Student nr. 35275809

MATH580

Project on Financial Stochastic Processes

Introduction

This report first covers the pricing of European options discussing how different factors such as the time to expiration, the volatility of the stock price, the strike price, the risk-free interest rate and the initial price affect the price of a European call option.

By applying the Black-Scholes model for stock prices, we were able to plot 5 different figures illustrating that the value of a European call option increases when the volatility, the initial price of the stock and the risk-free interest rate was increased respectively (by keeping all other variables constant). In addition, while there is usually a positive relation between the time to expiration and the value of a European call option, this is not always the case. On the other hand, there is a negative relation between the strike price and the European call option price.

While the Black-Scholes model describes the price of a stock at time t using a Wiener process (WP), the Vasicek model defines the price at time 0 of a bond paying one unit at time t by $Q_t = e^{\left(-\int_0^t R_s ds\right)}$, where R_s, the spot-rate process, is an Ornstein-Uhlenbeck process (OUP). If we had simulated R_s using a WP instead of an OUP we would have observed much more unsmooth realisations of R_s over time than for the ones given using the OUP. To understand the difference between these two processes, we have plotted the expected value and variance of the spot-rate process (as an OUP). By observing that $E(R_s)$ and $V(R_s)$ each converge to a finite limit over time, we can conclude that interest rates can be modelled using a OUP because it follows mean reversion, reverting to a constant, long run mean and therefore contrasting the random walk in the WP. Vasicek's model therefore succeeds in depicting the economic phenomena that interest rates appear over time to be pulled back to some long run average. Finally, we were able to conclude that Q_t follows a lognormal distribution for a fixed value of t.

Pricing of European Options

Figure 1: Price of a European call option against time

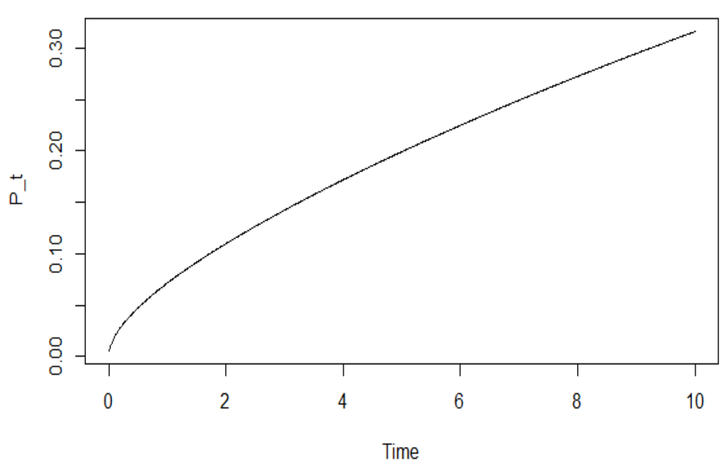
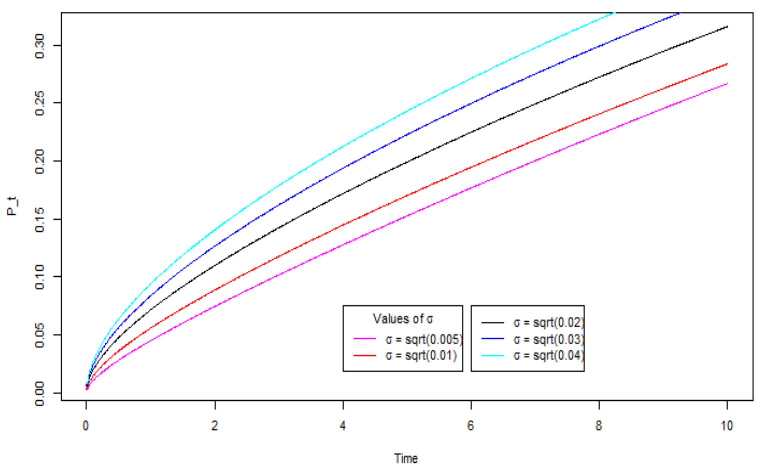


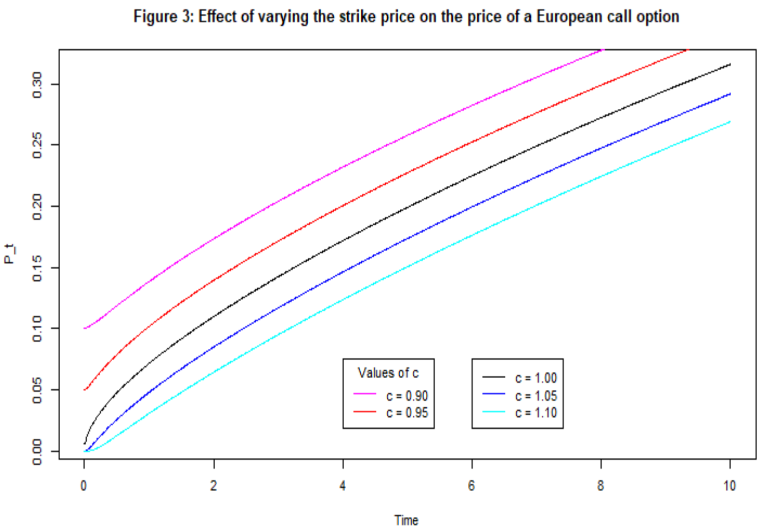
Figure 2: Effect of varying the volatility on the price of a European call option

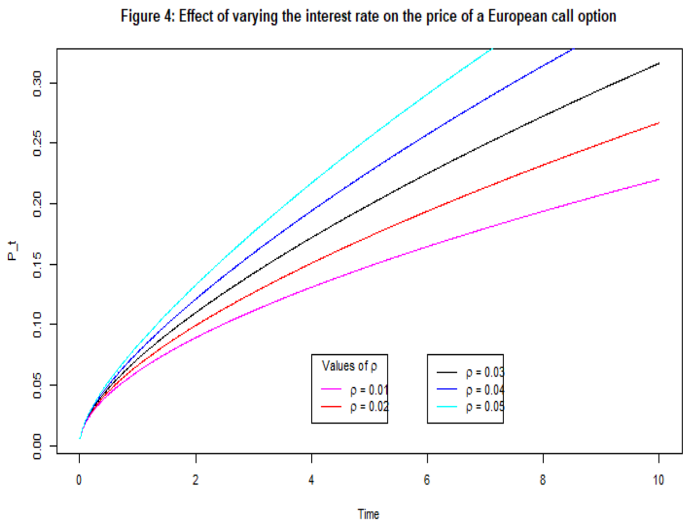


The first right figure above shows the effect of the expiration date (t) on the price of a European call option. We observe that the European call option becomes more valuable as the time to expiration increases, even though this is not always the case. Usually, a longer time to expiration implies more time for the stock price to change favourably and yields a higher call option price.

The second left figure above shows the effect of varying the volatility of the stock (σ) on the call option price. A rise in volatility means that the certainty of the stock performing very well or very poorly increases. Since the owner of a call benefits if the price of the stock increases and cannot lose more than the price of the option if the price of the stock decreases, the value a call option increases as the volatility of the stock price increases and vice versa.

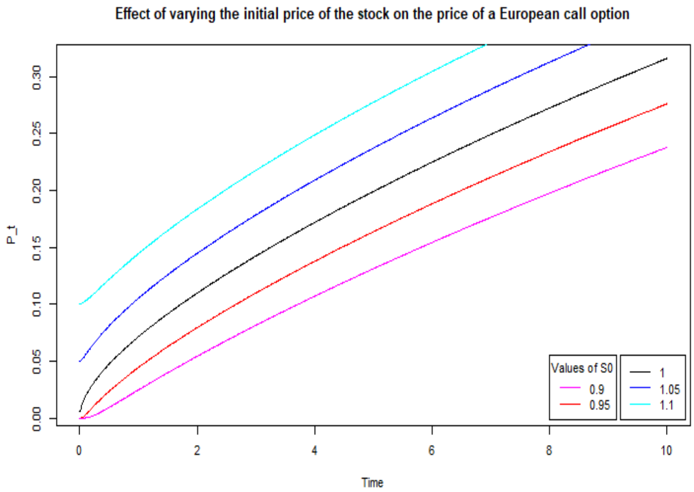
The third right figure below shows the effect of varying the strike price (c) on the call option price. When a call option is exercised at a specific future time, the payoff is the amount by which the stock price exceeds the strike price. Therefore, the call option becomes less valuable if the strike price increases and more valuable if the strike price decreases.





The fourth left figure above shows the effect of varying the interest rate (ρ) on a European call option price. An increase in interest rates means that the expected return from the stock required by investors usually increases while the present value of any future cash flow received by the holder of the option decreases. In other words, it becomes more valuable to invest in a call option rather than buying the stock right now. As a result, the value of the call option increases. On the other hand, if the interest rate decreases, buying the stock right now becomes more attractive than investing in a call option and thus the call option price decreases.

Finally, the figure below shows the effect of the initial price of the stock (S_0) on the price of the call option. We observe that when the initial stock price increases, the value of the call option increases and if the initial stock price decreases, the value of the call option decreases. Since the payoff when a call option is exercised is the amount by which the stock price exceeds the strike price, call options become more valuable as the stock price increases and less valuable as the strike price decreases.



The Vasicek model for interest rates and the OU process

The first figure below shows 10 realisations of the spot-rate process, R_s, over time s: $0 \leq s \leq 10$ while the second figure on the next page shows just the first realisation over the same period of time.

Figure 1: 10 realisations of the spot-rate process (R_s) over time (s)

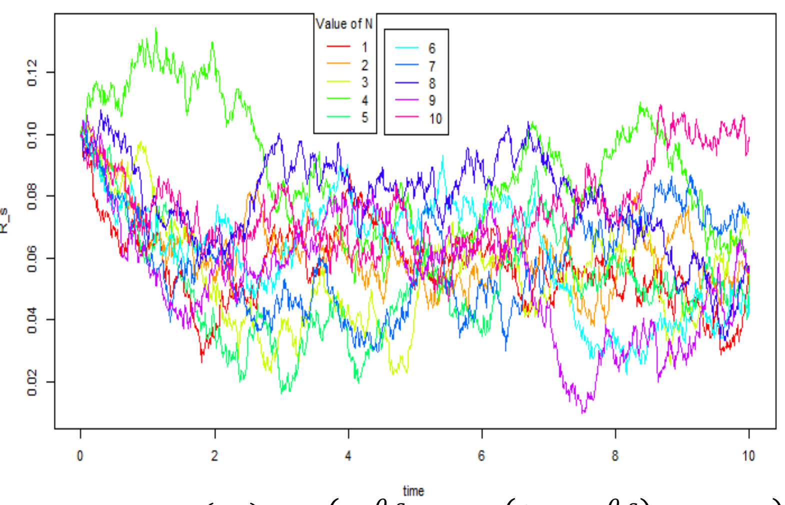
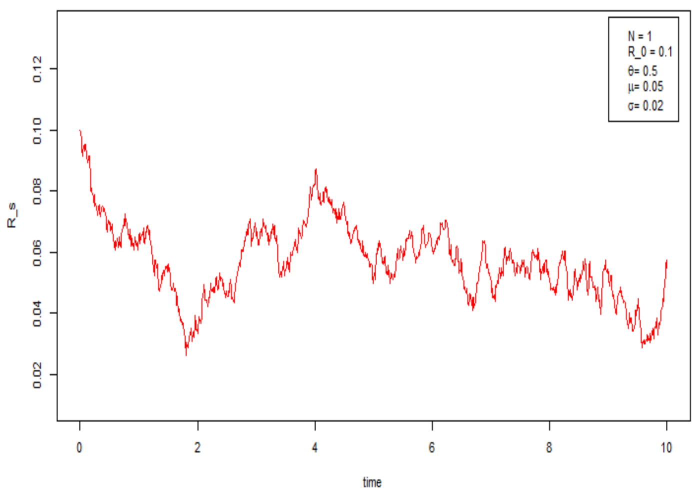


Figure 2: The first realisation of the spot-rate process



Next, $\boldsymbol{E(R_s)} = E\left(e^{-\theta \cdot s} \cdot R_0 + \left(1 - e^{-\theta \cdot s}\right) \cdot \mu + X_s\right) = e^{-\theta \cdot s} \cdot R_0 + \left(1 - e^{-\theta \cdot s}\right) \cdot \mu + E(X_s)$

$= \boldsymbol{e^{-\theta \cdot s} \cdot R_0 + \left(1 - e^{-\theta \cdot s}\right) \cdot \mu}$, and $\boldsymbol{V(R_s)} = V\left(e^{-\theta \cdot s} \cdot R_0 + \left(1 - e^{-\theta \cdot s}\right) \cdot \mu + X_s\right) = V(X_s) =$ $Cov(X_s, X_s) = \frac{\sigma^2}{2 \cdot \theta} \cdot e^{-\theta \cdot (s+s)} \cdot \left(e^{2 \cdot \theta \cdot \min(s,s)} - 1\right) = \boldsymbol{\frac{\sigma^2}{2 \cdot \theta} \cdot \left(1 - e^{-2 \cdot \theta \cdot s}\right)}$

Plotting these results for each 10 realisations of R_s for various values of θ and σ produces the two figures below:

Figure 3: The expected value of the spot-rate process

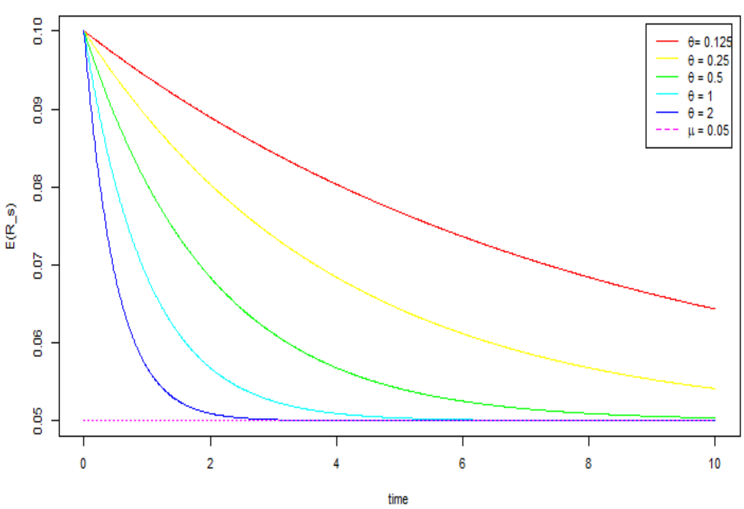
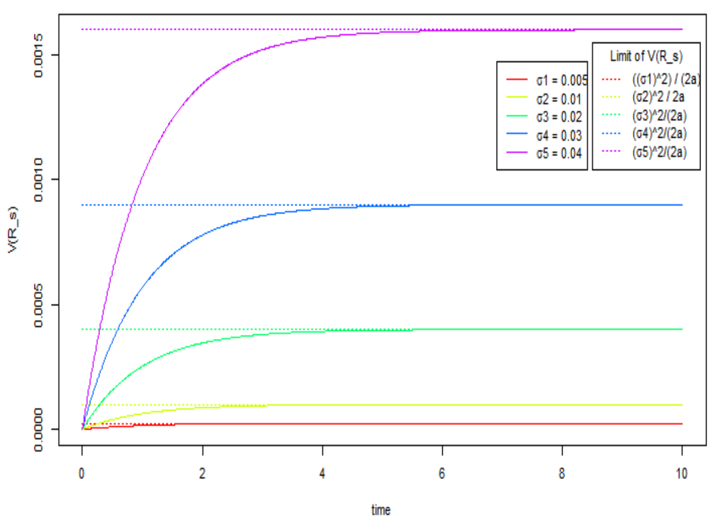


Figure 4: The variance of the spot-rate process



We note that the $E(R_s)$ and $V(R_s)$ for the 10 realisations of R_s will yield the same value for $E(R_s)$ and $V(R_s)$ as the expressions for $E(R_s)$ and $V(R_s)$ is independent of X_s (but dependent on $E(X_s)$ and $V(X_s)$).

We observe that $\lim_{s \to \infty} E(R_s) = \mu$ and $\lim_{s \to \infty} V(R_s) = \frac{\sigma^2}{2 \cdot \theta}$ for each 10 realisations illustrating mean reversion and reflecting the economic phenomena that interest rates appear over time to be pulled back to some long run average μ. For instance, if interest rates increase, the economy will slow down and there will be less demand for loans causing interest rates to fall, whereas a fall in interest rates will yield an increase in the demand for loans as well as stronger economic growth leading to a rise in interest rates. Furthermore, increasing the value of θ means the $E(R_s)$ will converge faster to μ as shown in Figure 3 above. On the other hand, increasing σ will cause the $V(R_s)$ to increase and as a result, $V(R_s)$ will converge more slowly to its limit $\frac{\sigma^2}{2 \cdot \theta}$, as illustrated in Figure 4 on the previous page.

By simulating the bond price Q_t at time 0 of a bond paying one unit at time t for the parameter values above we can plot 10 realisations of $\{Q_t : 0 \leq t \leq 10\}$ as illustrated in Figure 1 below.

Figure 1: The bond price at time 0 of a bond paying one unit at time t

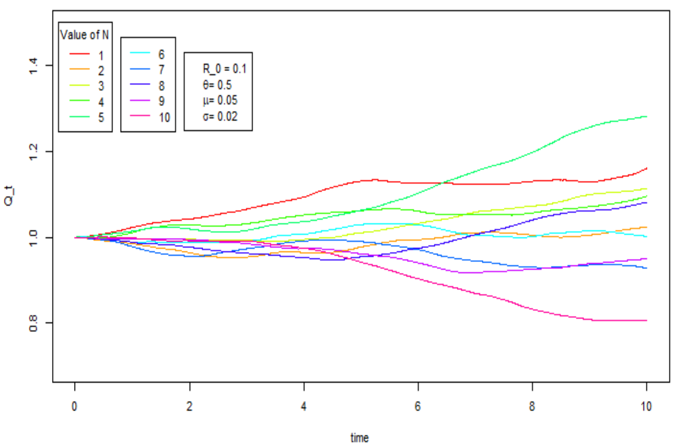
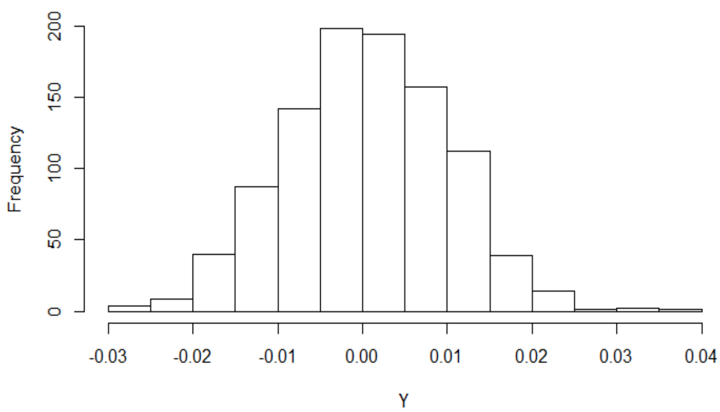
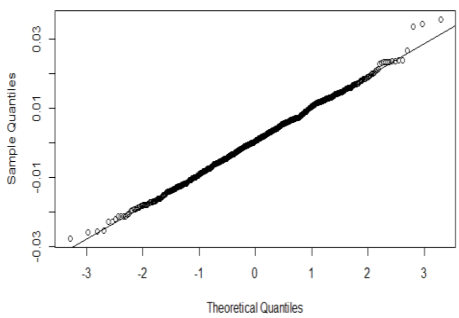


Figure 2: Histogram



To find the distribution of Q_t for a fixed value of t, we can a fix t at $t = 1$ and simulate N = 1000 realisations of Q_1. If we plot a histogram of $Y_1 = \int_0^1 R_s$ as illustrated in Figure 2 above and use QQplot as shown in Figure 3 below, we can conclude that Y_1 has a normal distribution. Since $Q_1 = e^{-Y_1}$ and $-Y_1$ also follows a normal distribution (because $-Y_1$is a linear transformation of Y_1), $Q_1 = e^{-Y_1}$ has a lognormal distribution. Consequently, the distribution of $Q_t$for a fixed value of t is lognormal.

Figure 3: Normal Q-Q Plot



Conclusion

Even though enabling us to conclude how different factors affect the price of a European call option, the weaknesses of the Black Scholes model are some of the non-real assumptions it relies on. For instance, the model assumes that the rate of return (ρ) on the stock and the volatility of the stock price (σ) are constant. In practice, very few market participants are able to obtain financial services at the risk-free rate. Furthermore, while the volatility of a stock can be constant in a short-term period, it is never constant in the long term. While the Vasicek model captures mean reversion and gives an explicit formula for derivatives such as bonds, it does also permit the interest rate to become negative which is a characteristic if we assume an economic pre-crisis situation. As every model is limited in some perspective, we must acknowledge what information can be extracted from applying these models regardless of their limitations. Only then can we draw appropriate conclusions by taking into consideration these limitations and evaluating these against the reality.

Appendix

Underneath follows the bibliography and the R code used for solving the Project on Financial Stochastic Processes.

Bibliography

Beaty, Adam. 7 factors that affect an option's price. Retrieved from https://theoptionprophet.com/blog/7-factors-that-affect-an-option-s-price

Black-Scholes model (2019). In *Wikipedia The Free Encyclopedia.* Retrieved from https://en.wikipedia.org/wiki/Black–Scholes_model

Brigo, Damiano & Dalessandri, Antonio & Neugebauer, Matthias & Triki, Fares. (2007). *A stochastic Toolkit for Risk Management.* https://wwwf.imperial.ac.uk/~dbrigo/toolboxweb.pdf

Ellman, Allan. (2012, November 10). Factors that Influence Option Value + The Black-Scholes Model. Retrieved from https://www.thebluecollarinvestor.com/factors-that-influence-option-value-the-black-scholes-model-2/

Hull, John C. (2012). O*ptions, futures and other derivatives* https://math.dartmouth.edu/~m86f17/Downloads/Hull8thedition.pdf

Kenton, Will (2019). Black Scholes Model. In *Investopedia*. Retrieved from https://www.investopedia.com/terms/b/blackscholes.asp

Konvexity. (2019). Factors affecting value of an option. Retrieved from http://konvexity.com/factors-affecting-value-of-an-option

Livii, J. *Limitations of the Black Scholes model. https://www.researchgate.net/publication/255995710_Limitations_of_the_Black-Scholes_model*

Macroption. Black-Scholes Strike Price Input. Retrieved from https://www.macroption.com/black-scholes-strike-price/

Mean reversion (2019). In *Wikipedia The Free Encyclopedia.* Retrieved from https://en.wikipedia.org/wiki/Mean_reversion_(finance)

Quant Connect. Introduction to Options. Retrieved from https://www.quantconnect.com/tutorials/introduction-to-options/options-pricing-black-scholes-merton-model

Stehlíkov, Beáta. (2015). Short rate models: Evolution of the short rate. Retrieved from http://www.iam.fmph.uniba.sk/institute/stehlikova/fd14en/lectures/03_interest_rates_1.pdf

Short-rate model (2019). In *Wikipedia The Free Encyclopedia.* Retrieved from https://en.wikipedia.org/wiki/Short-rate_model

Vasicek model (2019). In *Wikipedia The Free Encyclopedia.* Retrieved from https://en.wikipedia.org/wiki/Vasicek_model

What are the assumptions behind the Black-Scholes model? (2017) In *Maxi-Pedia.* Retrieved from http://www.maxi-pedia.com/Black+Scholes+model+assumptions

R code used for the Black-Scholes model for stock prices

```
# The Black-Scholes model for stock prices

N = 1
Delta = 0.01
n = 1000 # Delta*n = 10

WP = rWiener(n,N,Delta)  # 1 realisation of the standard Wiener Process


# Transformation of the standard Wiener Process

S0 = 1 # initial price of the stock
sigma = sqrt(0.02) # = 0.14142 - volatility
rho = 0.03 # interest rate
c = 1 # strike price
mu = rho - (sigma^2)/2  # implies the expected discounted stock price is
constant
# Value of mu is 0.02 (mu is the drift rate)

St = matrix(0,nrow = N, ncol = n + 1) # matrix to store realisation

St[N,] = S0*exp(mu*WP$times+sigma*WP$X[N,]) # price of stock at time t



Pt = matrix(0,nrow=N,ncol= n+1) # matrix to store realisation

Pt[N,] = S0*pnorm(  ( log(S0/c) + (rho + (sigma^2)/2 )*WP$times ) /
(sigma*sqrt(WP$times)) )  -c*exp(-rho*WP$times)*pnorm( ( log(S0/c) +
(rho - (sigma^2)/2  )*WP$times )/ (sigma*sqrt(WP$times) ) ) # the price
at timet of a European call option

Pt[1,1] = Pt[1,2] # when t = 0, R calculates P0= 0/0 = NaN, so to enable
R to calculate range(Pt) to plot
# Pt against t, we set value of Pt[1,1] equal to value of Pt[2,1]

# Plot of P_t against time
plot(range(WP$times), range(Pt),type="n", main = "Figure 1: Price of a
European call option against time ", xlab = "Time", ylab = "P_t")
lines(WP$times, Pt[N,])




              # Varying value of sigma

plot(range(WP$times), range(Pt),type="n", main = "Figure 2: Effect of
varying the volatility on the price of a European call option" , xlab =
"Time", ylab = "P_t")
lines(WP$times, Pt[N,])

# reducing value of sigma
```

```r
# reducing value of sigma to sqrt(0.01)

sigma = sqrt(0.01)

mu = rho - (sigma^2)/2

St = matrix(0,nrow = N, ncol = n + 1)

St[N,] = S0*exp(mu*WP$times+sigma*WP$X[N,])

Pt = matrix(0,nrow=N,ncol= n+1)

Pt[N,] = S0*pnorm(  ( log(S0/c) + (rho + (sigma^2)/2 )*WP$times ) /
(sigma*sqrt(WP$times)) )  -c*exp(-rho*WP$times)*pnorm( ( log(S0/c) +
(rho - (sigma^2)/2  )*WP$times )/ (sigma*sqrt(WP$times) ) )

Pt[1,1] = Pt[1,2]

lines(WP$times,  Pt[N,], col = 2)

# reducing value of sigma to sqrt(0.005)
sigma = sqrt(0.005)

mu = rho - (sigma^2)/2

St = matrix(0,nrow = N, ncol = n + 1)

St[N,] = S0*exp(mu*WP$times+sigma*WP$X[N,])

Pt = matrix(0,nrow=N,ncol= n+1)

Pt[N,] = S0*pnorm(  ( log(S0/c) + (rho + (sigma^2)/2 )*WP$times ) /
(sigma*sqrt(WP$times)) )  -c*exp(-rho*WP$times)*pnorm( ( log(S0/c) +
(rho - (sigma^2)/2  )*WP$times )/ (sigma*sqrt(WP$times) ) )

Pt[1,1] = Pt[1,2]

lines(WP$times,  Pt[N,], col = "magenta")


# Increasing value of sigma to sqrt(0.03)
sigma = sqrt(0.03) # = 0.1732

mu = rho - (sigma^2)/2  # mu = 0.015

St = matrix(0,nrow = N, ncol = n + 1)

St[N,] = S0*exp(mu*WP$times+sigma*WP$X[N,])

Pt = matrix(0,nrow=N,ncol= n+1)
```

```
Pt[N,] = S0*pnorm(  ( log(S0/c) + (rho + (sigma^2)/2 )*WP$times ) /
(sigma*sqrt(WP$times)) )  -c*exp(-rho*WP$times)*pnorm( ( log(S0/c) +
(rho - (sigma^2)/2  )*WP$times )/ (sigma*sqrt(WP$times) ) )
Pt[1,1] = Pt[1,2]

lines(WP$times,  Pt[N,], col = 4)

# Increasing value of sigma to sqrt(0.04)
sigma = sqrt(0.04) # = 0.2

mu = rho - (sigma^2)/2

St = matrix(0,nrow = N, ncol = n + 1)

St[N,] = S0*exp(mu*WP$times+sigma*WP$X[N,])

Pt = matrix(0,nrow=N,ncol= n+1)

Pt[N,] = S0*pnorm(  ( log(S0/c) + (rho + (sigma^2)/2 )*WP$times ) /
(sigma*sqrt(WP$times)) )  -c*exp(-rho*WP$times)*pnorm( ( log(S0/c) +
(rho - (sigma^2)/2  )*WP$times )/ (sigma*sqrt(WP$times) ) )

Pt[1,1] = Pt[1,2]

lines(WP$times,  Pt[N,], col = 5)

# legend
op = par(cex = 0.7)

legend(4,0.075,legend = c(expression(paste(sigma, " = sqrt(0.005)")),
expression(paste(sigma," = sqrt(0.01)"))),col=c("magenta", "red"), lty =
1, title = expression(paste("Values of ", sigma)))
legend(6,0.075,legend = c(expression(paste(sigma, " = sqrt(0.02)")),
expression(paste(sigma," = sqrt(0.03)")), expression(paste(sigma," =
sqrt(0.04)"))),col=c("black", "blue", "cyan"), lty = 1)




            # Varying value of c (strike price)

c = 1

sigma = sqrt(0.02)

mu = rho - (sigma^2)/2

St = matrix(0,nrow = N, ncol = n + 1)

St[N,] = S0*exp(mu*WP$times+sigma*WP$X[N,])

Pt = matrix(0,nrow=N,ncol= n+1)
```

```
Pt[N,] = S0*pnorm(  ( log(S0/c) + (rho + (sigma^2)/2 )*WP$times ) /
(sigma*sqrt(WP$times)) )  -c*exp(-rho*WP$times)*pnorm( ( log(S0/c) +
(rho - (sigma^2)/2  )*WP$times )/ (sigma*sqrt(WP$times) ) )

Pt[1,1] = Pt[1,2]

# Plot of P_t against time
plot(range(WP$times), range(Pt),type="n", main = "Figure 3: Effect of
varying the strike price on the price of a European call option", xlab =
"Time", ylab = "P_t")

lines(WP$times, Pt[N,])


# Varying value of c
# reducing value of c to 0.95
c = 0.95

Pt = matrix(0,nrow=N,ncol= n+1)

Pt[N,] = S0*pnorm(  ( log(S0/c) + (rho + (sigma^2)/2 )*WP$times ) /
(sigma*sqrt(WP$times)) )  -c*exp(-rho*WP$times)*pnorm( ( log(S0/c) +
(rho - (sigma^2)/2  )*WP$times )/ (sigma*sqrt(WP$times) ) )

Pt[1,1] = Pt[1,2]

lines(WP$times,  Pt[N,], col = 2)

# reducing value of c to 0.90
c = 0.90

Pt = matrix(0,nrow=N,ncol= n+1)

Pt[N,] = S0*pnorm(  ( log(S0/c) + (rho + (sigma^2)/2 )*WP$times ) /
(sigma*sqrt(WP$times)) )  -c*exp(-rho*WP$times)*pnorm( ( log(S0/c) +
(rho - (sigma^2)/2  )*WP$times )/ (sigma*sqrt(WP$times) ) )

Pt[1,1] = Pt[1,2]

lines(WP$times,  Pt[N,], col = "magenta")

# Increasing value of c to 1.05
c = 1.05

Pt = matrix(0,nrow=N,ncol= n+1)

Pt[N,] = S0*pnorm(  ( log(S0/c) + (rho + (sigma^2)/2 )*WP$times ) /
(sigma*sqrt(WP$times)) )  -c*exp(-rho*WP$times)*pnorm( ( log(S0/c) +
(rho - (sigma^2)/2  )*WP$times )/ (sigma*sqrt(WP$times) ) )

Pt[1,1] = Pt[1,2]

lines(WP$times,  Pt[N,], col = 4)
```

```
# Increasing value of c to 1.10
c = 1.10

Pt = matrix(0,nrow=N,ncol= n+1)

Pt[N,] = S0*pnorm(  ( log(S0/c) + (rho + (sigma^2)/2 )*WP$times ) /
(sigma*sqrt(WP$times)) )  -c*exp(-rho*WP$times)*pnorm( ( log(S0/c) +
(rho - (sigma^2)/2  )*WP$times )/ (sigma*sqrt(WP$times) ) )

Pt[1,1] = Pt[1,2]

lines(WP$times,  Pt[N,], col = 5)

# legend
op = par(cex = 0.7)

legend(4,0.075,legend = c("c = 0.90", "c = 0.95"),col=c("magenta",
"red"), lty = 1, title = "Values of c")
legend(6,0.075,legend = c("c = 1.00", "c = 1.05", "c =
1.10"),col=c("black", "blue", "cyan"), lty = 1)




               # Varying value of rho

# resetting rho to 0.03 and c = 1
rho = 0.03

c = 1

mu = rho - (sigma^2)/2

St = matrix(0,nrow = N, ncol = n + 1)

St[N,] = S0*exp(mu*WP$times+sigma*WP$X[N,])

Pt = matrix(0,nrow=N,ncol= n+1)

Pt[N,] = S0*pnorm(  ( log(S0/c) + (rho + (sigma^2)/2 )*WP$times ) /
(sigma*sqrt(WP$times)) )  -c*exp(-rho*WP$times)*pnorm( ( log(S0/c) +
(rho - (sigma^2)/2  )*WP$times )/ (sigma*sqrt(WP$times) ) )

Pt[1,1] = Pt[1,2]

plot(range(WP$times), range(Pt),type="n", main = "Figure 4: Effect of
varying the interest rate on the price of a European call option", xlab
= "Time", ylab = "P_t")
lines(WP$times, Pt[N,])


# Decreasing rho
# decreasing rho to 0.02
```

```
rho = 0.02

mu = rho - (sigma^2)/2

St = matrix(0,nrow = N, ncol = n + 1)

St[N,] = S0*exp(mu*WP$times+sigma*WP$X[N,])

Pt = matrix(0,nrow=N,ncol= n+1)

Pt[N,] = S0*pnorm(  ( log(S0/c) + (rho + (sigma^2)/2 )*WP$times ) /
(sigma*sqrt(WP$times)) )  -c*exp(-rho*WP$times)*pnorm( ( log(S0/c) +
(rho - (sigma^2)/2  )*WP$times )/ (sigma*sqrt(WP$times) ) )

Pt[1,1] = Pt[1,2]

lines(WP$times,  Pt[N,], col = 2)

# decreasing rho to 0.01
rho = 0.01

mu = rho - (sigma^2)/2

St = matrix(0,nrow = N, ncol = n + 1)

St[N,] = S0*exp(mu*WP$times+sigma*WP$X[N,])

Pt = matrix(0,nrow=N,ncol= n+1)

Pt[N,] = S0*pnorm(  ( log(S0/c) + (rho + (sigma^2)/2 )*WP$times ) /
(sigma*sqrt(WP$times)) )  -c*exp(-rho*WP$times)*pnorm( ( log(S0/c) +
(rho - (sigma^2)/2  )*WP$times )/ (sigma*sqrt(WP$times) ) )

Pt[1,1] = Pt[1,2]

lines(WP$times,  Pt[N,], col = "magenta")

# Increasing rho

# increasing rho to 0.04
rho = 0.04

mu = rho - (sigma^2)/2

St = matrix(0,nrow = N, ncol = n + 1)

St[N,] = S0*exp(mu*WP$times+sigma*WP$X[N,]) # price of stock at time t

Pt = matrix(0,nrow=N,ncol= n+1) # matrix to store realsiation

Pt[N,] = S0*pnorm(  ( log(S0/c) + (rho + (sigma^2)/2 )*WP$times ) /
(sigma*sqrt(WP$times)) )  -c*exp(-rho*WP$times)*pnorm( ( log(S0/c) +
(rho - (sigma^2)/2  )*WP$times )/ (sigma*sqrt(WP$times) ) )
```

```
Pt[1,1] = Pt[1,2]

lines(WP$times,  Pt[N,], col = 4)

# increasing rho to 0.05

rho = 0.05

mu = rho - (sigma^2)/2

St = matrix(0,nrow = N, ncol = n + 1)

St[N,] = S0*exp(mu*WP$times+sigma*WP$X[N,]) # price of stock at time t

Pt = matrix(0,nrow=N,ncol= n+1) # matrix to store realsiation

Pt[N,] = S0*pnorm(  ( log(S0/c) + (rho + (sigma^2)/2 )*WP$times ) /
(sigma*sqrt(WP$times)) )  -c*exp(-rho*WP$times)*pnorm( ( log(S0/c) +
(rho - (sigma^2)/2  )*WP$times )/ (sigma*sqrt(WP$times) ) )

Pt[1,1] = Pt[1,2]

lines(WP$times,  Pt[N,], col = 5)

op = par(cex = 0.7)

legend(4,0.075,legend =c(expression(paste(rho, " = 0.01")),
expression(paste(rho," = 0.02"))),col=c("magenta", "red"), lty = 1,
title = expression(paste("Values of ", rho)))
legend(6,0.075,legend =c(expression(paste(rho, " = 0.03")),
expression(paste(rho," = 0.04")), expression(paste(rho," =
0.05"))),col=c("black", "blue", "cyan"), lty = 1)
# Studying the effect of the initial stock price on the price of a call
option

# The Black-Scholes model for stock prices

N = 1
Delta = 0.01
n = 1000 # Delta*n = 10

WP = rWiener(n,N,Delta)  # 1 realisation of the standard Wiener Process

# Transformation of the standard Wiener Process

S0 = 1 # initial price of the stock
sigma = sqrt(0.02) # = 0.14142 - volatility
rho = 0.03 # interest rate
c = 1 # strike price
mu = rho - (sigma^2)/2  # implies the expected discounted stock price is
constant
# Value of mu is 0.02 (mu is the drift rate)
```

```
St = matrix(0,nrow = N, ncol = n + 1) # matrix to store realisation

St[N,] = S0*exp(mu*WP$times+sigma*WP$X[N,]) # price of stock at time t


Pt = matrix(0,nrow=N,ncol= n+1) # matrix to store realisation

Pt[N,] = S0*pnorm(  ( log(S0/c) + (rho + (sigma^2)/2 )*WP$times ) /
(sigma*sqrt(WP$times)) )  -c*exp(-rho*WP$times)*pnorm( ( log(S0/c) +
(rho - (sigma^2)/2  )*WP$times )/ (sigma*sqrt(WP$times) ) ) # the price
at timet of a European call option

Pt[1,1] = Pt[1,2] # when t = 0, R calculates P0= 0/0 = NaN, so to enable
R to calculate range(Pt) to plot
# Pt against t, we set value of Pt[1,1] equal to value of Pt[2,1]

# Plot of P_t against time
plot(range(WP$times), range(Pt),type="n", main = "Figure 5: Price of a
European call option against time ", xlab = "Time", ylab = "P_t")
lines(WP$times, Pt[N,])



# Varying S0

plot(range(WP$times), range(Pt),type="n", main = "Effect of varying the
initial price of the stock on the price of a European call option" ,
xlab = "Time", ylab = "P_t")
lines(WP$times, Pt[N,])

# reducing value of S0

# reducing value of S0 to 0.95



S0 = 0.95

St = matrix(0,nrow = N, ncol = n + 1)

St[N,] = S0*exp(mu*WP$times+sigma*WP$X[N,])

Pt = matrix(0,nrow=N,ncol= n+1)

Pt[N,] = S0*pnorm(  ( log(S0/c) + (rho + (sigma^2)/2 )*WP$times ) /
(sigma*sqrt(WP$times)) )  -c*exp(-rho*WP$times)*pnorm( ( log(S0/c) +
(rho - (sigma^2)/2  )*WP$times )/ (sigma*sqrt(WP$times) ) )

Pt[1,1] = Pt[1,2]

lines(WP$times,  Pt[N,], col = 2)

# reducing value of S0 to 0.90
S0 = 0.90
```

```
St = matrix(0,nrow = N, ncol = n + 1)

St[N,] = S0*exp(mu*WP$times+sigma*WP$X[N,])

Pt = matrix(0,nrow=N,ncol= n+1)

Pt[N,] = S0*pnorm(  ( log(S0/c) + (rho + (sigma^2)/2 )*WP$times ) /
(sigma*sqrt(WP$times)) )  -c*exp(-rho*WP$times)*pnorm( ( log(S0/c) +
(rho - (sigma^2)/2  )*WP$times )/ (sigma*sqrt(WP$times) ) )

Pt[1,1] = Pt[1,2]

lines(WP$times,  Pt[N,], col = "magenta")


# Increasing value of S0 to 1.05
S0 = 1.05


St = matrix(0,nrow = N, ncol = n + 1)

St[N,] = S0*exp(mu*WP$times+sigma*WP$X[N,])

Pt = matrix(0,nrow=N,ncol= n+1)

Pt[N,] = S0*pnorm(  ( log(S0/c) + (rho + (sigma^2)/2 )*WP$times ) /
(sigma*sqrt(WP$times)) )  -c*exp(-rho*WP$times)*pnorm( ( log(S0/c) +
(rho - (sigma^2)/2  )*WP$times )/ (sigma*sqrt(WP$times) ) )
Pt[1,1] = Pt[1,2]

lines(WP$times,  Pt[N,], col = 4)

# Increasing value of S0 to 1.1

S0 = 1.1


St = matrix(0,nrow = N, ncol = n + 1)

St[N,] = S0*exp(mu*WP$times+sigma*WP$X[N,])

Pt = matrix(0,nrow=N,ncol= n+1)

Pt[N,] = S0*pnorm(  ( log(S0/c) + (rho + (sigma^2)/2 )*WP$times ) /
(sigma*sqrt(WP$times)) )  -c*exp(-rho*WP$times)*pnorm( ( log(S0/c) +
(rho - (sigma^2)/2  )*WP$times )/ (sigma*sqrt(WP$times) ) )

Pt[1,1] = Pt[1,2]

lines(WP$times,  Pt[N,], col = 5)

# legend
```

```
op = par(cex = 0.7)

legend(8,0.055,legend = c(0.90,0.95),col=c("magenta", "red"), lty = 1,
title = "Values of S0")
legend(9.2,0.055,legend = c(1,1.05,1.1),col=c("black", "blue", "cyan"),
lty = 1)
```

R code used for the Vasicek model for interest rates

```
rOU = function(n,N,Delta,theta, sigma) {
  times = (0:n)*Delta    ## Vector of t_0, t_1, ..., t_n
  X = matrix(0, nrow = N, ncol = n+1)

  for (i in 1:n) {
    x = X[,i] # current value
    m = x*exp(-theta*Delta) # mean of the new value
    v = (sigma^2)*(1 - exp(-2*theta*Delta)) / (2*theta) ## Variance of
new value
    X[,i+1] = rnorm(N,m,sqrt(v)) # simulate new value
  }
  return(list(X=X, times =times))
}

N = 10
n = 1000
Delta = 0.01
theta = 0.5
sigma = 0.02

OU = rOU(n, N, Delta, theta, sigma)


# now transform

R0 = 0.1
mu = 0.05

# storage for R_s
Rs = matrix(0,nrow = N, ncol = n + 1)
# Loop to trandform OU to R_s
for (i in 1:N) {
  Rs[i,] = exp(-theta*OU$times)*R0 + (1 - exp(-theta*OU$times))*mu +
OU$X[i,]
}

## plot
colors = rainbow(10)
rev(colors)
plot(range(OU$times), range(Rs), type="n", xlab = "time", ylab = "R_s",
main ="Figure 1: 10 realisations of the spot-rate process (R_s) over
time (s)")
for (i in 1:N) {
  lines(OU$times, Rs[i,],  col =colors[i])
}


op = par(cex = 0.7)

legend( 3.48, 0.14, legend = c(1,2,3,4,5 ), col= c(colors[1],colors[2],
colors[3], colors[4], colors[5]), title = "Value of N", lty = 1 )
```

```
legend( 4.54,0.134,legend = c(6,7,8,9,10),  col= c(colors[6],colors[7],
colors[8], colors[9], colors[10]), lty = 1 )



plot(range(OU$times), range(Rs), type="n", xlab = "time", ylab = "R_s",
main ="Figure 2: The first realisation of the spot-rate process")
lines(OU$times,Rs[1,],col= col[1])

op = par(cex = 0.7)

legend( 9, 0.137, legend = c("N = 1","R_0 = 0.1",
expression(paste(theta, "= 0.5")), expression(paste(mu, "= 0.05")),
expression(paste(sigma, "= 0.02")) ))



# Plotting E(Rs)

theta = 0.5

E_Rs = matrix(0,nrow = N, ncol = n + 1)
for (i in 1:N) {
 E_Rs[i,] = exp(-theta*OU$times)*R0 + (1 - exp(-theta*OU$times))*mu
}

col_6 = rainbow(6)

#plot(range(OU$times), range(E_Rs), type="n", main = "Figure 3: The
expected value of the spot-rate process", xlab = "time", ylab =
"E(R_s)")
#for (i in 1:N) {
  #lines(OU$times, E_Rs[i,],  col = col_5[3])
#}

E = matrix(0,nrow = 5, ncol = n + 1)
E[3,] = E_Rs[1,]

# reduction of theta
theta = 0.25
E[2,] = exp(-theta*OU$times)*R0 + (1 - exp(-theta*OU$times))*mu
# lines(OU$times, E[2,],  col = col_5[4])

theta = 0.125
E[1,] = exp(-theta*OU$times)*R0 + (1 - exp(-theta*OU$times))*mu
# lines(OU$times, E[1,], col = col_5[5])

# increasing theta

theta = 1
E[4,] = exp(-theta*OU$times)*R0 + (1 - exp(-theta*OU$times))*mu
```

```
# lines(OU$times, E[1,],  col =col_5[2])


theta = 2
E[5,] = exp(-theta*OU$times)*R0 + (1 - exp(-theta*OU$times))*mu
# lines(OU$times, E[1,],  col =col_5[1])

mu_vec = rep(mu,n+1)


op = par(cex = 0.7)

plot(range(OU$times), range(E), type="n", main = "Figure 3: The expected
value of the spot-rate process", xlab = "time", ylab = "E(R_s)")
for (i in 1:5) {
  lines(OU$times, E[i,],  col =col_6[i])
}
lines(OU$times,mu_vec, lty=3, col = col_6[6])

legend( 8.90,0.101,legend = c( expression(paste(theta, "= 0.125")),
expression(paste(theta," = 0.25")), expression(paste(theta," =
0.5")),expression(paste(theta," = 1")), expression(paste(theta," = 2")),
expression(paste(mu, " = 0.05")) ),col= c(col_6[1],col_6[2], col_6[3],
col_6[4], col_6[5], col_6[6]), lty = c(1,1,1,1,1,2) )

# -----------------------------------------------------------------

# Plotting the V(R_s)
sigma = 0.02
theta = 0.5
V_Rs = matrix(0,nrow = N, ncol = n + 1)

for (i in 1:N) {
  V_Rs[i,] = ((sigma^2)/(2*theta))*(1 - exp(-2*theta*OU$times))
}

y = rep(((sigma)^2)/(2*theta),n+1)


V = matrix(0,nrow = 5, ncol = n + 1)
Y = matrix(5,nrow=5,ncol=n+1)

V[3,] = V_Rs[3,]
Y[3,] = y

# reducing sigma
sigma = 0.01
V[2,] = ((sigma^2)/(2*theta))*(1 - exp(-2*theta*OU$times))


y = rep(((sigma)^2)/(2*theta),n+1)

Y[2,] = y
```

```
sigma = 0.005
V[1,] = ((sigma^2)/(2*theta))*(1 - exp(-2*theta*OU$times))

y = rep(((sigma)^2)/(2*theta),n+1)
Y[1,] = y



# increasing sigma
sigma = 0.03
V[4,] = ((sigma^2)/(2*theta))*(1 - exp(-2*theta*OU$times))

y = rep(((sigma)^2)/(2*theta),n+1)
Y[4,] = y



sigma = 0.04
V[5,] = ((sigma^2)/(2*theta))*(1 - exp(-2*theta*OU$times))

y = rep(((sigma)^2)/(2*theta),n+1)
Y[5,] = y


col_5 = rainbow(5)
op = par(cex = 0.7)

plot(range(OU$times), range(V), type="n", main = "Figure 4: The variance
of the spot-rate process", xlab = "time", ylab = "V(R_s)")
for (i in 1:5) {
  lines(OU$times, V[i,],  col =col_5[i])
  lines(OU$times,Y[i,],col = col_5[i], lty = 3)
}



legend( 6.90, 0.001475, legend = c( expression(paste(sigma, "1 =
0.005")), expression(paste(sigma,"2 = 0.01")), expression(paste(sigma,"3
= 0.02")), expression(paste(sigma,"4 = 0.03")),
expression(paste(sigma,"5 = 0.04")) ), col= c(col_5[1],col_5[2],
col_5[3], col_5[4], col_5[5]), lty = 1 )
legend( 8.50,0.00155,legend = c( expression(paste("((",sigma, "1)^2) /
(2a)")),  expression(paste("(",sigma, "2)^2 / 2a")),
expression(paste("(",sigma, "3)^2/(2a)")),  expression(paste("(",sigma,
"4)^2/(2a)")), expression(paste("(",sigma, "5)^2/(2a)"))),  col=
c(col_5[1],col_5[2], col_5[3], col_5[4], col_5[5]), title = "Limit of
V(R_s)", lty = 3 )



# Integrated Ornstein-Uhlenbeck process
```

```
N = 1000
n = 100   # n*Delta = 100*0.01 = 1
Delta = 0.01
theta = 0.5
sigma = 0.02

OU = rOU(n, N, Delta, theta, sigma)

Y = rep(0,N) ## Vector to store realisations

for (i in 1:N) {
      Y[i] = Delta*sum(OU$X[i,2:(n+1)])
}

Q = rep(0,N)
for (i in 1:N) {
  Q[i] = exp(-Y[i])
}

hist(Y, main = "Figure 2: Histogram")
qqnorm(Y, main = "Figure 3: Normal Q-Q Plot")  # normal qq-plot
qqline(Y, col = "red")


mean(Y)
[1] 0.0005101343
var(Y)
[1] 9.052268e-05

> mean(Q)
[1] 0.9995352
> var(Q)
[1] 9.039525e-05

hist(Y, main = "Figure 2: Histogram")
qqnorm(Y, main = "Figure 3: Normal Q-Q Plot")  # normal qq-plot
qqline(Y, col = "red")



> mean(Q)
[1] 1.000329
> var(Q)
[1] 8.972977e-05

N = 1000
n = 1000
Delta = 0.01
OU = rOU(n, N, Delta, theta, sigma)

Qt = matrix(0,nrow=N,ncol=n+1) # Matrix to store realisations
## loop to calculate Q
for (i in 1:N) {
  Qt[i,] = exp(-Delta*cumsum(OU$X[i,]))
```

```
}


#  PLOT

op = par(cex = 0.7)
color = rainbow(10)
plot( range(OU$times), range(Qt), type ="n",main ="Figure 1: The bond
price at time 0 of a bond paying one unit at time t", xlab = "time",
ylab = "Q_t")
for (i in 1:10) {
  lines(OU$times,Qt[i,], col = color[i])
}

legend( -0.3, 1.5, legend = c(1,2,3,4,5 ), col= c(colors[1],colors[2],
colors[3], colors[4], colors[5]), title = "Value of N", lty = 1 )
legend(0.8, 1.465, legend = c(6,7,8,9,10), col= c(colors[6],colors[7],
colors[8], colors[9], colors[10]), lty = 1 )

legend( 1.9, 1.43,legend = c("R_0 = 0.1", expression(paste(theta, "=
0.5")), expression(paste(mu, "= 0.05")), expression(paste(sigma, "=
0.02")) ))
```