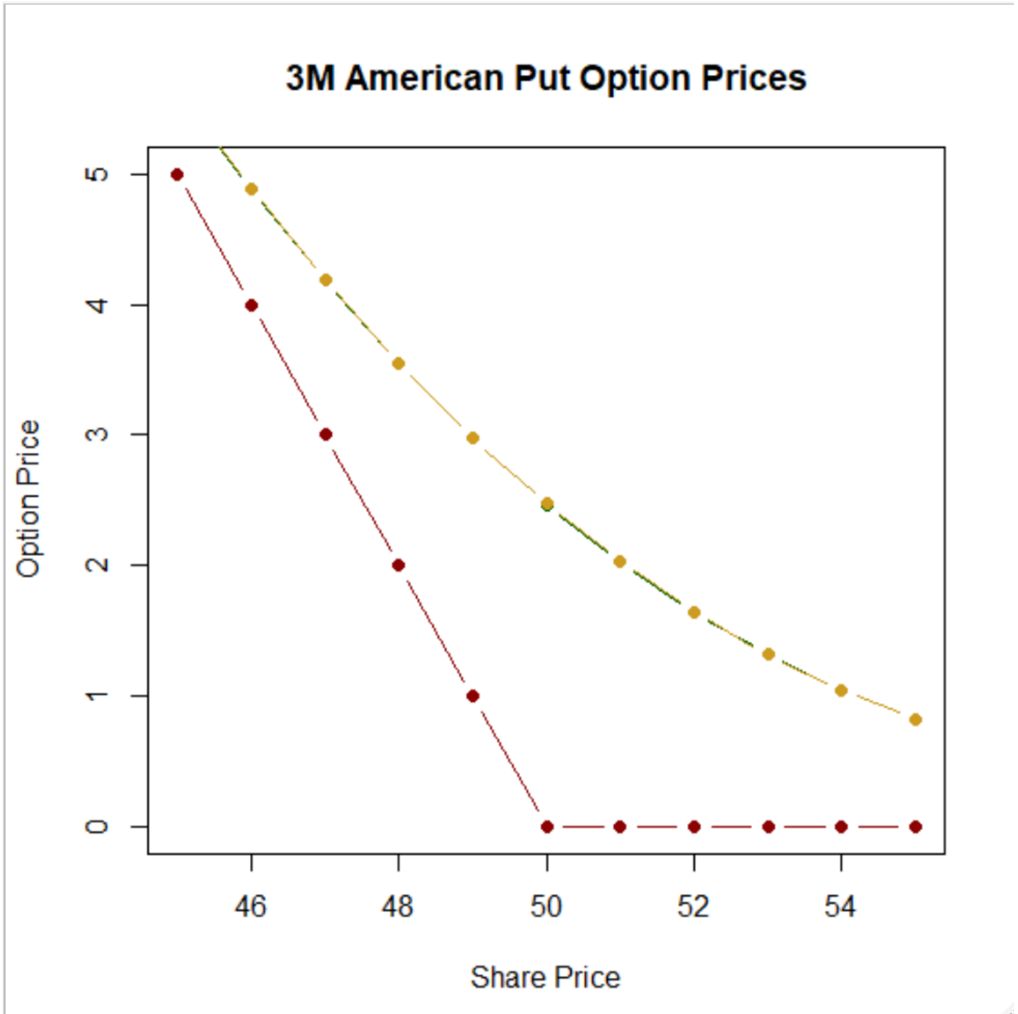
- Finally, with the Binomial Tree approximation: Chop the timeline into 100 increments of length Δt
- As we can't vectorize this model with the vector of equity prices, use a loop

```
v <- vector("double", 0L)
for(s in SVec)
{
  put.JR <- JRBinomialTreeOption(TypeFlag = "pa", S = s, X,
                    Time, r, b, sigma, n = 100, title = "American 3M Puts",
                    description = "Varying Equity Prices")

  v <- c(v, put.JR@price)
}
```

American Options

- Binomial Tree approximation (continued): Overlay the vector of option prices at each underlying share price

```
lines(x = svec, y = v, type = "b", col = 'goldenrod3',
      lwd = 1.5, pch = 16)
```



- [END]