Black & Scholes Model

Th. Warin

Black-Scholes

- Most widely-used option pricing model
- Hard to fully understand
- Hard to prove
- Easy to use (especially in Excel)

Digression: Mathematical theorems

- Every good mathematical theorem:
  - Starts with a couple of **Ifs**
  - Has a **Then**
  - A really good theorem follows "Then" with **Where**
- **Black-Scholes is a really good theorem!**

Black-Scholes as a Theorem: IF

- **If** the stock price is lognormally distributed
- **If** the stock has no dividends before option expiration **T**
- **If** the option is European

Black-Scholes as a theorem: THEN

- **Then** the call and put price are given by

$$C = SN(d_1) - Xe^{-rT}N(d_2)$$

$$P = \text{-}SN(\text{-}d_1) + Xe^{-rT}N(\text{-}d_2)$$

$$where$$

$$d_1 = \frac{Ln(S/X) + \left(r + \sigma^2/2\right)T}{\sigma\sqrt{T}}$$

$$d_2 = d_1 - \sigma\sqrt{T}$$

Black-Scholes theorem: WHERE

- **Where** N() indicates values of the standard normal distribution
- σ is the standard deviation of the stock's return
- T is the option exercise date

Black-Scholes proof

- Based on arbitrage between the Stock and riskless Bond
- The arbitrage is **dynamic**, meaning that it is continuously updated
  - Depends on stock price S_t
  - Time to option maturity $T - t$
  - Riskiness of stock return σ

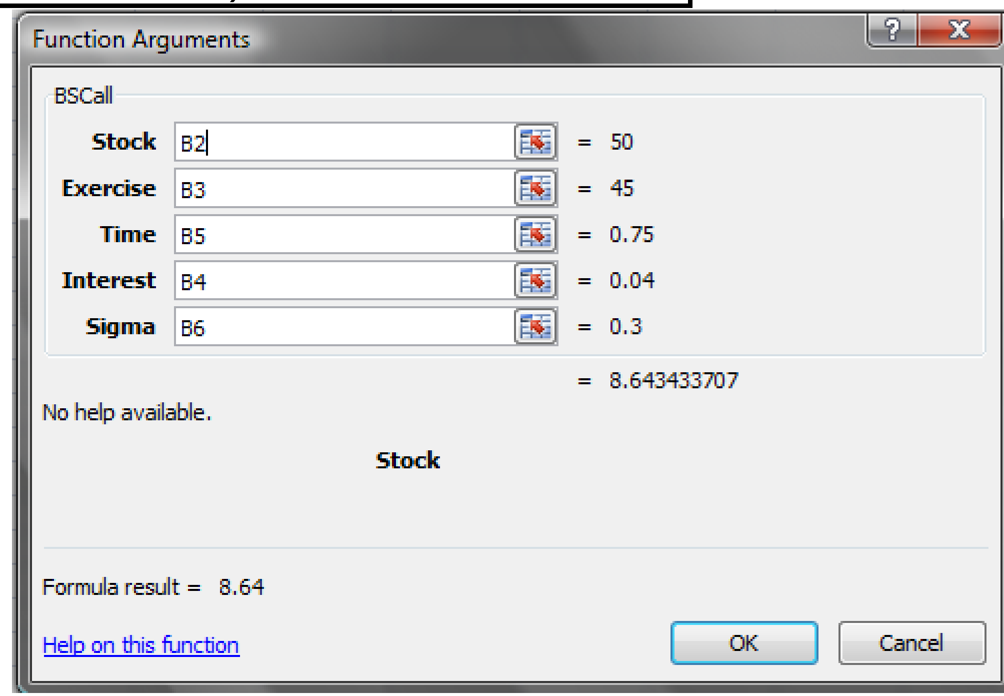
Black-Scholes is easy to implement in Excel

	A	B	C
1	**Black-Scholes Option-Pricing Formula**		
2	S	50	Current stock price
3	X	45	Exercise price
4	r	4.00%	Risk-free rate of interest
5	T	0.75	Time to maturity of option (in years)
6	Sigma	30%	Stock volatility, σ
7			
8	d_1	0.6509	<-- (LN(S/X)+(r+0.5*sigma^2)*T)/(sigma*SQRT(T))
9	d_2	0.3911	<-- d_1-sigma*SQRT(T)
10			
11	$N(d_1)$	0.7424	<-- Uses formula NormSDist(d_1)
12	$N(d_2)$	0.6521	<-- Uses formula NormSDist(d_2)
13			
14	Call price	8.64	<-- S*N(d_1)-X*exp(-r*T)*N(d_2)
15	Put price	2.31	<-- call price - S + X*Exp(-r*T): by Put-Call parity
16		2.31	<-- X*exp(-r*T)*N(-d_2) - S*N(-d_1): direct formula

Note use of Excel formula **NormSDist** to compute N( )

Black-Scholes

	A	B	C
1	**Black-Scholes in VBA**		
2	S	50	Current stock price
3	X	45	Exercise price
4	r	4.00%	Risk-free rate of interest
5	T	0.75	Time to maturity of option (in years)
6	Sigma	30%	Stock volatility, σ
7			
8	Call	8.64	<-- =BSCall(B2,B3,B5,B4,B6)
9	Put	2.31	<-- =BSPut(B2,B3,B5,B4,B6)



VBA functions BSCall and BSPut

- Part of the file **fm3_chapter19.xls** that comes with *Financial Modeling*
- If you want them in the other spreadsheets, you have to **Copy|Paste** as indicated in the file "Adding GetFormula to your spreadsheet.doc" that is on the disk to Financial Modeling.

BSCall and BSPut

```
Function dOne(Stock, Exercise, Time, Interest, sigma)
    dOne = (Log(Stock / Exercise) + Interest * Time) / _
     (sigma * Sqr(Time)) + 0.5 * sigma * Sqr(Time)
End Function
Function dTwo(Stock, Exercise, Time, Interest, sigma)
    dTwo = dOne(Stock, Exercise, Time, Interest, sigma) - _
     sigma * Sqr(Time)
End Function
Function BSCall(Stock, Exercise, Time, Interest, sigma)
    BSCall = Stock * Application.NormSDist(dOne(Stock, Exercise, _
   Time, Interest, sigma)) - Exercise * Exp(-Time * Interest) * _
   Application.NormSDist(dTwo(Stock, Exercise, Time, Interest, sigma))
End Function
'Put pricing function uses put-call parity theorem
Function BSPut(Stock, Exercise, Time, Interest, sigma)
    BSPut = BSCall(Stock, Exercise, Time, Interest, sigma) _
     + Exercise * Exp(-Interest * Time) - Stock
End Function
```

The BS parameters

- Most are easy:
  - S = current stock price
  - X = option exercise price
  - T = time to option maturity
  - r = risk-free interest rate (Treasury rate)
- **Only σ is difficult!**

Two approaches to σ

- σ from historical prices/returns
  - Annalize standard deviation by multiplying by $\sqrt{n}$:
    - n = 12 if data is monthly
    - n = 52 if data is weekly
    - n = 250, 252, 365 (???) if data is daily
- Implied σ
  - Find σ so that BS price = market price

Historical σ: Gerdau (GNA) and SP500

	A	B	C	D	E	F	G
1	**COMPUTING HISTORICAL VOLATILITY FOR GERDAU (GNA) AND THE SP500**						
2		**Prices**			**Returns**		
3	**Date**	**GNA**	**SP500**		**GNA**	**SP500**	
4	10/1/2004	4.23	1130.20				
5	11/1/2004	5.05	1173.82		17.72%	3.79%	<-- =LN(C5/C4)
6	12/1/2004	5.90	1211.92		15.56%	3.19%	<-- =LN(C6/C5)
7	1/3/2005	5.17	1181.27		-13.21%	-2.56%	<-- =LN(C7/C6)
8	2/1/2005	6.25	1203.60		18.97%	1.87%	<-- =LN(C8/C7)
9	3/1/2005	5.30	1180.59		-16.49%	-1.93%	
10	4/1/2005	4.32	1156.85		-20.45%	-2.03%	
11	5/2/2005	4.28	1191.50		-0.93%	2.95%	
12	6/1/2005	3.81	1191.33		-11.63%	-0.01%	

Gerdau is steel producer, Brazil-based, listed on NYSE. The data is monthly data from Nov. 2004 through Jan. 2009.

	I	J	K	L
4		**GNA**	**SP500**	
5	Monthly sigma	16.49%	4.23%	<-- =STDEVP(F5:F55)
6	Annualized sigma	57.12%	14.65%	<-- =SQRT(12)*K5

Black-Scholes requires annualized σ

	I	J	K	L
4		**GNA**	**SP500**	
5	Monthly sigma	16.49%	4.23%	<-- =STDEVP(F5:F55)
6	Annualized sigma	57.12%	14.65%	<-- =SQRT(12)*K5
7				
8				
9	**Implement Black-Scholes using historical σ** **At-the-money options**			
10	Stock price, S	6.22	831.95	<-- =C55
11	Exercise price, X	6.22	831.95	
12	Time to maturity, T	1		
13	Interest rate, r	2.00%		
14	Sigma, σ	57.12%	14.65%	
15	BS Call	1.45	56.80	<-- =BSCall(K10,K11,J12,J13,K14)
16	BS Put	1.32	123.12	<-- =BSPut(K10,K11,J12,K14,J14)
17				
18	BSCall/Stock price	23.26%	6.83%	<-- =K15/K10
19	BSPut/Stock price	21.28%	14.80%	<-- =K16/K10

Rows 18&19: The Call and Put are relatively more valuable (wrt to stock price) for GNA than for SP500. Why? $\sigma_{GNA} > \sigma_{SP500}$.

Note about Gerdau, other steel companies, and S&P

	A	B	C	D	E	F	G	H
1	**RETURN DATA FOR GERDAU AND COMPETITORS November 2004 - January 2009**							
2		**Arcelor Mittal (MT)**	**AK Steel Holding (AKS)**	**United States Steel (X)**	**Nucor (NUE)**	**Gerdau (GNA)**	**Posco (PKX)**	**SP500 (^SPX)**
3	Alpha	0.0075	0.0185	0.0122	0.0184	0.0217	0.0181	
4	Beta	2.7716	3.6625	3.1937	1.1476	2.5523	1.7582	
5	R-squared	0.5371	0.4738	0.5492	0.1817	0.4286	0.4352	
6	T-statistic, alpha	0.4746	0.7768	0.6836	1.2198	1.1936	1.4627	
7	T-statistic, beta	7.4634	6.5738	7.6471	3.2647	6.0004	6.0813	
8								
9	Monthly sigma	15.98%	22.49%	18.21%	11.38%	16.48%	11.26%	4.23%
10	Annualized sigma	55.36%	77.90%	63.09%	39.41%	57.07%	39.02%	14.64%

Steel has high beta and high sigma!

Implied call volatility VBA

```
Function CallVolatility(Stock, Exercise, Time, Interest, Target)
    High = 2
    Low = 0
    Do While (High - Low) > 0.0001
    If BSCall(Stock, Exercise, Time, Interest, (High + Low) / 2) > _
        Target Then
            High = (High + Low) / 2
            Else: Low = (High + Low) / 2
    End If
    Loop
    CallVolatility = (High + Low) / 2
End Function
```

Book has similar function **PutVolatility**

Note stopping condition **While(High-Low) > 0.0001** . Do not use High = Low as a stopping condition—it may never end!

Implied volatility

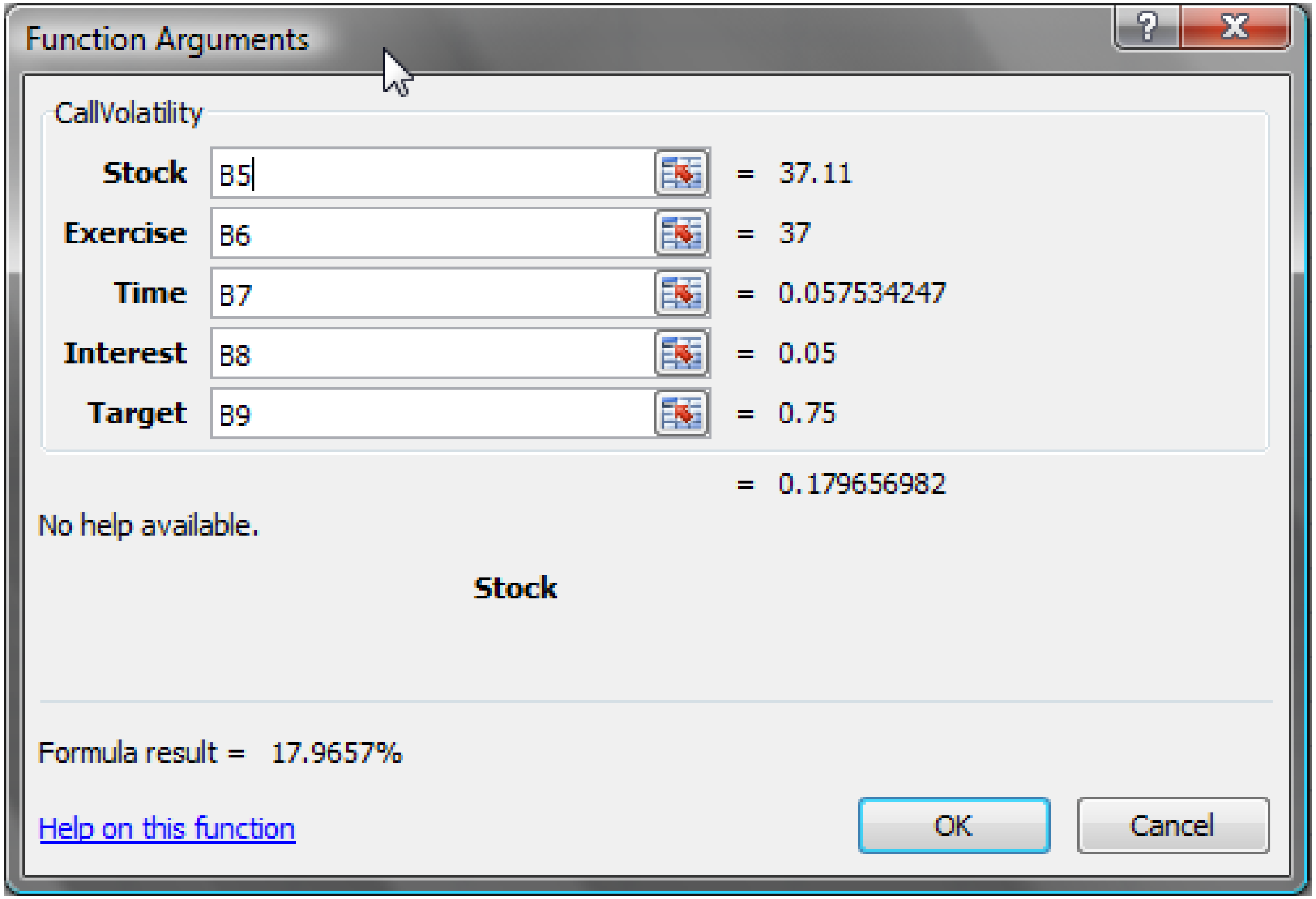
- Book example, page 517: QQQQ call has price \$0.75 (Nasdaq 100 index). With details below ➔implied σ = 17.97%.

	A	B	C
1	**IMPLIED VOLATILITY FOR THE AUGUST 2006 QQQQ OPTIONS**		
2	Current date	28-Jul-06	
3	Option expiration date	18-Aug-06	
4			
5	S	37.11	
6	X	37.00	
7	T	0.06	<-- =(B3-B2)/365
8	Interest	5.00%	
9	Implied volatility, σ	17.97%	
10			
11	Call price	0.7500	<-- =BSCall(B5,B6,B7,B8,B9)

Using CallVolatility function for same example

	A	B	C
1	**Using the *Financial Modeling* Function CallVolatility to find the Implied σ**		
2	Current date	28-Jul-06	
3	Option expiration date	18-Aug-06	
4			
5	S	37.11	
6	X	37.00	
7	T	0.06	<-- =(B3-B2)/365
8	Interest	5.00%	
9	Current call price	0.75	
10	Implied volatility	17.97%	<-- =CallVolatility(B5,B6,B7,B8,B9)

CallVolatility dialog box



CallVolatility is a user-defined function which is on the file **fm3_chapter19.xls** . To use function in other spreadsheets, it must be copied into VBA.

Dividend adjustments

- Black-Scholes must be adjusted for known or anticipated dividends on stock before option maturity T
- Method: Deduct from stock price S_0 the present value of future dividends before maturity.
- For discrete dividends: See *Financial Modeling* for Coca Cola example (p. 523)

Continuous dividends: Merton adjustment to Black-Scholes

- Assumption: stock pays out continuous dividend stream
- Appropriate for indices (say, options on S&P500)
- Used to price options on foreign currency (forex)

Merton formula, continuous dividends

- *k* is dividend yield on stock

$$C = S\,e^{-kT} N(d_1) - Xe^{-rT} N(d_2),$$

where

$$d_1 = \frac{\ln(S/X) + \left(r - k + \sigma^2/2\right)T}{\sigma\sqrt{T}}$$

$$d_2 = d_1 - \sigma\sqrt{T}$$

Pricing spiders: k = 1.7% (cell B6)

	A	B	C
1	**Merton's Dividend-Adjusted Option Pricing Model used here to price S&P 500 Spiders (symbol: SPY)**		
2	S	127.98	current stock price
3	X	127.00	exercise price
4	T	0.6329	<-- option expires 16-Mar-07, today's date 28-Jul-06
5	r	5.00%	risk-free rate of interest
6	k	1.70%	dividend yield
7	Sigma	14%	stock volatility
8	d_1	0.3122	<-- =(LN(B2/B3)+(B5-B6+0.5*B7^2)*B4)/(B7*SQRT(B4))
9	d_2	0.2008	<-- =B8-B7*SQRT(B4)
10			
11	$N(d_1)$	0.6226	<--- Uses formula NormSDist(d_1)
12	$N(d_2)$	0.5796	<--- Uses formula NormSDist(d_2)
13			
14	Call price	7.51	<-- S*Exp(-k*T)*N(d_1)-X*exp(-r*T)*N(d_2)
15	Put price	3.94	<-- call price - S*Exp(-k*T) + X*Exp(-r*T): by Put-Call parity
16		3.94	<-- X*exp(-r*T)*N(-d_2)-S*Exp(-k*T)*N(-d_1): direct formula