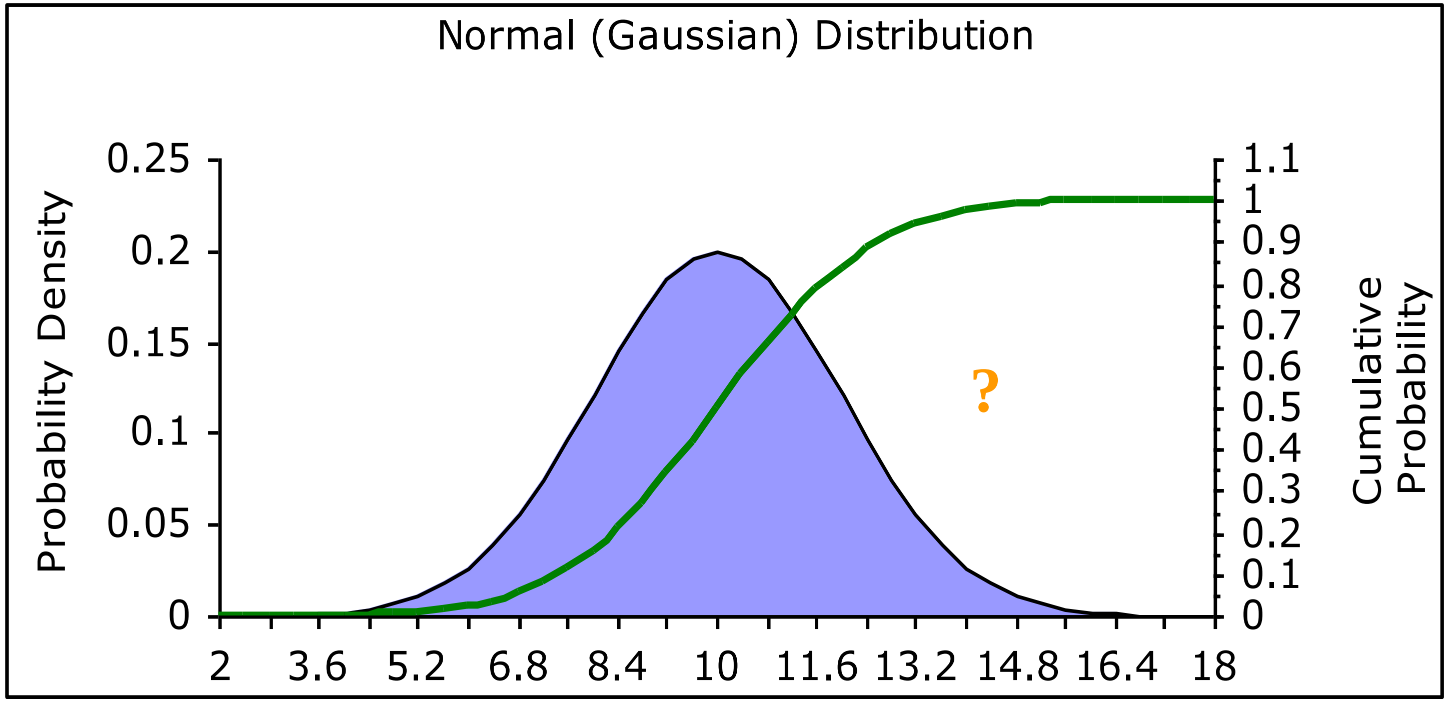
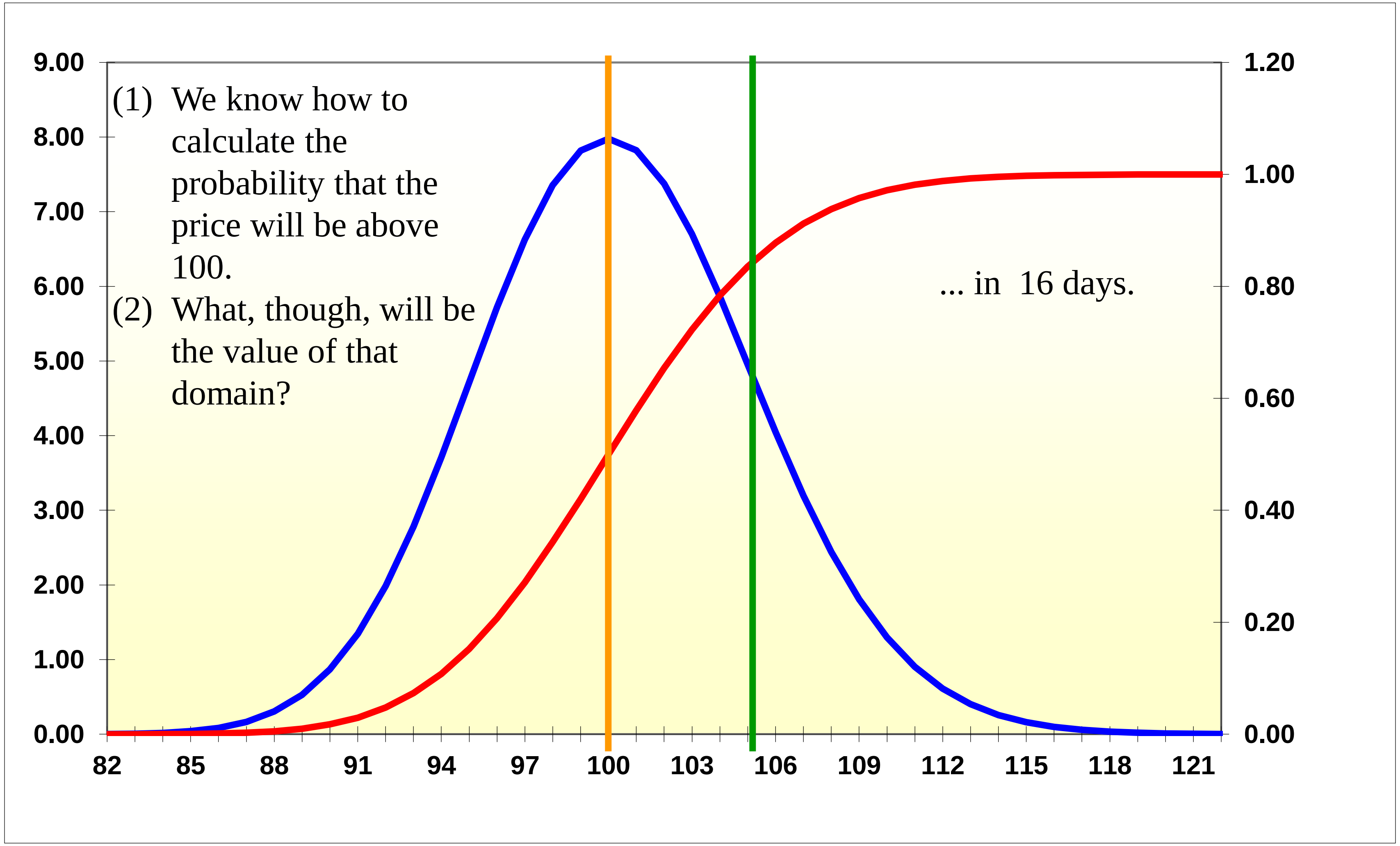


Modified from the original: slide 4 and 5 corrected for typos and slide 17, showing the models as methods, and slide 18, showing the model with our notation, added February 13, 2020 11:00 AM.

# The Black-Scholes-Merton Model

## … pricing options and calculating some Greeks

# Calculating what a 105 OTM call option should be worth if the present price of the stock is 100, expiry in 16 days, daily sigma at 0.025 ...



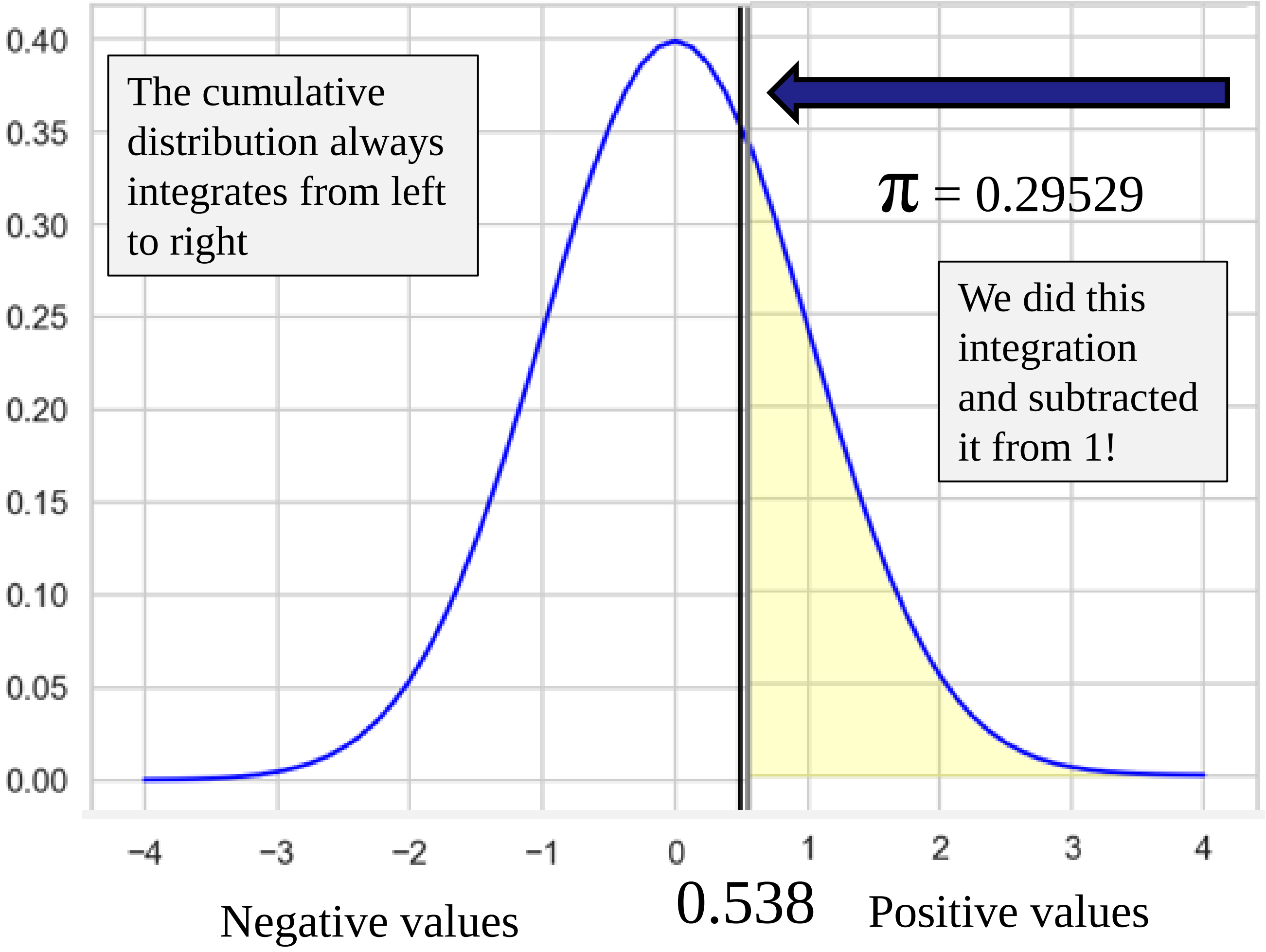
(1) We know how to calculate the probability that the price will be above 100.

(2) What, though, will be the value of that domain?

... in 16 days.

# Remembering from our homework ...

**Standard normal density function**



The cumulative distribution always integrates from left to right

π = 0.29529

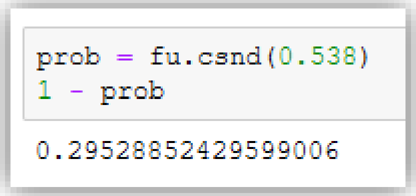
We did this integration and subtracted it from 1!

$$\frac{\ln\left(P_{str}/P_{sto}\right)}{\sigma\sqrt{t}} + \frac{\sigma\sqrt{t}}{2} = Z$$

..or

$$\frac{\ln\left(P_{str}/P_{sto}\right) + \left(\sigma^2/2\right)t}{\sigma\sqrt{t}} = Z$$

$$Z_{dv} = {}^{\ln\left({}^{105}/_{100}\right)}/_{\sigma\sqrt{16}} + {}^{\sigma\sqrt{16}}/_{2} = 0.538$$



```
prob = fu.csnd(0.538)
1 - prob
```

```
0.29528852429599006
```

Our original method ...

1. $P_{str} = P_{sto} e^{\left[\left(-\sigma^2/_2\right)t + \sigma\sqrt{t} \times Z\right]}$

2. $ln\left(^{P_{str}}/_{P_{sto}}\right) = \left(-\sigma^2/_2\right)t + \sigma\sqrt{t} \times Z$

3. $ln\left(^{P_{str}}/_{P_{sto}}\right) + \left(\sigma^2/_2\right)t = \sigma\sqrt{t} \times Z$

4. $\dfrac{ln\left(^{P_{str}}/_{P_{sto}}\right) + \left(\sigma^2/_2\right)t}{\sigma\sqrt{t}} = Z$

or ...

With our Z, we calced
1 – csnd(Z)

5. $\dfrac{ln\left(^{P_{str}}/_{P_{sto}}\right)}{\sigma\sqrt{t}} + \dfrac{\sigma\sqrt{t}}{2} = Z$

The BSM method ...

1. $ln(P_{str}) = ln(P_{sto}) - \left(\sigma^2/_2\right)t + \sigma\sqrt{t} \times Z$

2. $ln(P_{sto}) - ln(P_{str}) = \left(\sigma^2/_2\right)t - \sigma\sqrt{t} \times Z$

3. $ln\left(^{P_{sto}}/_{P_{str}}\right) = \left(\sigma^2/_2\right)t - \sigma\sqrt{t} \times Z$

$$\frac{ln\left(^{P_{sto}}/_{P_{str}}\right) - \left(\sigma^2/_2\right)t}{-\sigma\sqrt{t}} = Z$$

or ...

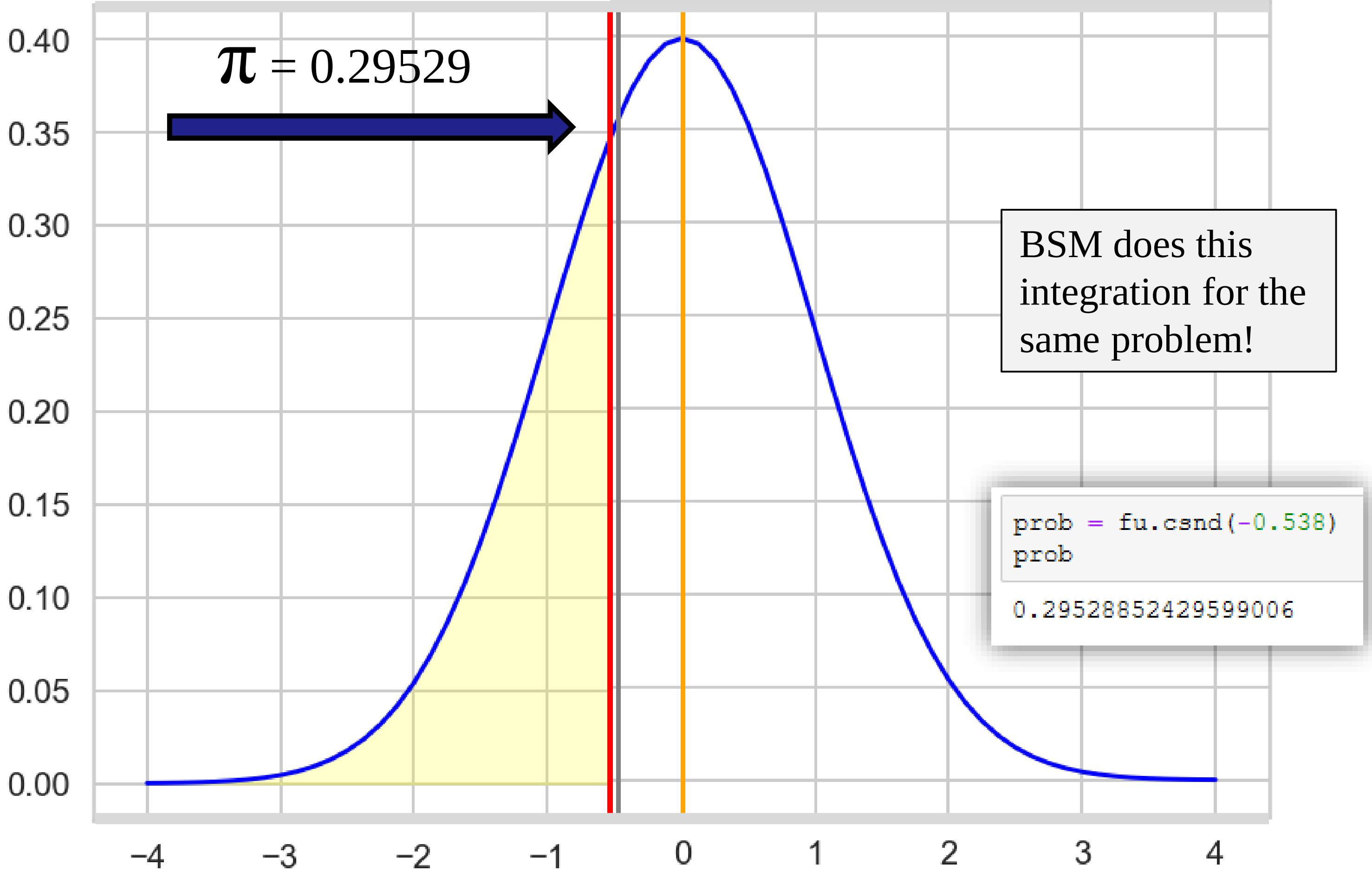
4. $\dfrac{ln\left(^{P_{sto}}/_{P_{str}}\right)}{-\sigma\sqrt{t}} + \dfrac{\sigma\sqrt{t}}{2} = Z$

or ...

5. $\dfrac{ln\left(^{P_{sto}}/_{P_{str}}\right) - \left(\sigma^2/_2\right)t}{\sigma\sqrt{t}} = -Z$

Remembering that
1 – csnd(Z) = csnd(-Z)

# ... and knowing this will come in handy when we look at Black-Scholes-Merton:



```
prob = fu.csnd(-0.538)
prob
```

```
0.29528852429599006
```

BSM does this integration for the same problem!

-0.538

This has a negative value when done this way (for an OTM option)..

$$\frac{\ln\left({}^{P_{sto}}/_{P_{str}}\right)}{\sigma\sqrt{t}} - \frac{\sigma\sqrt{t}}{2} = -Z$$

$$\frac{\ln\left({}^{P_{sto}}/_{P_{str}}\right)}{\sigma\sqrt{t}} - \frac{\sigma\sqrt{t}}{2} = -Z$$

$$Z_{dv} = {}^{\ln\left({}^{100}/_{105}\right)}/_{\sigma\sqrt{16}} - {}^{\sigma\sqrt{16}}/_{2} = -0.538$$

(-0.4879) - (0.005)

Notice here that the Ito adjustment is shifting the line to the left, slightly lowering the probability of ITM.

# How Black-Scholes-Merton works ...

The Black-Scholes-Merton model is used to price European options (which assumes that they must be held to expiration) and related custom derivatives. It takes into account that you have the option of investing in an asset earning the risk-free interest rate.

It acknowledges that the option price is purely a function of the volatility of the stock's price (the higher the volatility the higher the premium on the option).

Black-Scholes-Merton treats a call option as a forward contract to deliver stock at a contractual price, which is, of course, the strike price.

# The Essence of the Black-Scholes Approach

- Only volatility matters, the *mu* (drift) is not important.
- The option's premium will suffer from time decay as we approach expiration (Theta in the European model).
- The stock's underlying volatility contributes to the option's premium (Vega).
- The sensitivity of the option to a change in the stock's value (Delta) and the rate of that sensitivity (Gamma) is important [these variables are represented mathematically in the Black-Scholes DE].
- Option values arise from arbitrage opportunities in a world where you have a risk-free choice.

# The BSM Model: European Calls

$$C = SN(d_1) - Ke^{-r\left(t/365\right)} N(d_2)$$

C = theoretical call value
S = current stock price
N = standard normal probability distribution integrated to point ***d**_x*.
t = days until expiration
K = option strike price
r = risk free interest rate
$\sigma =$ daily stock volatility

$$d_1 = \frac{\ln\left(S/K\right) + \left(r/365 + \sigma^2/2\right)t}{\sigma\sqrt{t}}$$

$$d_2 = \frac{\ln\left(S/K\right) + \left(r/365 - \sigma^2/2\right)t}{\sigma\sqrt{t}}$$

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

**Note:** Hull's version (13.20) uses annual volatility. Note the difference.

# Breaking this down ...

$$C = SN(d_1) - Ke^{-r\left(t/365\right)}N(d_2)$$

This term discounts the price of the stock at which you will have the right to buy it (the strike price) back to its present value using the risk-free interest rate. Let's assume in the next slide that r = 0.

$$d_1 = \frac{\ln\left(S/K\right) + \left(r/365 + \sigma^2/2\right)t}{\sigma\sqrt{t}}$$

Dividing by this term (the standard deviation of stock's daily volatility adjusted for time) turns the distribution into a standard normal distribution with a standard deviation of 1.

# … or simplifying it some (r is zero)

$$CP = SP \times \Pi\left(d_1\right) - STR \times \Pi\left(d_2\right)$$

… assume that r is 0 and t is 1:

This is the absolute log growth spread between the strike price and the stock price.

The only difference is the sign …

$$d_1 = \frac{ln\left({}^{SP}/_{STR}\right) + \sigma^2/2}{\sigma}$$

$$d_2 = \frac{ln\left({}^{SP}/_{STR}\right) - \sigma^2/2}{\sigma}$$

μ is zero so this is the log-normal zero mean Ito-adjustment

**Note:** We already recognize that CSND($d_2$) is the probability that the call will be in ITM!

We are calculating the cumulative probability to this standard normal point.

This normalizes it to standard normal (the numerator is now “number of standard deviations.”).

**Note:** It turns out that this value CSND($d_1$) is the value of “delta”!

# An example (back to the full model) ...

Consider an otm option with 16 days to expiration. The strike price is 110 and the price of the stock is 100 and the stock has an daily volatility of 0.02. Assume an interest rate of 0.01 (1% annual).

black_scholes_merton_logical.ipynb

$$d_1 = \frac{ln(100/110) + (0.01/365 + 0.02^2/2)16}{0.02\sqrt{16}} = -1.1459$$

$$d_2 = d_1 - 0.02\sqrt{16} = -1.2259$$

$$C = 100csnd(-1.1459) - 110e^{-0.01\left({}^{16}/_{365}\right)}csnd(-1.2259) = 0.4841$$

# The BSM Model: European Puts

$$C = SN(d_1) - Ke^{-r\left({t}/{365}\right)}N(d_2)$$

(for comparison)

$$P = Ke^{-r\left({t}/{365}\right)}N(-d_2) - S \times N(-d_1)$$

C = theoretical call value
S = current stock price
N = standard normal probability distribution integrated to point ***d**$_x$*.
t = days until expiration
K = option strike price
r = risk free interest rate
$\sigma =$ daily stock volatility

$$d_1 = \frac{\ln\left(S/K\right) + \left(r/365 + \sigma^2/2\right)t}{\sigma\sqrt{t}}$$

$$d_2 = \frac{\ln\left(S/K\right) + \left(r/365 - \sigma^2/2\right)t}{\sigma\sqrt{t}}$$

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

$d_1$ and $d_2$ are calculated the same way they are calculated for calls.

**Note:** Hull's version (13.20) uses annual volatility. Note the difference.

# Using the BSM Model

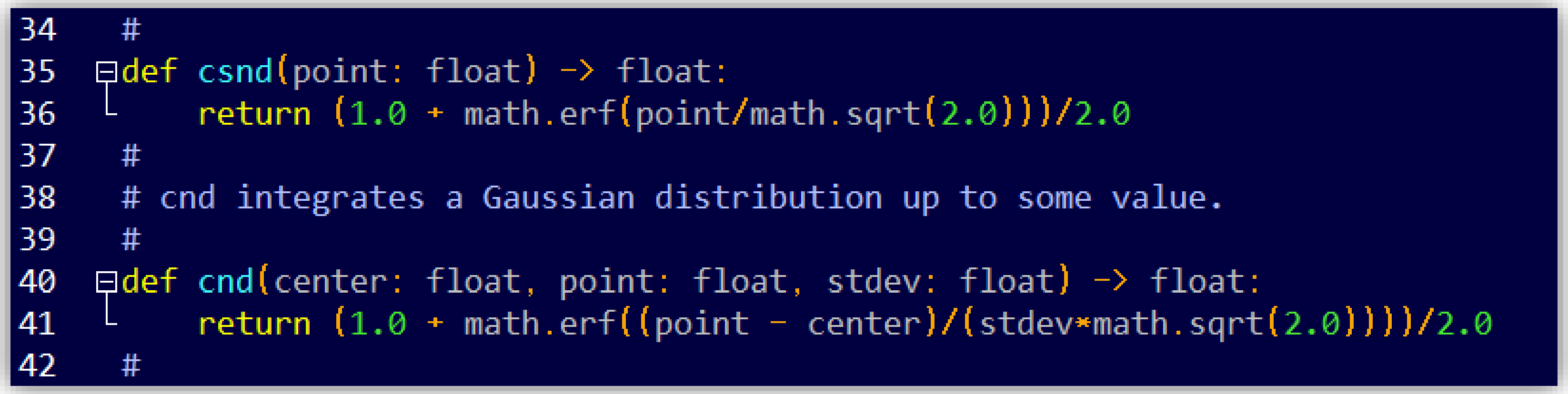
There are variations of the Black-Scholes model that prices for dividend payments (within the option period). See Hull section 13.12 to see how that is done (easy to understand). However, because of what is said below, you really can't use BSM to estimate values of options for dividend-paying American stocks

There is no easy estimator for American options prices, but as Hull points out in chapter 9 section 9.5, with the exception of exercising a call option just prior to an ex-dividend date, "it is never optimal to exercise an American call option on a non-dividend paying stock before the expiration date."

The BSM model can be used to estimate "***implied volatility***". To do this, however, given an actual option value, you have to iterate to find the volatility solution (see Hull's discussion of this in 13.12). This procedure is easy to program and not very time-consuming in even an Excel version of the model.

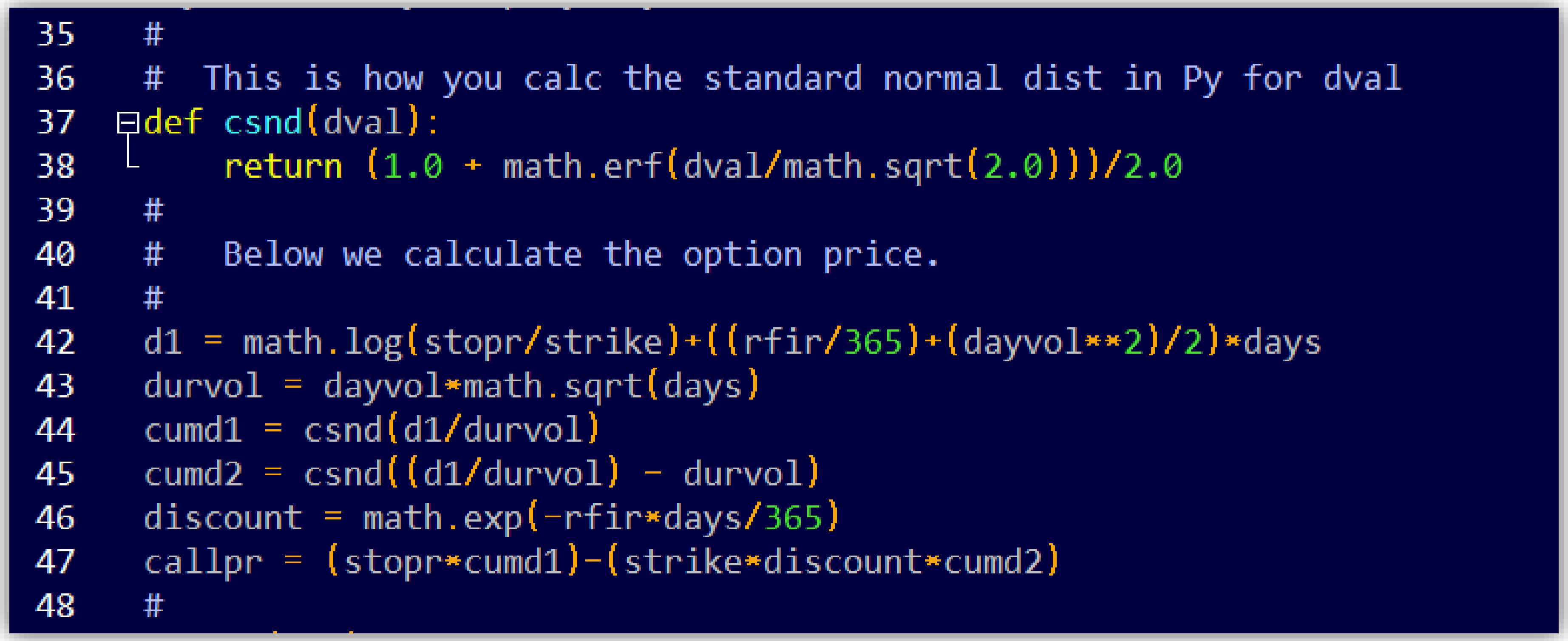
For those of you interest in another elegant implied volatility model, see Hull's discussion of the IVF model in 26.3. There you will see a role played by delta and vega, but again you would have to iterate to get the value of the sensitivity of the call to the strike price.

# (Footnote slide)



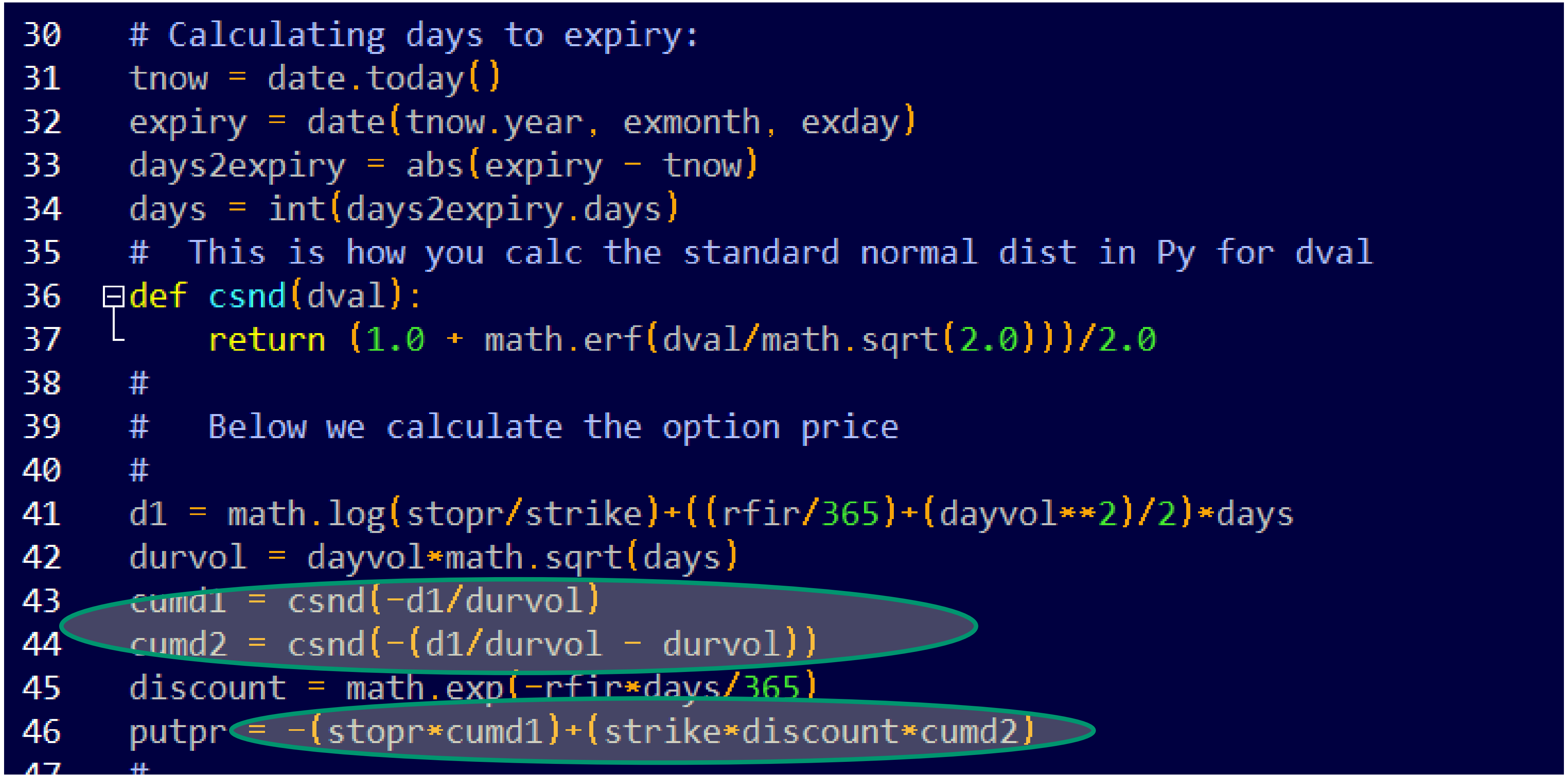
```
#
def csnd(point: float) -> float:
    return (1.0 + math.erf(point/math.sqrt(2.0)))/2.0
#
# cnd integrates a Gaussian distribution up to some value.
#
def cnd(center: float, point: float, stdev: float) -> float:
    return (1.0 + math.erf((point - center)/(stdev*math.sqrt(2.0))))/2.0
#
```

# Doing this in Python for a Call



```
#
#  This is how you calc the standard normal dist in Py for dval
def csnd(dval):
    return (1.0 + math.erf(dval/math.sqrt(2.0)))/2.0
#
#   Below we calculate the option price.
#
d1 = math.log(stopr/strike)+((rfir/365)+(dayvol**2)/2)*days
durvol = dayvol*math.sqrt(days)
cumd1 = csnd(d1/durvol)
cumd2 = csnd((d1/durvol) - durvol)
discount = math.exp(-rfir*days/365)
callpr = (stopr*cumd1)-(strike*discount*cumd2)
#
```

# Doing this in Python for a Put

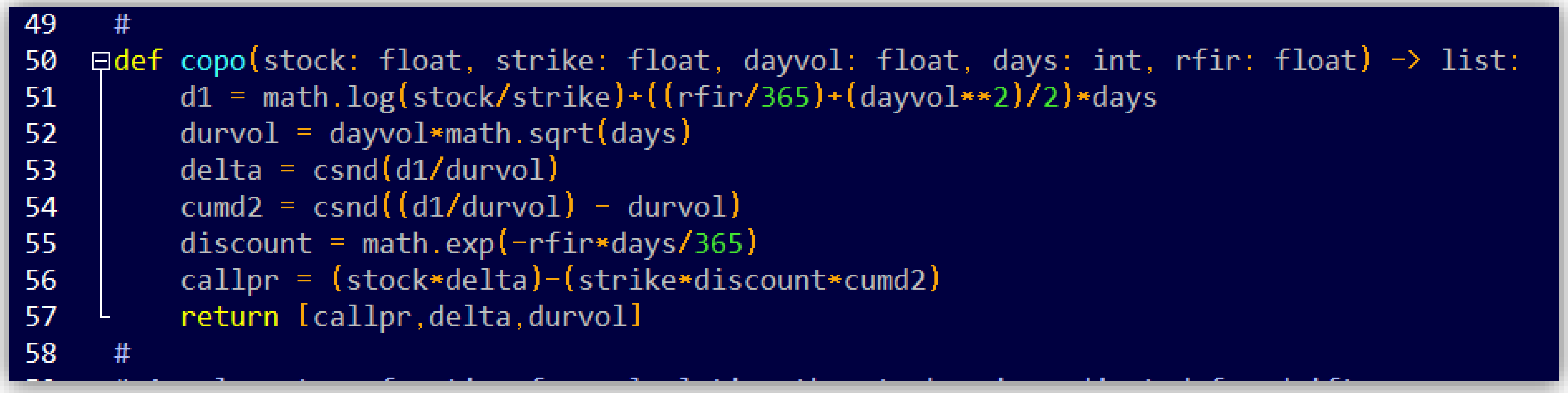


```
# Calculating days to expiry:
tnow = date.today()
expiry = date(tnow.year, exmonth, exday)
days2expiry = abs(expiry - tnow)
days = int(days2expiry.days)
#  This is how you calc the standard normal dist in Py for dval
def csnd(dval):
    return (1.0 + math.erf(dval/math.sqrt(2.0)))/2.0
#
#   Below we calculate the option price
#
d1 = math.log(stopr/strike)+((rfir/365)+(dayvol**2)/2)*days
durvol = dayvol*math.sqrt(days)
cumd1 = csnd(-d1/durvol)
cumd2 = csnd(-(d1/durvol - durvol))
discount = math.exp(-rfir*days/365)
putpr = -(stopr*cumd1)+(strike*discount*cumd2)
#
```

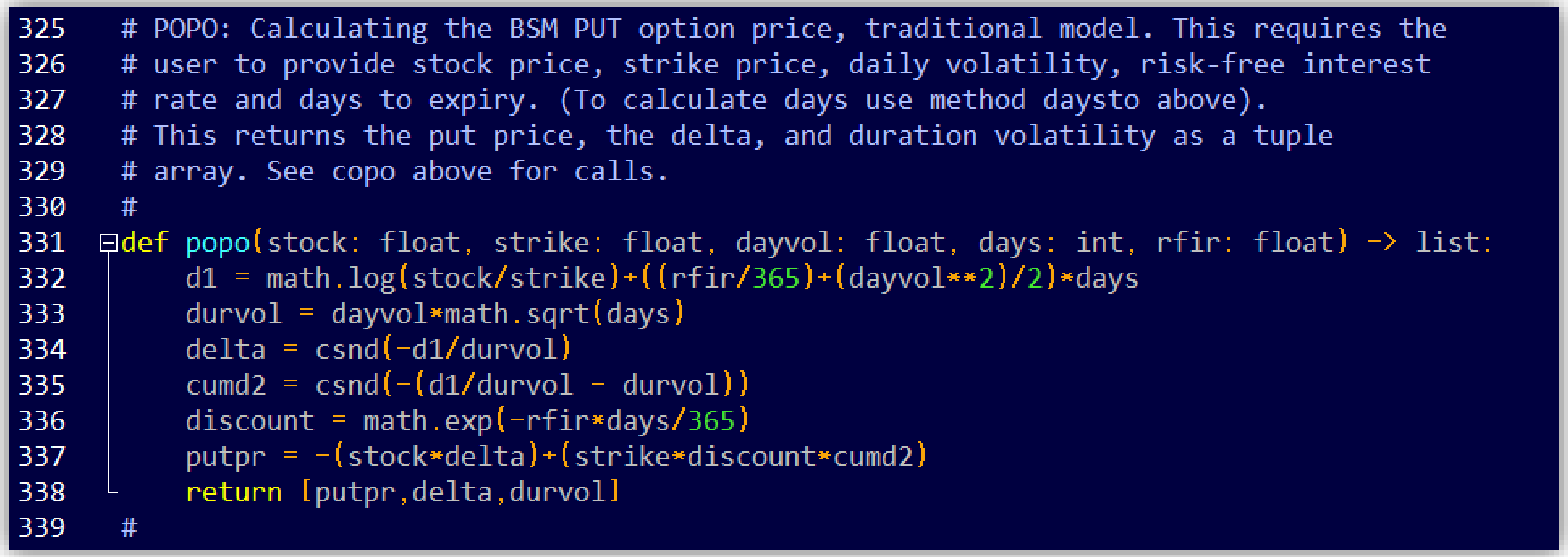
.. a couple of important sign changes.

# Doing the same using methods (for rapid repeat application):

This slide was added
to the original ...



```
#
def copo(stock: float, strike: float, dayvol: float, days: int, rfir: float) -> list:
    d1 = math.log(stock/strike)+((rfir/365)+(dayvol**2)/2)*days
    durvol = dayvol*math.sqrt(days)
    delta = csnd(d1/durvol)
    cumd2 = csnd((d1/durvol) - durvol)
    discount = math.exp(-rfir*days/365)
    callpr = (stock*delta)-(strike*discount*cumd2)
    return [callpr,delta,durvol]
#
```



```
# POPO: Calculating the BSM PUT option price, traditional model. This requires the
# user to provide stock price, strike price, daily volatility, risk-free interest
# rate and days to expiry. (To calculate days use method daysto above).
# This returns the put price, the delta, and duration volatility as a tuple
# array. See copo above for calls.
#
def popo(stock: float, strike: float, dayvol: float, days: int, rfir: float) -> list:
    d1 = math.log(stock/strike)+((rfir/365)+(dayvol**2)/2)*days
    durvol = dayvol*math.sqrt(days)
    delta = csnd(-d1/durvol)
    cumd2 = csnd(-(d1/durvol - durvol))
    discount = math.exp(-rfir*days/365)
    putpr = -(stock*delta)+(strike*discount*cumd2)
    return [putpr,delta,durvol]
#
```

# BSM Model – our notation

$$P_{call} = P_{sto} \times \pi(Z_0) - P_{str}e^{-r\left({}^{t}/_{365}\right)} \times \pi(Z_1)$$

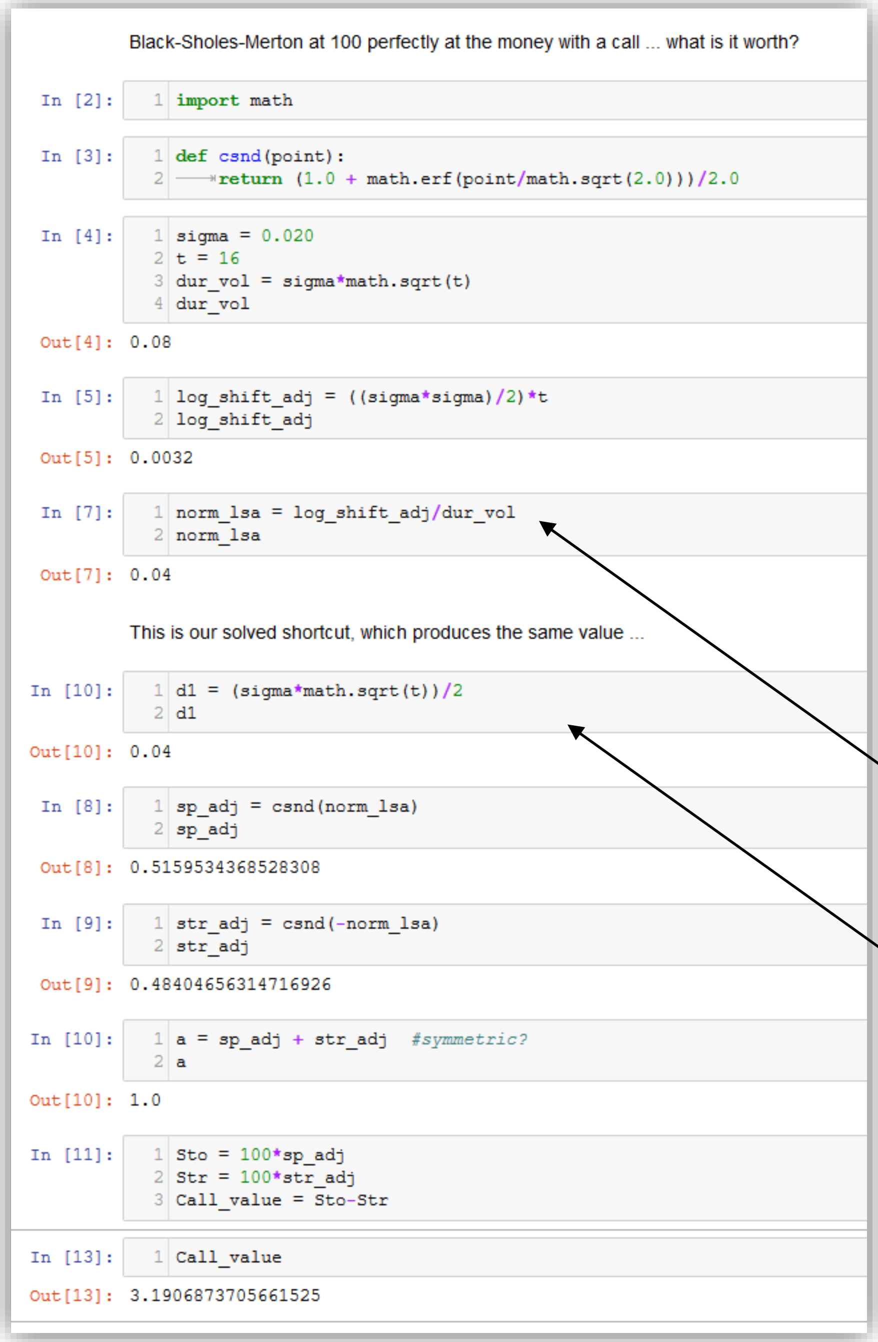
$$P_{put} = -P_{sto} \times \pi(-Z_0) + P_{str}e^{-r\left({}^{t}/_{365}\right)} \times \pi(-Z_1)$$

$$Z_0 = \frac{ln\left({}^{P_{sto}}/_{P_{str}}\right) + \left[\left({}^{r}/_{365}\right) + (\sigma^2/2)\right]t}{\sigma\sqrt{t}}$$

$$Z_1 = \frac{ln\left({}^{P_{sto}}/_{P_{str}}\right) - \left[\left({}^{r}/_{365}\right) + (\sigma^2/2)\right]t}{\sigma\sqrt{t}}$$

where probability..

$\pi(z) = csnd(z)$



Black-Sholes-Merton at 100 perfectly at the money with a call ... what is it worth?

```
In [2]:   import math

In [3]:   def csnd(point):
              return (1.0 + math.erf(point/math.sqrt(2.0)))/2.0

In [4]:   sigma = 0.020
          t = 16
          dur_vol = sigma*math.sqrt(t)
          dur_vol

Out[4]:   0.08

In [5]:   log_shift_adj = ((sigma*sigma)/2)*t
          log_shift_adj

Out[5]:   0.0032

In [7]:   norm_lsa = log_shift_adj/dur_vol
          norm_lsa

Out[7]:   0.04
```

This is our solved shortcut, which produces the same value ...

```
In [10]:  d1 = (sigma*math.sqrt(t))/2
          d1

Out[10]:  0.04

In [8]:   sp_adj = csnd(norm_lsa)
          sp_adj

Out[8]:   0.5159534368528308

In [9]:   str_adj = csnd(-norm_lsa)
          str_adj

Out[9]:   0.48404656314716926

In [10]:  a = sp_adj + str_adj   #symmetric?
          a

Out[10]:  1.0

In [11]:  Sto = 100*sp_adj
          Str = 100*str_adj
          Call_value = Sto-Str

In [13]:  Call_value

Out[13]:  3.1906873705661525
```

# What role is being played by half variance??

Let’s look at this simplified where we have a call at the money exactly, with these assumptions, assuming r = 0:

1. $$d_1 = \frac{\ln\left({}^{Sto}/_{Str}\right) + {}^{\sigma^2}/_{2}\, t}{\sigma\sqrt{t}}$$

2. $$\ln\left({}^{100}/_{100}\right) = 0$$

3. $$d_1 = \frac{{}^{\sigma^2}/_{2}\, t}{\sigma\sqrt{t}}$$

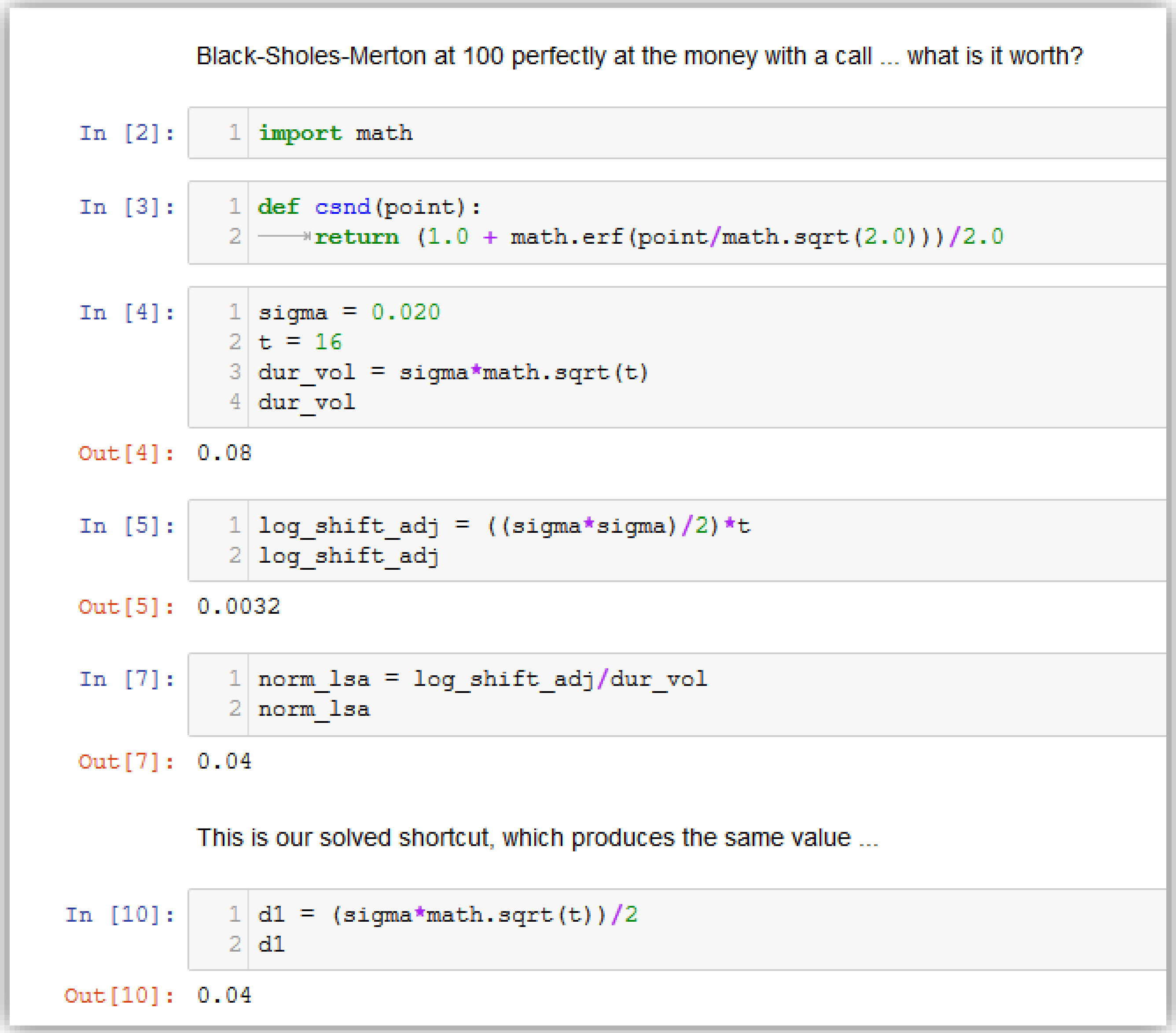
4. $$d_1 = \frac{\sigma\sqrt{t}}{2}$$

Stock price = 100

Strike price = ~~110~~ 100

Sigma = 0.020

days = 16



Black-Sholes-Merton at 100 perfectly at the money with a call ... what is it worth?

```
In [2]: import math
```

```
In [3]: def csnd(point):
            return (1.0 + math.erf(point/math.sqrt(2.0)))/2.0
```

```
In [4]: sigma = 0.020
        t = 16
        dur_vol = sigma*math.sqrt(t)
        dur_vol

Out[4]: 0.08
```

```
In [5]: log_shift_adj = ((sigma*sigma)/2)*t
        log_shift_adj

Out[5]: 0.0032
```

```
In [7]: norm_lsa = log_shift_adj/dur_vol
        norm_lsa

Out[7]: 0.04
```

This is our solved shortcut, which produces the same value ...

```
In [10]: d1 = (sigma*math.sqrt(t))/2
         d1

Out[10]: 0.04
```

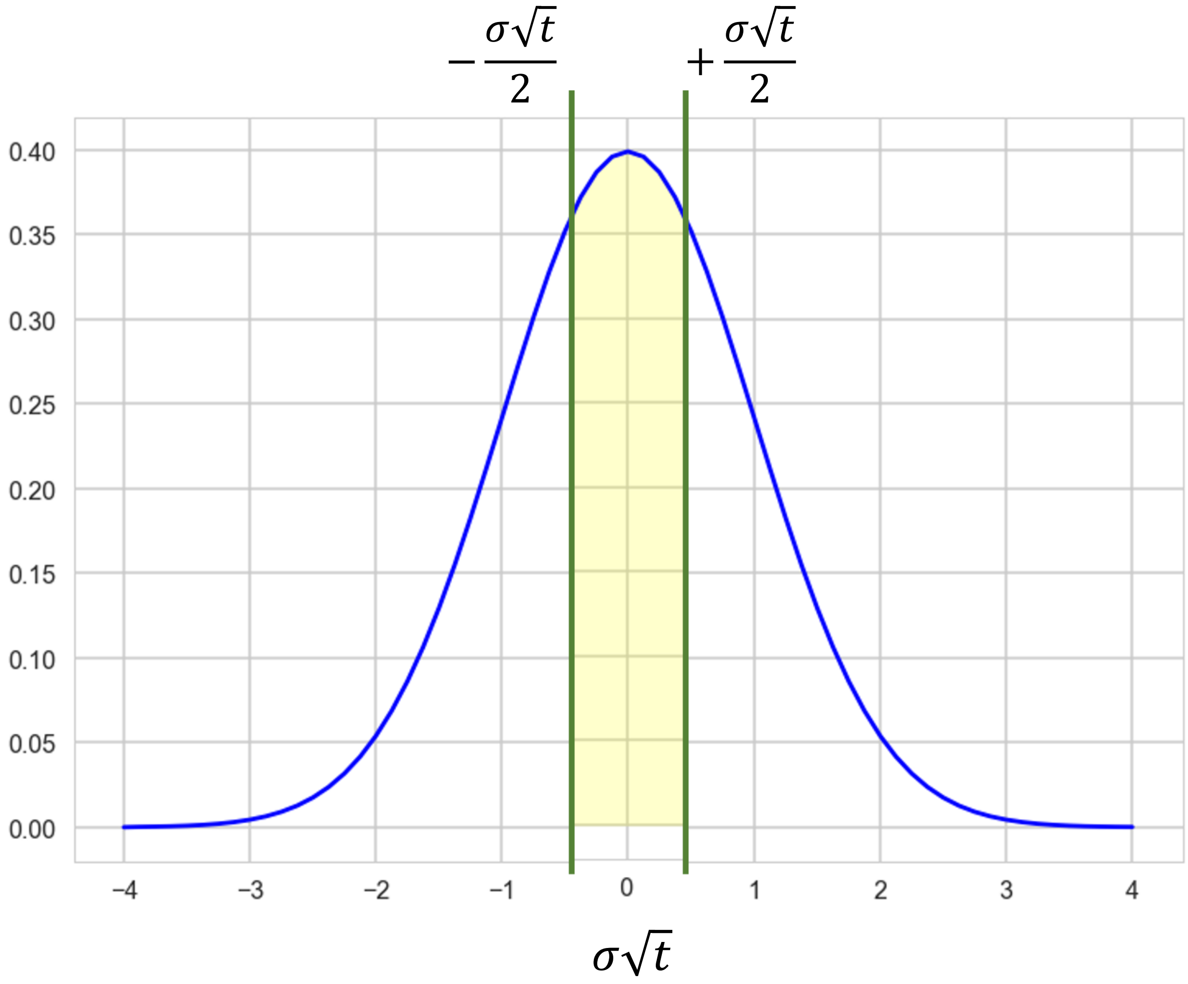
In this example, we are pricing a call that is expiring in 16 days.

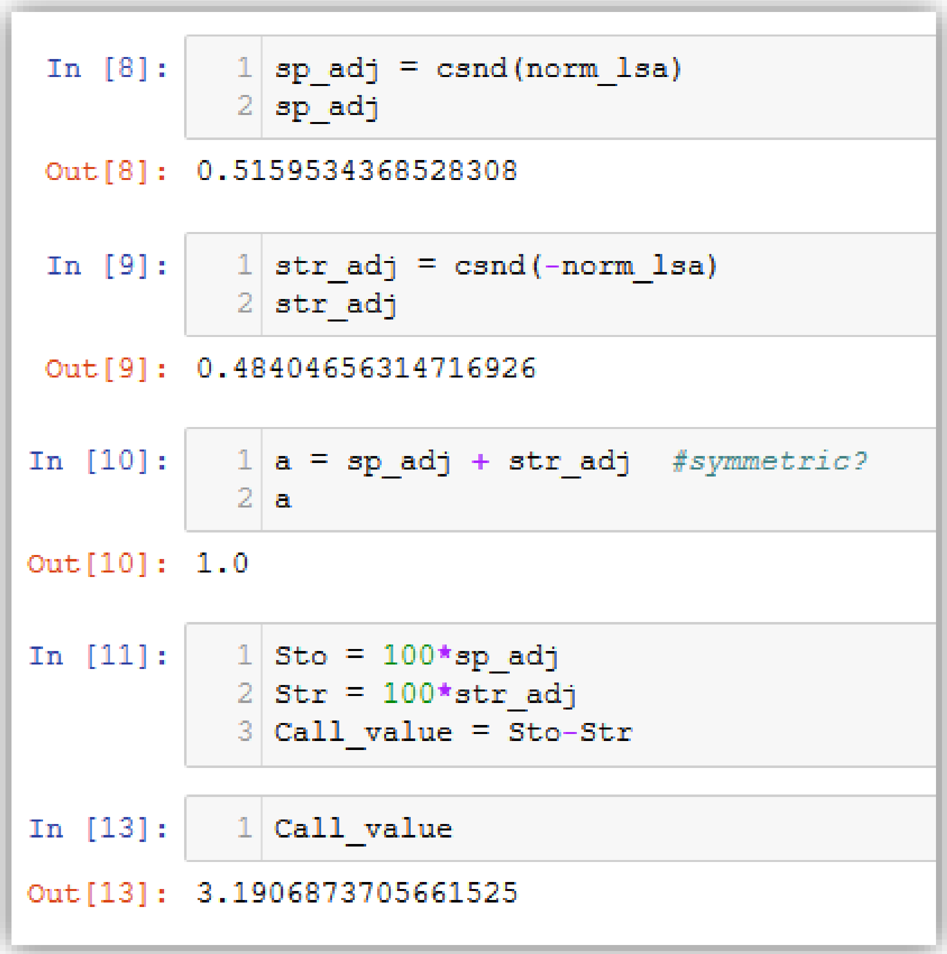
The price of the stock is $100 and the call strike price is $100.

Because BSM does not allow drift, the expected value of the stock in 16 days is $100.

If you own this call, you have the right to buy the stock for $100 in 16 days, no matter what the price.

Why does this have any value at all?





```
In [8]:   sp_adj = csnd(norm_lsa)
          sp_adj

Out[8]:   0.5159534368528308

In [9]:   str_adj = csnd(-norm_lsa)
          str_adj

Out[9]:   0.48404656314716926

In [10]:  a = sp_adj + str_adj   #symmetric?
          a

Out[10]:  1.0

In [11]:  Sto = 100*sp_adj
          Str = 100*str_adj
          Call_value = Sto-Str

In [13]:  Call_value

Out[13]:  3.190687370566l525
```

This spread is what gives the ATM option value ... and it is equal to the duration-volatility adjusted random draw.