Assignment 2 Doctoral Seminar

Joseph P. Janko

Read and consult the paper by Berk, Green, and Naik, titled 'Optimal Investment, Growth Options, and Security Returns' in the Journal of Finance 1999. The main idea of the paper is that firms consist of existing assets and growth options. Study and analyze equations (11) to (13), and eventually equation (32). To this end, form a model with binomial trees (say 3-period model but a more general k-period model works as well) to mimic a similar problem to Berk, Green, and Naik

Overview

In departure from traditional empirical asset pricing methodology, where market returns are found to be related to book-to-market value, Berk, Green, and Naik in their 1999 paper "Optimal Investment, Growth Options, and Security Returns" replicate the results with a dynamic model using a valuation methodology of growth investment options and current existing cash flow generating assets adjusted for systematic risk. This type of approach provides the opportunity to explain cross-sectional variation in expected returns through fluctuations in firm-specific variables. Thereby, B/M is a state variable that represents the firm's risk to its asset base, while asset turnover represents the relative value of potential growth options to existing cash flows. Furthermore, the authors show when combining the firm's systematic risks and dynamic turnover of a firm's assets that the sum of the NPVs of these options can be equated to a bond. This assignment aims to examine equations 11 to 32 and form a k-period binomial model to mimic a similar methodology in the above-mentioned paper. The following contents of this paper will give an overview of Berk's equations, a derivation of the binomial model of interest and implementation of the Berk, Green, and Naik methodology for the valuation of a firm.

The Model

The valuation model in the paper of interest is derived in an infinite horizon time scale and discrete-time where the investment of a project to be undertaken must be done immediately. Let I be the investment required to undertake a project at date t. The cash flows from a project that was undertaken at date j<t are given by the following equation:

$$C_j(t) = I\exp(\bar{C} - .5 * \sigma_j^2 + \sigma_j * \epsilon_j(t)$$

Furthermore, to account for the probability that a project may be obsolescent, we add indicator values

$$x_j(t+1) = x_j(t) * Y_j(t+1)$$

Where $Y_j(t+1)$ equals one with probability Π and zero with 1-Π

The pricing kernel is defined below:

$$\frac{z(t+1)}{z(t)} = \exp[-r(t) - .5 * \sigma_z{}^2 - \sigma_z * v(t+1)$$

Where the dynamics of r(t) follow the Vasicek model:

$$r(t+1) = \kappa r(t) + (1-\kappa)\bar{r} + \sigma_r \varepsilon(t+1)$$

The "systematic risk" of a project's cash flow is given by its beta

$$\beta_j = \sigma_j \sigma_z cov(\varepsilon_j(t), v(t))$$

Higher beta is indicative of cash flows being negatively correlated with cash flows of the pricing kernel

Thus, the value of the security is

$$E_t\{\frac{z(t+1)}{z} * C_j(t+1)\} = I\exp\{\bar{C} - \beta_j - r(t)\}$$

The expression for the value of the cash flow from a given project is as follows:

$$V_j = E_t\left\{\sum_{s=t+1}^{\infty} \frac{z(s)}{z(t)} C_j(s) x_j(s)\right\}$$

To calculate the value of options to invest in projects in the future, the value of the option the maximum of the NPV of the project or zero. Thus,

$$V^*(t) = E_t\left\{\sum_{s=t+1}^{\infty} \frac{z(s)}{z(t)} \max[V_s(s) - I, 0]\right\}$$

Finally, P (t) denotes the value of the firm. It is equal to the value of the future cash flows plus the value of future investment opportunities.

$$P(t) = \sum_{j=0}^{t} V_j(t) X_j(t) + V^*(t)$$

Deriving the Binomial Model

P1: One European Call Option and cash equaling to value Ke^{-rt}

P2: One European Put Option and one share of the underlying stock

At t=0: You have C + Ke^{-rt} and P + S(0)

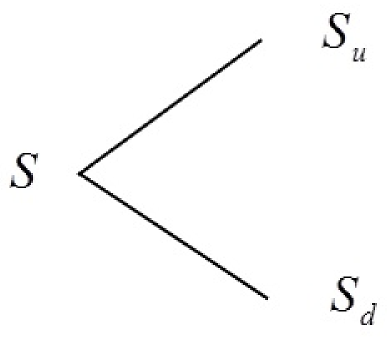
At t=T where S(T) > K: The portfolio's payoff is equal to [S(T) – K] + K = 0 + S(T)

At t=T where S(T) < K: The portfolio's payoff is equal to [0] + K = [K-S(T)] + S(T)

Thus, the put-call parity relationship has been proven

Now let one consider a one-step binomial model

Figure 1



*u-up and d-down

Consider a single time step of δt

Consider a riskless bond B(t=0) with value 1. In time dt we have B(t=1) = $e^{rdt * r}$

Now consider adding an underlying asset to the portfolio S.

Let Π represent the portfolio, δ represent the number of shares of S, and λ represent the number of bonds

Thus at t=0

$$\Pi_0 = \delta S_0 + \lambda * 1$$

At t= dt

$$v_u = \delta S_0 u + \lambda * e^{rdt}$$

$$v_d = \delta S_0 d + \lambda * e^{rdt}$$

Where v_u and v_d are the values of the option in the up and down state

Solve the system of the equations for Δ

$$v_u - v_d = \delta S_0 (u-d)$$

$$\delta = \frac{v_u - v_d}{S_0(u-d)}$$

$$v_d = \frac{v_u - v_d}{S_0(u-d)} S_0 d + \lambda * e^{rdt}$$

$$v_d = \frac{v_u - v_d}{(u-d)} d + \lambda * e^{rdt}$$

$$v_d - \frac{v_u - v_d}{(u-d)} d = \lambda * e^{rdt}$$

$$\frac{(u-d)v_d}{(u-d)} - \frac{v_u - v_d}{(u-d)} d = \lambda * e^{rdt}$$

$$\lambda = \ e^{-rdt} * \frac{u * \ v_d - d * v_u}{u - d}$$

Find the value of the option at t=0 buy substitution of Δ and Ψ

$$v_0 = \frac{v_u - \ v_d}{S_0(u - d)} * S_0 + \ \ e^{-rdt} * \frac{u * \ v_d - d * v_u}{u - d} * 1$$

$$v_0 = \frac{v_u - \ v_d}{(u - d)} + \ \ e^{-rdt} * \frac{u * \ v_d - d * v_u}{u - d}$$

$$v_0 = \frac{v_u - \ v_d + e^{-rdt}(u * \ v_d - d * v_u)}{u - d}$$

$$v_0 = \frac{v_u - \ v_d + e^{-rdt} * u * \ v_d - e^{-rdt} * d * v_u)}{u - d}$$

$$v_0 = e^{-rdt} * \{ \frac{e^{rdt} * v_u - \ e^{rdt} * v_d + u * \ v_d - d * v_u)}{u - d} \}$$

$$v_0 = \ e^{-rdt} * \{ \frac{e^{rdt} - d}{u - d} * \ v_u + \ \ \frac{u - e^{rdt}}{u - d} * \ v_d \}$$

Let π_u and π_d be defined as the following

$$\pi_u = \frac{e^{rdt} - d}{u - d}$$

$$\pi_d = \frac{u - e^{rdt}}{u - d}$$

$$v_0 = \ e^{-rdt} * \{\pi_u \ * \ f_u + \ \ \pi_d * \ f_d\}$$

$$\pi_u + \pi_d = 1$$

$$E[S_{t+1}] = \ S_0 e^{rdt}$$

Where π represents the risk neutral probability

Now consider a three-step model

Let u = 1/d to allow for recombination, which implies S_0*u*d = S_0

Figure 2

$$\begin{tikzcd}
 & & & S_0u^3 \\
 & & S_0u^2 \arrow[ur] \arrow[dr] & \\
 & S_0u \arrow[ur] \arrow[dr] & & S_0u \\
S_0 \arrow[ur] \arrow[dr] & & S_0 \arrow[ur] \arrow[dr] & \\
 & S_0d \arrow[ur] \arrow[dr] & & S_0d \\
 & & S_0d^2 \arrow[ur] \arrow[dr] & \\
 & & & S_0d^3
\end{tikzcd}$$

Assume a risk-neutral process to calibrate the binomial process

$$dS = rSdt + \sigma SdB$$

$$\frac{dS}{S} = rdt + \sigma dB$$

We assume log normal distribution

$$\ln\left(\frac{S_{t+dt}}{S_t}\right) \sim N(r - .5\sigma^2, \sigma^2 dt)$$

We know the moments of the lognormal distribution

$$E\left[\frac{S_{t+dt}}{S_t}\right] = e^{rdt}$$

$$Var\left[\frac{S_{t+dt}}{S_t}\right] = e^{2rdt}(e^{2rdt} - 1)$$

$$S_t * e^{rdt} = P_u * S_t * u + (1 - P_u) * S_t * d = E[S_{t+dt}]$$

$$P_u = \frac{e^{rdt} - d}{u - d}$$

$$P_d = 1 - \frac{e^{rdt} - d}{u - d} = \frac{u - e^{rdt}}{u - d}$$

$$Var(S_{t+dt}) = E\left[{S_{t+dt}}^2\right] - E[S_{t+dt}]^2$$

$$Var(S_{t+dt}) = (pu^2 + (1-p)d^2)S_t^2 - S_t^2 * e^{2rdt}$$

$$Var(S_{t+dt}) = S_t^2 * [(pu^2 + (1-p)d^2) - e^{2rdt}]$$

Use the known formula

$$Var(S_{t+dt}) = S_t^2 * e^{2rdt}\,(e^{\sigma^2 dt} - 1)$$

$$S_t^2 * [(pu^2 + (1-p)d^2) - e^{2rdt}] = e^{2rdt}\,(e^{\sigma^2 dt} - 1)$$

$$(pu^2 + (1-p)d^2) - e^{2rdt} = e^{2rdt}\,(e^{\sigma^2 dt} - 1)$$

$$(pu^2 + (1-p)d^2) = e^{2rdt}\,(e^{\sigma^2 dt} - 1) + e^{2rdt}$$

$$(pu^2 + (1-p)d^2) = e^{2rdt + \sigma^2 dt}$$

$$\left(\frac{e^{rdt} - d}{u-d}u^2 + \frac{u - e^{rdt}}{u-d}d^2\right) = e^{2rdt + \sigma^2 dt}$$

$$(u+d)e^{rdt} - 1 = e^{2rdt + \sigma^2 dt}$$

$$ue^{rdt} + (\frac{1}{u})e^{rdt} - 1 = e^{2rdt + \sigma^2 dt}$$

$$u^2\,e^{rdt} + e^{rdt} - u = ue^{2rdt + \sigma^2 dt}$$

$$u^2\,e^{rdt} - u(e^{2rdt + \sigma^2 dt} + 1) + e^{rdt} = 0$$

Solve the quadratic equation

$$u = \frac{(e^{2r\delta t + \sigma^2 \delta t} + 1) \pm \sqrt{(e^{2r\delta t + \sigma^2 \delta t} + 1)^2 - 4\,e^{r\delta t}e^{r\delta t}}}{2\,e^{r\delta t}}$$

$$e^{2rdt + \sigma^2 dt} = 2 + (2r + \sigma^2)dt$$

$$u = \left(2 + (2r + \sigma^2)dt + \sqrt{4\sigma^2 dt}\right)/2\,e^{rdt}$$

$$u = \left(2 + (2r + \sigma^2)dt + 2\sigma\sqrt{dt}\right)/2\,e^{rdt}$$

$$u = (1 + dtr + .5 * dt * \sigma^2 + \sigma\sqrt{dt})/\,e^{rdt}$$

$$u = (1 + dtr + .5 * dt * \sigma^2 + \sigma\sqrt{dt})(-rdt + 1)$$

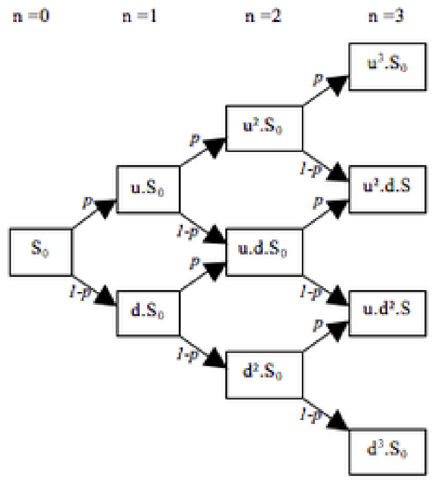
$$u = 1 + .5dt * \sigma^2 + \sigma\sqrt{dt} = Second\ Order\ Expansion\ of\ e^{\sigma\sqrt{dt}}$$

$$u = e^{\sigma\sqrt{dt}}$$

$$d = e^{-\sigma\sqrt{dt}}$$

We now have the ability to calculate the value of the option

Figure 3



$$p = \frac{e^{rt/n} - d}{u - d}$$

$$u = e^{\sigma\sqrt{t/n}}$$

$$d = e^{-\sigma\sqrt{t/n}}$$

To compute the value of the option by hand at n = 3, calculate the value of the underlying at the final state's nodes and then find the payoff of the call option: max[S-K, 0]. Now we begin the inductive analysis. At n=2 find the value of the option in each of the three nodes by considering the one-step model. If this was an American option, you would check max [option value, exercise value] at each node. Keep repeating this process until you reach time zero where you will be left one value, which is the value of the European call option. To extend this process to a k-period process simply repeat this inductive backward procedure from state k.

It is easy to see that this general model can be applied to the valuation of projects of a firm like Berk, Green and Naik. Now the following equation is modified:

$$\ln\left(\frac{S_t + dt}{S_t}\right) \sim N(r - .5\sigma^2, \sigma^2 dt)$$

$$\ln\left(\frac{V_t + dt}{V_t}\right) \sim N(r - .5\sigma^2, \sigma^2 dt)$$

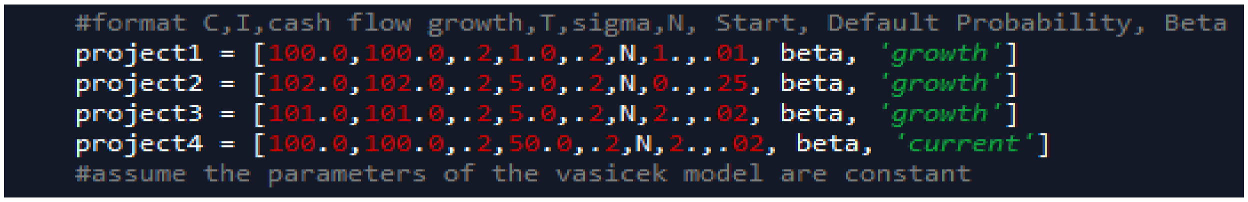
$$u = e^{\sigma\sqrt{}dt}$$

$$d = e^{-\sigma\sqrt{}dt}$$

Where S_0 is simply the initial investment I., σ represents the volatility of the cash flows, and r will now be replaced by $\bar{C} - r - \beta$. r is determined by the average expectation of the Vasicek model at each period where the value of the option is being evaluated. For example, if n=3, you would discount each period with the expectation of a 3 period interest rate model. The strike price, K, is simply just the initial investment. In the interest of reducing complexity, the infinite case is not considered. The infinite case is rather nonsensical considering strategic initiatives for determining whether to take on a project rarely require projections greater than 3-5 years. The value of the option is adjusted by the default probability, where it is assumed that none of the project's value can be recovered. This is a rather aggressive assumption depending on the type of investment. To mimic a company having current assets in place generating cash flows in perpetuity, I simply consider an option of 50 years. The decay of value in the option is slow, but also captures the idea that other firms reduce competitive edge over time, thereby reducing the value of the firm unless new projects are undertaken.

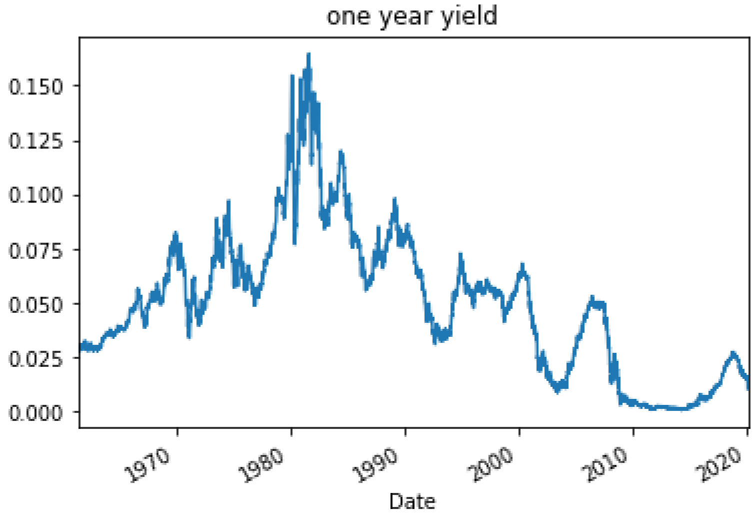
Implementation

To implement the above paper's model with the binomial method, I used a 3-year projection of the firm's values. For the sake of simplicity, the firm's value is solely due to four projects. The reason I chose to do four projects instead of considering one project that represents the technology of the firm is that I would like to build a robust framework, which incorporates the changes in valuation with investment projects as they are introduced. You could use a framework like this to evaluate the effect of the news on the valuation of a firm.

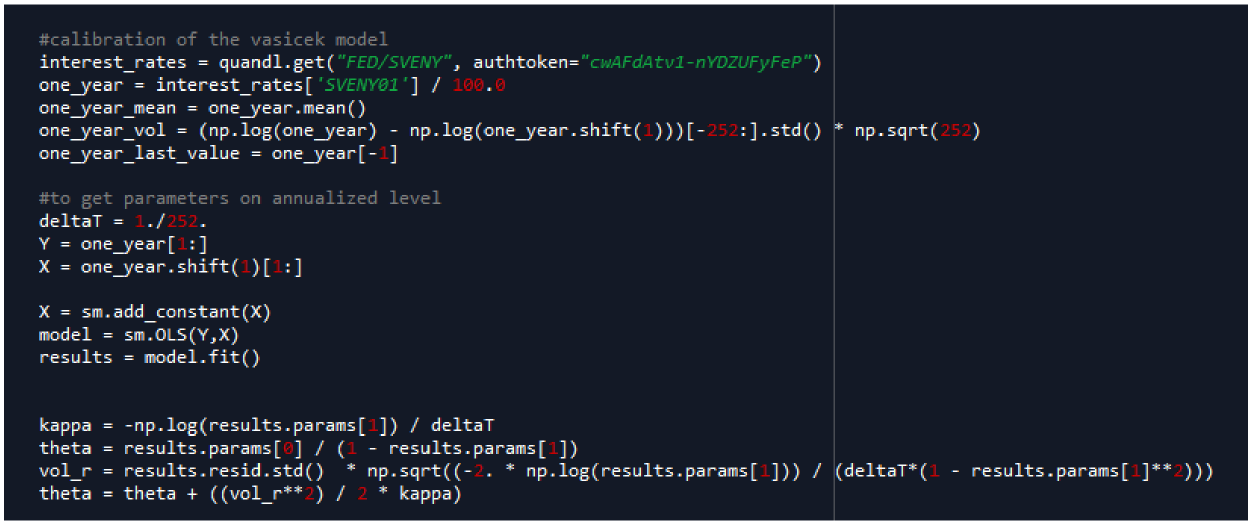


```
    #format C,I,cash flow growth,T,sigma,N, Start, Default Probability, Beta
    project1 = [100.0,100.0,.2,1.0,.2,N,1.,.01, beta, 'growth']
    project2 = [102.0,102.0,.2,5.0,.2,N,0.,.25, beta, 'growth']
    project3 = [101.0,101.0,.2,5.0,.2,N,2.,.02, beta, 'growth']
    project4 = [100.0,100.0,.2,50.0,.2,N,2.,.02, beta, 'current']
    #assume the parameters of the vasicek model are constant
```

For example, project 1 has an initial value of $100, which is equal to the initial investment. The period of the project is one year with a volatility of the value of the investment of .2. The mean growth rate of the cash flows of the project is assumed a value of .2. We will use a three period binomial model to evaluate the project. The default probability of the project is 1%, which is assumed constant across the life of the project. The interest rate of .05 is a placeholder in this context. The beta was presumed a constant 0.0. As done in the paper, I replicated the discounting via the Vasicek model by discounting each period of the binomial model using the average expectation of the interest rate for that period. For each period, I assumed the same r0, kappa, theta, and sigma across time for the model in the interest of reproducibility and simplicity. The parameters chosen were determined from downloading the Federal Reserve's daily coupon yield from the quandl api and calculating the parameters of interest over the one year yields going back to 6/14/61. The mean rate found was approximately 5% with an autocorrelation of .998, which is expected considering Berk found a sample correlation of .969 for a one-month T-bill rate. One would expect autocorrelation to increase with maturity. The average rate used in my model is lower than the .07483 rate used by Berk because of recent quantitative easing.

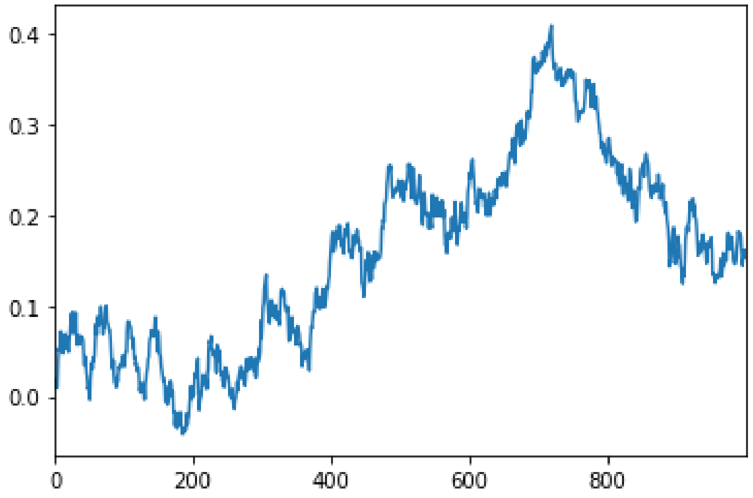


Please see the code below for the implementation of the calibration of the Vasicek model.

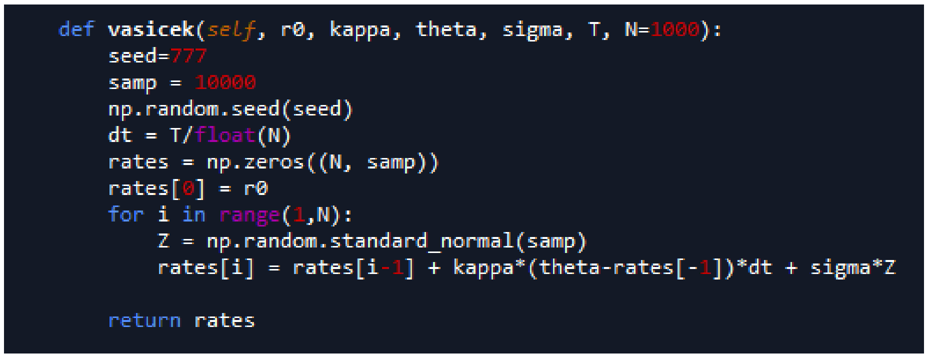


```
#calibration of the vasicek model
interest_rates = quandl.get("FED/SVENY", authtoken="cwAFdAtv1-nYDZUFyFeP")
one_year = interest_rates['SVENY01'] / 100.0
one_year_mean = one_year.mean()
one_year_vol = (np.log(one_year) - np.log(one_year.shift(1)))[-252:].std() * np.sqrt(252)
one_year_last_value = one_year[-1]

#to get parameters on annualized level
deltaT = 1./252.
Y = one_year[1:]
X = one_year.shift(1)[1:]

X = sm.add_constant(X)
model = sm.OLS(Y,X)
results = model.fit()


kappa = -np.log(results.params[1]) / deltaT
theta = results.params[0] / (1 - results.params[1])
vol_r = results.resid.std()  * np.sqrt((-2. * np.log(results.params[1])) / (deltaT*(1 - results.params[1]**2)))
theta = theta + ((vol_r**2) / 2 * kappa)
```

To apply this model to real world data and in real time you would determine the parameters at each point in time based off the current yield curve. Please see below for an example simulation path for the short rate of one year.

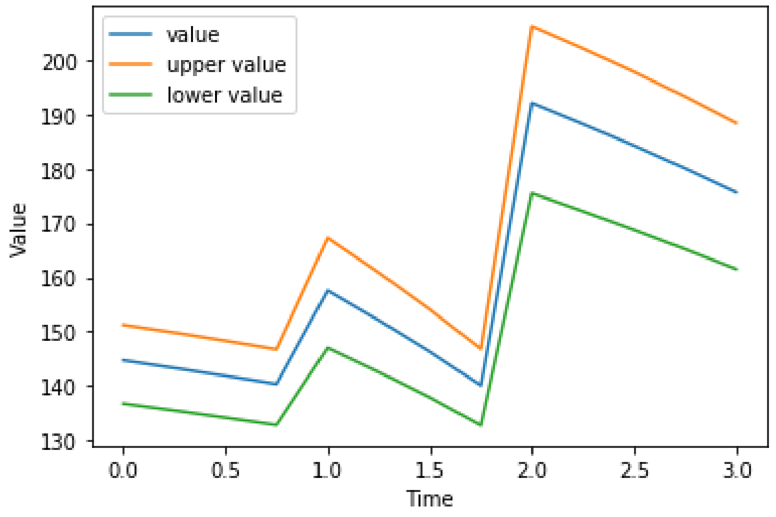


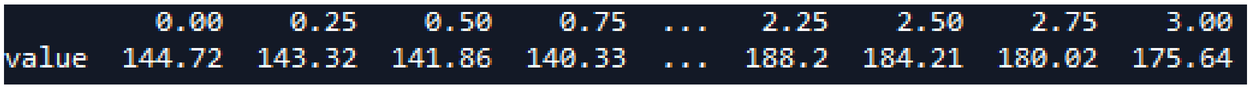
The Vasicek model parameters calibration was done via an OLS on the interest rates and daily lagged interest rates. The parameters were calculated in the same methodology given in Carlos Vega’s paper “Calibration of the exponential Ornstein–Uhlenbeck process…”.[7] The method for calculating the Vasicek model is contained below.



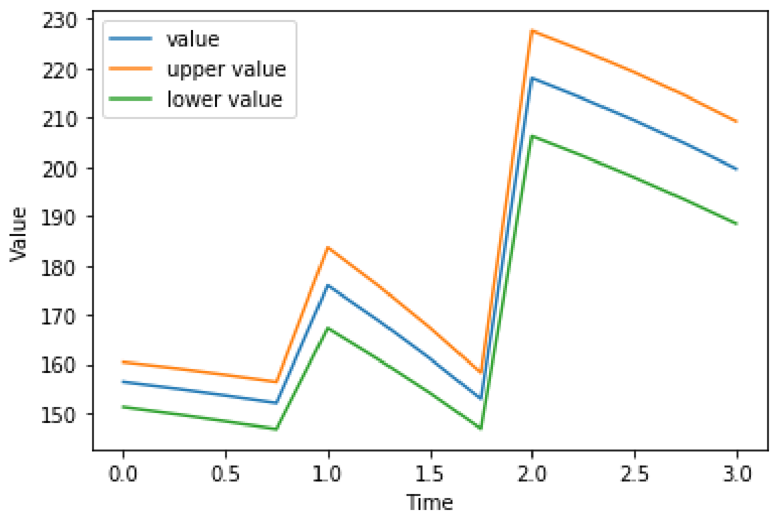
```
def vasicek(self, r0, kappa, theta, sigma, T, N=1000):
    seed=777
    samp = 10000
    np.random.seed(seed)
    dt = T/float(N)
    rates = np.zeros((N, samp))
    rates[0] = r0
    for i in range(1,N):
        Z = np.random.standard_normal(samp)
        rates[i] = rates[i-1] + kappa*(theta-rates[-1])*dt + sigma*Z

    return rates
```

To calculate the value for the firm over three years, I analyzed the firm's value at every .25 years to mimic the effect of a quarterly statement release. Instead of using a perpetual riskless consol bond to value the firm's current assets, I used the value of the option of the projects in place going forward. For example, a 3-year project started at time zero evaluated in time one will be evaluated as an option with 2 years of future cash flows. To value the potential growth options of projects as done in Berk's paper I added the value of the growth option in the period that project is about to be undertaken. This assumes that there is no delay in the decision when the project is financed. I chose not to use the perpetual riskless consol bond in the valuation of the firm's value because it is not consistent in the thought process of nearly all corporate finance professionals who typically value projects in three to five year time period. The value of all the projects together was calculated to give the value of the firm in each period. In addition, a bull and bear case was considered by adding a positive and negative shock of 5% to the growth rate of cash flows of the growth options of the project. Please see the graph below for output.




```
         0.00    0.25    0.50    0.75  ...   2.25    2.50    2.75    3.00
value  144.72  143.32  141.86  140.33  ...  188.2  184.21  180.02  175.64
```

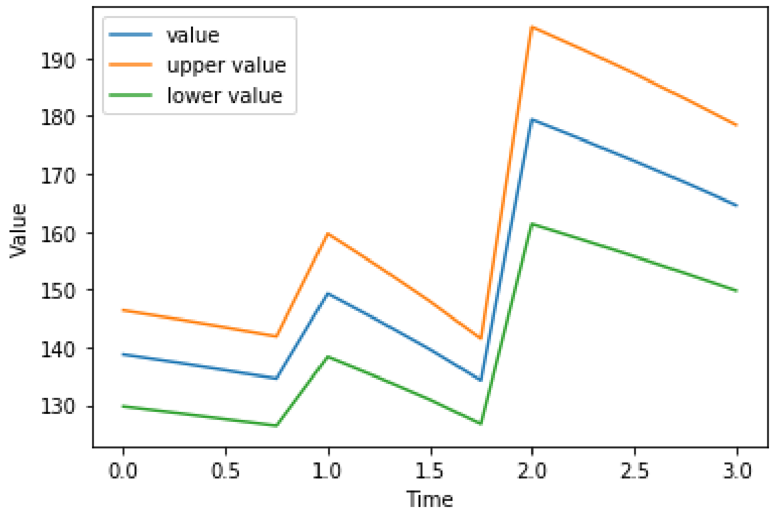
To check the effect of changing beta, I will run the program with a beta of -.1 for all the projects. We should see the value of the firm increase. Diversified firms that take on projects with cash flows with lower betas should have higher valuations.



```
        0.00    0.25    0.50    0.75   ...    2.25    2.50   2.75    3.00
value 156.38  155.05  153.61  152.07   ...  213.86  209.45  204.7  199.61
```

Not surprisingly, shocking the beta increased the 3rd year valuation of the firm by 13.6%.

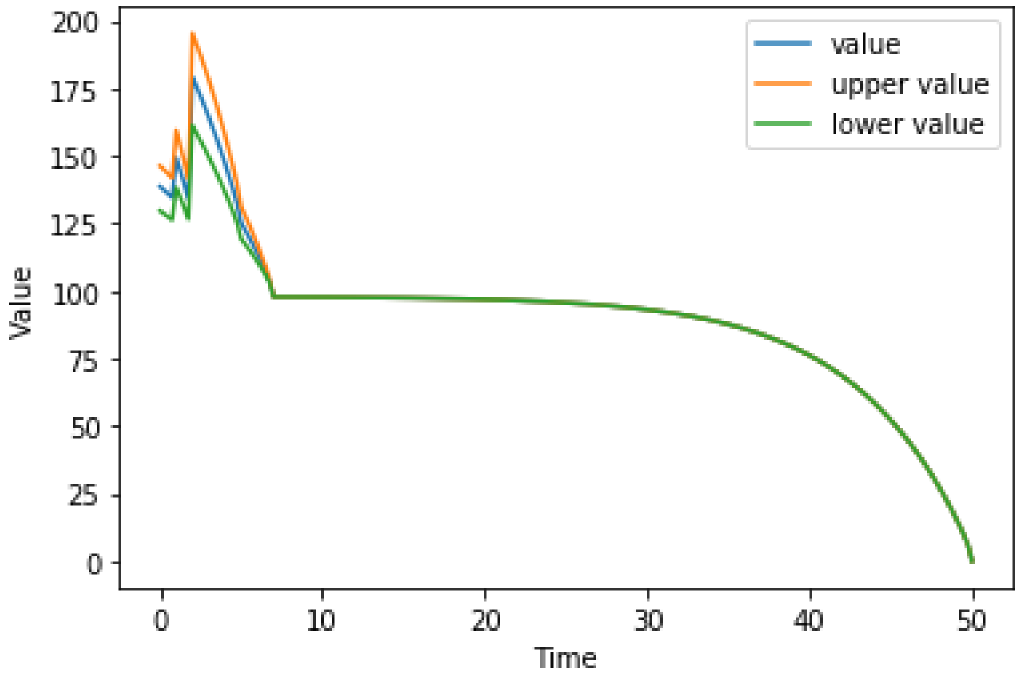
In addition to looking at the effect of beta, a simulation was run using a constant interest rate of approximately 5%. I would assume that the valuation of the firm in period 3 would be approximately similar to the model including the Vasicek methodology.



```
        0.00    0.25    0.50    0.75   ...    2.25    2.50    2.75    3.00
value 138.74  137.41  136.02  134.59   ...  175.87  172.25  168.49  164.59
```

The binomial model with a constant rate had a valuation that was 6.3% less due to the current lower interest rate environment. Lower interest rates inflate valuations of a firm.

I did not assume that any of the projects ended in the three-year period as it is illogical to assume that you can foresee the end of an economy. Also, all firm's option values were adjusted for the default probability of the cash flows, which were assumed constant across time for the sake of convenience. You can see that in periods one and two there are shocks to the value of the firm as the value of the growth options was added. The growth of the value of the firm is consistent with what you would see in the real world as indicated by its growth like a stock. Obviously, the value of the firm decreases, as there are less potential cash flows as projects near completion. You should see that in year fifty the value of the firm goes to zero as all projects are completed and no new projects are taken on.



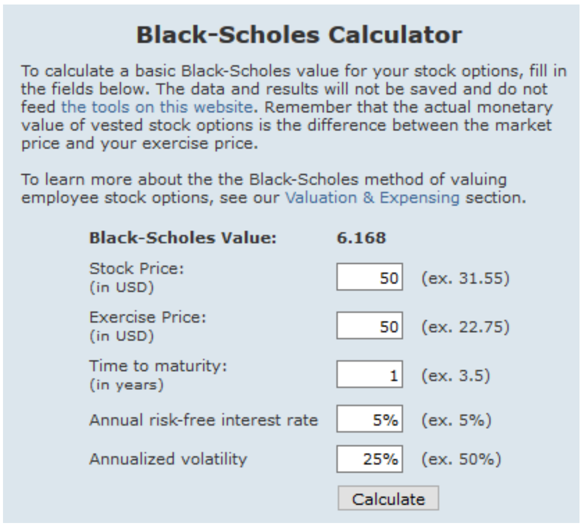
Further Improvements

There are obviously many improvements that I could make the methodology outlined above. For example, I could incorporate the ability to add the value of growth options in future states through the multiplication of a risk-free discount bond. The rate used could be simulated with the Vasicek model as well. In addition, the model could be improved by adding the ability to recover some investment into the value of the investment options. Path dependence was considered in the volatility parameter nor were greeks calculated.

In making this paper more consistent with the original methodology, I should consider the case where the value of the firm is not the sum of several projects but rather with parameters reflecting all projects contributing to the firm’s technology. The binomial case would have to be extended to an infinite period for creating the perpetual bond. Simulations then can be run to compare the results to the Fama French model to align the results with Berk, Green, and Naik.

Thank you for taking the time to read my write up!

Model Validation
Figure 4

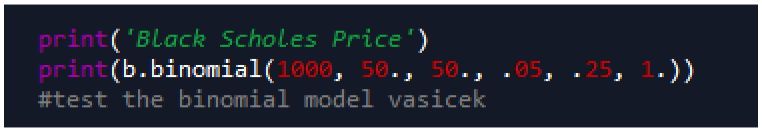


Black-Scholes Calculator

To calculate a basic Black-Scholes value for your stock options, fill in the fields below. The data and results will not be saved and do not feed the tools on this website. Remember that the actual monetary value of vested stock options is the difference between the market price and your exercise price.

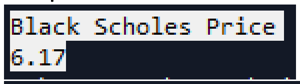
To learn more about the the Black-Scholes method of valuing employee stock options, see our Valuation & Expensing section.

Black-Scholes Value:	**6.168**	
Stock Price: (in USD)	50	(ex. 31.55)
Exercise Price: (in USD)	50	(ex. 22.75)
Time to maturity: (in years)	1	(ex. 3.5)
Annual risk-free interest rate	5%	(ex. 5%)
Annualized volatility	25%	(ex. 50%)

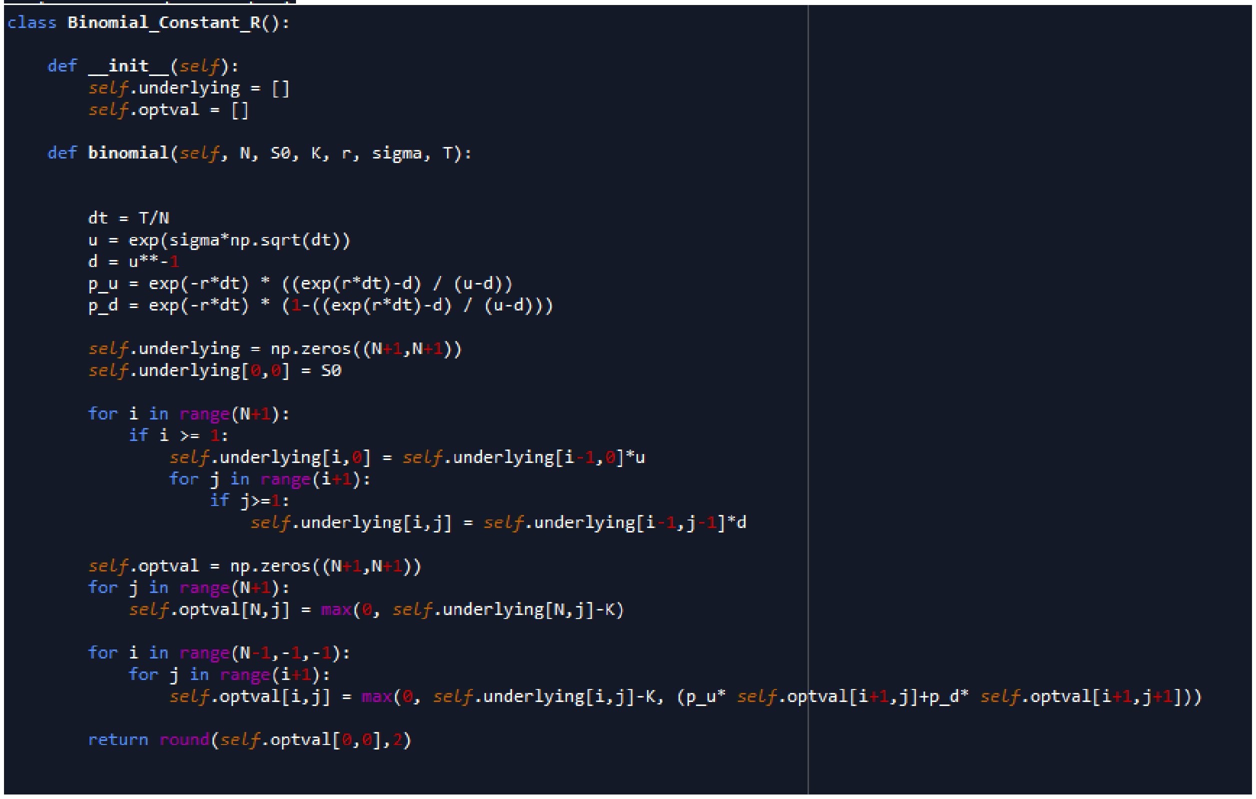


```python
print('Black Scholes Price')
print(b.binomial(1000, 50., 50., .05, .25, 1.))
#test the binomial model vasicek
```

Output from code



```
Black Scholes Price
6.17
```



```python
class Binomial_Constant_R():

    def __init__(self):
        self.underlying = []
        self.optval = []

    def binomial(self, N, S0, K, r, sigma, T):


        dt = T/N
        u = exp(sigma*np.sqrt(dt))
        d = u**-1
        p_u = exp(-r*dt) * ((exp(r*dt)-d) / (u-d))
        p_d = exp(-r*dt) * (1-((exp(r*dt)-d) / (u-d)))

        self.underlying = np.zeros((N+1,N+1))
        self.underlying[0,0] = S0

        for i in range(N+1):
            if i >= 1:
                self.underlying[i,0] = self.underlying[i-1,0]*u
                for j in range(i+1):
                    if j>=1:
                        self.underlying[i,j] = self.underlying[i-1,j-1]*d

        self.optval = np.zeros((N+1,N+1))
        for j in range(N+1):
            self.optval[N,j] = max(0, self.underlying[N,j]-K)

        for i in range(N-1,-1,-1):
            for j in range(i+1):
                self.optval[i,j] = max(0, self.underlying[i,j]-K, (p_u* self.optval[i+1,j]+p_d* self.optval[i+1,j+1]))

        return round(self.optval[0,0],2)
```

References

1. Berk, J. B., Green, R. C., & Naik, V. (1999). Optimal Investment, Growth Options, and Security Returns. *The Journal of Finance*, *54*(5), 1553–1607. doi: 10.1111/0022-1082.00161
2. Figure 1. *Binomial Pricing*. quantinsti. https://quantra.quantinsti.com/glossary/Binomial-Pricing
3. Figure 2. Goddard Consulting. https://www.goddardconsulting.ca/option-pricing-binomial-index.html
4. Figure 3. *Binomial Options Pricing Model*. Wikipedia. https://en.wikipedia.org/wiki/Binomial_options_pricing_model#/media/File:Arbre_Binomial_Options_Reelles.png
5. Figure 4. *Black-Scholes Calculator*. mystockoptions. https://www.mystockoptions.com/black-scholes.cfm
6. https://advancesindifferenceequations.springeropen.com/articles/10.1186/s13662-018-1718-4
7. Vega, C. A. M. (2018). Calibration of the exponential Ornstein–Uhlenbeck process when spot prices are visible through the maximum log-likelihood method. Example with gold prices. *Advances in Difference Equations*, *2018*(1). doi: 10.1186/s13662-018-1718-4
8. Wang, Jyran. Ch04 Binomial Tree Model. NTU. http://homepage.ntu.edu.tw/~jryanwang/course/Financial%20Computation%20or%20Financial%20Engineering%20(graduate%20level)/FE_Ch04%20Binomial%20Tree%20Model.pdf

Python Code

```
"""
Created on Sat Feb 15 14:48:13 2020

@author: jjanko
"""
from math import exp, sqrt
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import quandl
import statsmodels.api as sm

class Binomial_Constant_R():

    def __init__(self):
        self.underlying = []
        self.optval = []

    def binomial(self, N, S0, K, r, sigma, T):



        dt = T/N
        u = exp(sigma*np.sqrt(dt))
        d = u**-1
        p_u = exp(-r*dt) * ((exp(r*dt)-d) / (u-d))
        p_d = exp(-r*dt) * (1-((exp(r*dt)-d) / (u-d)))

        self.underlying = np.zeros((N+1,N+1))
        self.underlying[0,0] = S0

        for i in range(N+1):
            if i >= 1:
                self.underlying[i,0] = self.underlying[i-1,0]*u
                for j in range(i+1):
                    if j>=1:
                        self.underlying[i,j] = self.underlying[i-1,j-1]*d

        self.optval = np.zeros((N+1,N+1))
        for j in range(N+1):
            self.optval[N,j] = max(0, self.underlying[N,j]-K)

        for i in range(N-1,-1,-1):
            for j in range(i+1):
                self.optval[i,j] = max(0, self.underlying[i,j]-K, (p_u* self.optval[i+1,j]+p_d* self.optval[i+1,j+1]))

        return round(self.optval[0,0],2)



class project():

    def __init__(self, inputlist):

        self.S0 = inputlist[0]
        self.K=inputlist[1]
        self.mu=inputlist[2]
        self.Time = inputlist[3]
        self.sigma = inputlist[4]

        self.N=inputlist[5]

        self.Start = inputlist[6]

        self.defaultProbability = inputlist[7]

        self.beta = inputlist[8]

        self.Type = inputlist[9]

        self.underlying = []

        self.optval = []


    def vasicek(self, r0, kappa, theta, sigma, T, N=1000):

        seed=777

        samp = 10000

        np.random.seed(seed)

        dt = T/float(N)

        rates = np.zeros((N, samp))

        rates[0] = r0

        for i in range(1,N):

            Z = np.random.standard_normal(samp)

            rates[i] = rates[i-1] + kappa*(theta-rates[-1])*dt + sigma*Z


        return rates



    def get_p(self, r, dt, u, d):

        r = - self.beta + self.mu - r

        p_u = exp(-r*dt) * ((exp(r*dt)-d) / (u-d))

        p_d = exp(-r*dt) * (1-((exp(r*dt)-d) / (u-d)))

        return p_u, p_d
    def binomial(self, T, r):

        self.underlying = []

        self.optval = []

        dt = T/self.N

        u = exp(self.sigma*np.sqrt(dt))

        d = u**-1

        self.underlying = np.zeros((self.N+1,self.N+1))

        self.underlying[0,0] = self.S0


        for i in range(N+1):

            if i >= 1:

                self.underlying[i,0] = self.underlying[i-1,0]*u

                for j in range(i+1):

                    if j>=1:

                        self.underlying[i,j] = self.underlying[i-1,j-1]*d


        self.optval = np.zeros((self.N+1,self.N+1))

        for j in range(self.N+1):

            self.optval[self.N,j] = max(0, self.underlying[self.N,j]-self.K)


        for i in range(self.N-1,-1,-1):

            for j in range(i+1):

                p_u, p_d = self.get_p(r[i], dt, u, d)

                self.optval[i,j] = max(0, self.underlying[i,j]-self.K,
(p_u*self.optval[i+1,j]+p_d*self.optval[i+1,j+1]))

        return round(self.optval[0,0],2)
    def get_value_bv(self,T, r):

        return self.binomial(T,r) * (1.0-self.defaultProbability)

if __name__ == "__main__":

    N = 1000
    simulation_length = 3

    #this is for model validation purposes
    b = Binomial_Constant_R()

    print('Black Scholes Price')
    print(b.binomial(N, 50., 50., .05, .25, 1.))
    #test the binomial model vasicek



    api_key ='cwAFdAtv1-nYDZUFyFeP'



    #calibration of the vasicek model
    interest_rates = quandl.get("FED/SVENY", authtoken="cwAFdAtv1-nYDZUFyFeP")
    one_year = interest_rates['SVENY01'] / 100.0
    one_year_mean = one_year.mean()
    one_year_vol = (np.log(one_year) - np.log(one_year.shift(1)))[-252:].std() * np.sqrt(252)
    one_year_last_value = one_year[-1]

    #to get parameters on annualized level
    #assume there is 252 trading days in the year
```

```
dt = 1./252.

Y = one_year[1:]

X = one_year.shift(1)[1:]


X = sm.add_constant(X)

model = sm.OLS(Y,X)

results = model.fit()



kappa = -np.log(results.params[1]) / dt

theta = results.params[0] / (1 - results.params[1])

vol_r = results.resid.std()  * np.sqrt((-2. * np.log(results.params[1])) / (dt*(1 - results.params[1]**2)))

theta = theta + ((vol_r**2) / 2 * kappa)



#since the growth of the cash flows are considered constant

beta = -0.1


shock = .05


#format C,I,cash flow growth,T,sigma,N, Start, Default Probability, Beta

project1 = [100.0,100.0,.2,1.0,.2,N,1.,.01, beta, 'growth']

project2 = [102.0,102.0,.2,5.0,.2,N,0.,.25, beta, 'growth']

project3 = [101.0,101.0,.2,5.0,.2,N,2.,.02, beta, 'growth']

project4 = [100.0,100.0,.2,50.0,.2,N,0.,.02, beta, 'current']

#assume the parameters of the vasicek model are constant


projects = [project1, project2, project3, project4]

projects_upper_bound = projects.copy()
```

```
projects_lower_bound = projects.copy()



for i in range(len(projects)):
    #get a list of objects

    projects[i] = project(projects[i])

    projects_upper_bound[i] = project(projects_upper_bound[i])
    projects_lower_bound[i] = project(projects_lower_bound[i])

    if projects[i].Type == 'growth':
        projects_upper_bound[i].mu = projects_upper_bound[i].mu + shock
        projects_lower_bound[i].mu = projects_lower_bound[i].mu - shock

value = {}
upper_value = {}
lower_value = {}




constant_r = [one_year.mean()] * N

i = 0
while i <= simulation_length :
    v = 0.0
    u_v = 0.0
    l_v = 0.0
    for j in projects:
        if j.Start - i <= 0.0 and j.Time + j.Start - i > 0.0:
            r = j.vasicek(one_year_last_value, kappa, theta, vol_r, j.Time + j.Start - i)
            r = pd.DataFrame(r).transpose().mean().transpose().to_numpy()
            v = v + j.get_value_bv(j.Time + j.Start - i, r)

    value[i] = round(v,2)

    for j in projects_upper_bound:
        if j.Start - i <= 0.0 and j.Time + j.Start - i > 0.0:
            r = j.vasicek(one_year_last_value, kappa, theta, vol_r, j.Time + j.Start - i)
            r = pd.DataFrame(r).transpose().mean().transpose().to_numpy()
            u_v = u_v + j.get_value_bv(j.Time + j.Start - i, r)



    upper_value[i] = round(u_v, 2)



    for j in projects_lower_bound:
        if j.Start - i <= 0.0 and j.Time + j.Start - i > 0.0:
            r = j.vasicek(one_year_last_value, kappa, theta, vol_r, j.Time + j.Start - i)
            r = pd.DataFrame(r).transpose().mean().transpose().to_numpy()
            l_v = l_v + j.get_value_bv(j.Time + j.Start - i, r)

    lower_value[i] = round(l_v, 2)

    i = i + .25

val =pd.DataFrame.from_dict(value, orient = 'index')
```

```
val.columns = ['value']

print(val.transpose())

plt.figure()


x = pd.DataFrame.from_dict(value, orient = 'index')

x.columns  = ['dynamic value of firm']


plt.plot(x, label = 'value')

plt.plot(pd.DataFrame.from_dict(upper_value, orient = 'index'), label = 'upper value')

plt.plot(pd.DataFrame.from_dict(lower_value, orient = 'index'), label = 'lower value')


plt.xlabel('Time')

plt.ylabel('Value')

plt.legend()

plt.show()


rates = projects[0].vasicek(one_year_last_value, kappa, theta, vol_r, j.Time + j.Start - i)

sample_path = pd.DataFrame(rates[:,-2])

sample_path.plot(legend = False)
```