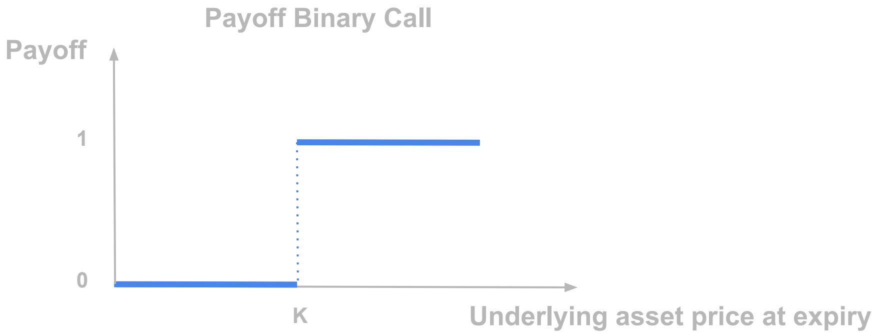
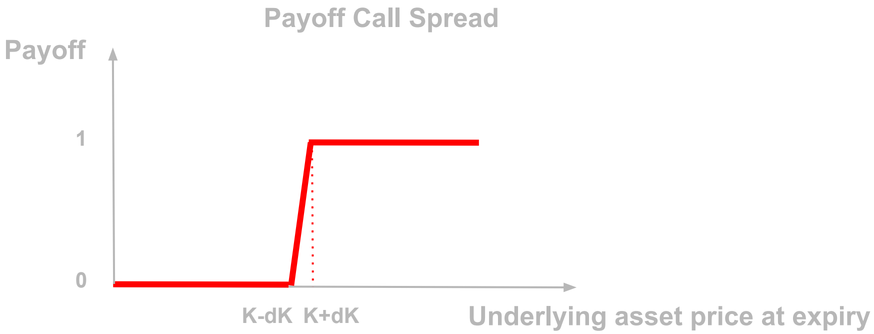
Binary Options: Replication and Skew Sensitivity

Binary Payoff and Replication

A binary (or digital) cash-or-nothing call option has the following payoff, with K the strike price and S the underlying asset:



This payoff can be replicated by a call spread with strikes K1 = K - dK and K2 = K + dK, and a quantity = 1 / (2 x dK) with dK small enough



Payoff binary call option

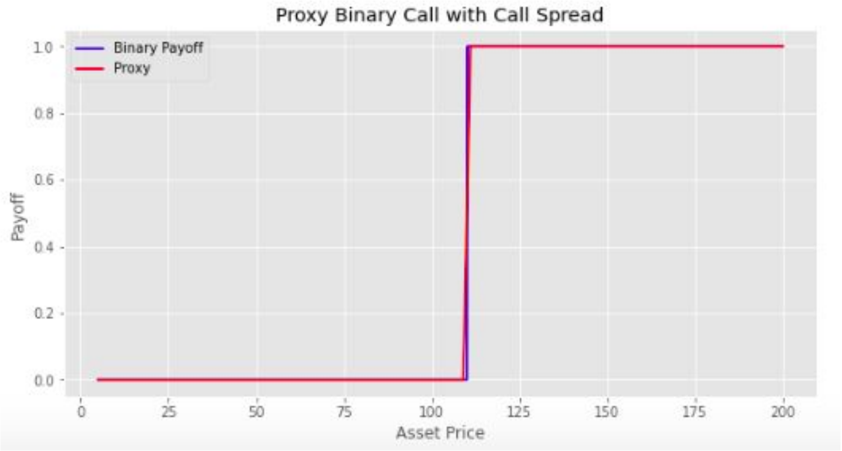
$$\begin{cases} 0 & \text{if} \quad S < K \\ 1 & \text{if} \quad S \geq K \end{cases}$$

$$\text{Binary Call} = \lim_{dK \to 0} \frac{1}{2 \cdot dK} \cdot \left(Call(K - dK) - Call(K + dK)\right)$$

Binary Payoff and Replication

```python
def PayoffCallSpread(S, K1, K2):
    call1 = np.maximum(S - K1, 0)
    call2 = np.maximum(S - K2, 0)
    return call1 - call2
```

```python
K = 110

S = np.arange(5,200,.1)
dK = 1

PayOffBinary = [0 if x < K else 1 for x in S]
PayOffProxy = [1 / (2 * dK) * PayoffCallSpread(x , K - dK, K + dK) for x in S]

plt.rcParams["figure.figsize"] = (10,5)
plt.plot(S,PayOffBinary, color = 'b')
plt.plot(S, PayOffProxy, color = 'r')
plt.ylabel('Payoff')
plt.xlabel('Asset Price')
plt.legend(['Binary Payoff', 'Proxy'])
plt.title('Proxy Binary Call with Call Spread')
plt.show;
```



Binary Call Price in Black-Scholes Model

If we assume that the only changing variable is the strike price, the price of a binary call is equal to the opposite of the derivatives of the call price with respect to the strike price:

$$\text{Binary Call} = \lim_{dK \to 0} \frac{1}{2 \cdot dK} \cdot (Call(K - dK) - Call(K + dK))$$

$$C_{\text{binary}} = -\frac{\partial Call}{\partial K}$$

It can be proxy by finite difference method:

$$C_{\text{binary}} \approx \frac{Call(S,K-\delta K,\sigma,r,T) - Call(S,K+\delta K,\sigma,r,T)}{2 \cdot \delta K}$$

In the Black-Scholes framework there is a closed form solution for the price of a call option:

$$Call = S \cdot N(d_1) - K \cdot e^{-r \cdot T} \cdot N(d_2)$$

$$\text{with } d_2 = \frac{\ln\left(\frac{S_t}{K}\right) - \left(r - \frac{\sigma^2}{2}\right) \cdot (T-t)}{\sigma \cdot \sqrt{T-t}} \qquad d_1 = d_2 + \sigma \cdot \sqrt{T}$$

$$N(x) = \frac{1}{\sqrt{2 \cdot \pi}} \cdot \int_{-\infty}^{x} e^{-\frac{u^2}{2}} du$$

The price of the binary call is equal to the probability that the option will be exercised times the discount factor:

$$C_{\text{binary}} = e^{-r \cdot T} \cdot N(d_2)$$

Binary Call Price in Black-Scholes Model

```
def CallPrice(S, K, r, sigma, T):
    d1 = (math.log(S / K) + (r + .5 * sigma**2) * T) / (sigma * T**.5)
    d2 = d1 - sigma * T**.5
    n1 = norm.cdf(d1)
    n2 = norm.cdf(d2)
    DF = math.exp(-r * T)
    price=S * n1 - K * DF * n2
    return price

def PutPrice(S, K, r, sigma, T):
    d1 = (math.log(S / K) + (r + .5 * sigma**2) * T) / (sigma * T**.5)
    d2 = d1 - sigma * T**.5
    n1 = norm.cdf(-d1)
    n2 = norm.cdf(-d2)
    DF = math.exp(-r * T)
    price= K * DF * n2 - S * n1
    return price

def DigitalCallPriceBS(S, K, r, sigma, T):
    d1 = (math.log(S / K) + (r + .5 * sigma**2) * T) / (sigma * T**.5)
    d2 = d1 - sigma * T**.5
    n2 = norm.cdf(d2)
    DF = math.exp(-r * T)
    price= DF * n2
    return price

def DigitalPutPriceBS(S, K, r, sigma, T):
    d1 = (math.log(S / K) + (r + .5 * sigma**2) * T) / (sigma * T**.5)
    d2 = d1 - sigma * T**.5
    n2 = norm.cdf(-d2)
    DF = math.exp(-r * T)
    price= DF * n2
    return price
```

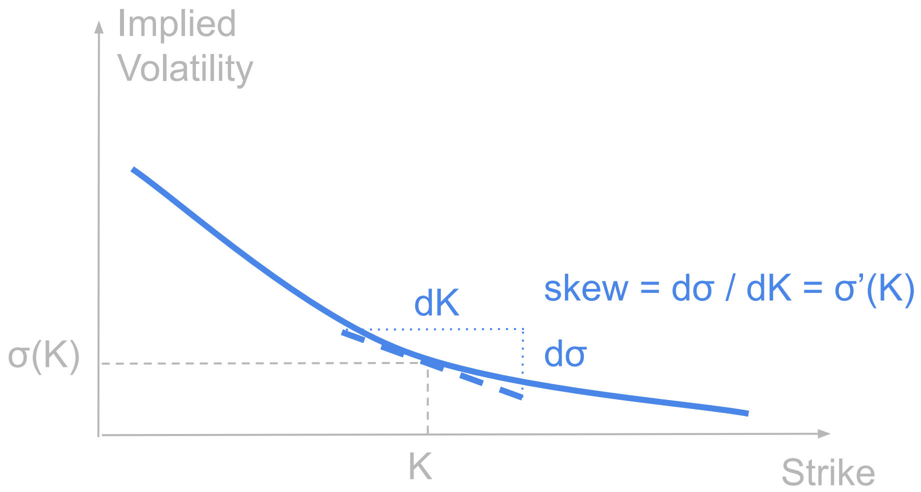
```
#parameters
S0 = 100 #asset price
K = 110 #strike price
r= .05 #risk-free interest rate
sigma = .3 #volatility
T = .25 #time to maturity
dK = 0.1

pricedigital = DigitalCallPriceBS(S0, K, r, sigma, T)
priceproxy = 1/(2 * dK) * (CallPrice(S0, K - dK, r, sigma, T) - \
                           CallPrice(S0, K + dK, r, sigma, T))
print("Binary Call Price:" + str(np.round(pricedigital, 3)))
print("Proxy Call Spread:" + str(np.round(priceproxy, 3)))
```

```
Binary Call Price:0.262
Proxy Call Spread:0.262
```

What if the implied volatility is not constant?

Skew Sensitivity of a Binary Call Option

We assume that the implied volatility is no more constant, it is a function of the strike price. σ(K) is the implied volatility of a call option at the strike K.



$$Call(K) = Call_{BS}(K, \sigma(K))$$

$$C_{\text{binary}} = -\frac{\partial Call}{\partial K}$$

$$C_{\text{binary}} = \underbrace{-\frac{\partial Call_{BS}}{\partial K}}_{\text{Black-Scholes price}} \underbrace{-\frac{\partial Call_{BS}}{\partial \sigma}} \cdot \sigma'(K)$$

Black-Scholes price

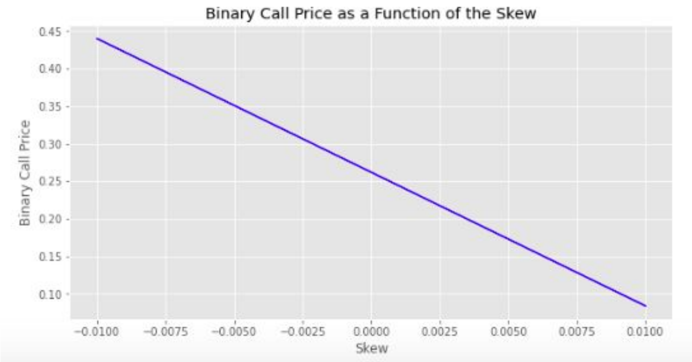
$$e^{-r \cdot T} \cdot N(d_2)$$

$$\nu = S \cdot e^{-\frac{d_1^2}{2}} \cdot \sqrt{\frac{T}{2 \cdot \pi}}$$

> 0 when $\sigma'(K) < 0$

Skew Sensitivity of a Binary Call Option

```
#parameters
S0 = 100 #asset price
K = 110 #strike price
r= .05 #risk-free interest rate
sigma = .3 #volatility
T = .25 #time to maturity
skew = np.arange(-.01, .011, .001)

PriceBS = DigitalCallPriceBS(S0, K, r, sigma, T)
d1 = (math.log(S0 / K) + (r + .5 * sigma**2) * T) / (sigma * T**.5)
vega = S0 * math.exp(- .5 * d1**2) * (T / (2 * np.pi))**.5

priceproxy = [PriceBS - vega * s for s in skew]
plt.rcParams["figure.figsize"] = (10,5)
plt.plot(skew, priceproxy, color = 'b')
plt.ylabel('Binary Call Price')
plt.xlabel('Skew')
plt.title('Binary Call Price as a Function of the Skew')
plt.show;
```



$$Call(K) = Call_{BS}(K, \sigma(K))$$

$$C_{\text{binary}} = -\frac{\partial Call}{\partial K}$$

$$C_{\text{binary}} = \underbrace{-\frac{\partial Call_{BS}}{\partial K}}_{\text{Black-Scholes price}} \underbrace{-\frac{\partial Call_{BS}}{\partial \sigma}}_{} \cdot \sigma'(K)$$

Black-Scholes price

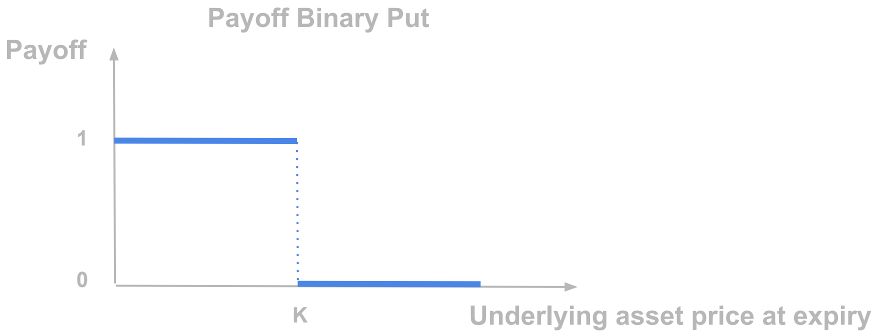
$$e^{-r \cdot T} \cdot N(d_2)$$

$$\nu = S \cdot e^{-\frac{d_1^2}{2}} \cdot \sqrt{\frac{T}{2 \cdot \pi}}$$

> 0 when σ’(K) < 0

Skew Sensitivity of a Binary Put Option

A binary (or digital) cash-or-nothing put option has the following payoff, with K the strike price and S the underlying asset:



And it can be replicated by a put spread:

$$\text{Binary Put} = \lim_{dK \to 0} \frac{1}{2 \cdot dK} \cdot (Put(K + dK) - Put(K - dK))$$

$$P_{\text{binary}} = \frac{\partial P}{\partial K}$$

$$Put(K) = Put_{BS}(K, \sigma(K))$$

$$P_{\text{binary}} = \frac{\partial P}{\partial K}$$

$$P_{\text{binary}} = \underbrace{\frac{\partial Put_{BS}}{\partial K}}_{\text{Black-Scholes price}} + \underbrace{\frac{\partial Put_{BS}}{\partial \sigma}}_{\nu} \cdot \sigma'(K)$$

Black-Scholes price

$$e^{-r \cdot T} \cdot N(-d_2)$$

$$\nu = S \cdot e^{-\frac{d_1^2}{2}} \cdot \sqrt{\frac{T}{2 \cdot \pi}}$$

< 0 when σ'(K) < 0

Contact Us

website: www.quant-next.com

email: contact@quant-next.com

Follow us on LinkedIn

Disclaimer

This document is for educational and information purposes only.

It does not intend to be and does not constitute financial advice, investment advice, trading advice or any other advice, recommendation or promotion of any particular investments.