Documentation

Joris Baiutti & Claudio Polo

June 4, 2018

Load Libraries

```
library(Quandl)
library(ggplot2)
library(fOptions)
```

Data https://www.quandl.com/data/EURONEXT-Euronext-Stock-Exchange

Introduction

This document goes through a Monte Carlo Simulation of Deutsche Telekom, Volkswagen and Airbus closing stock prices.

Calculation of standard deviation and mean

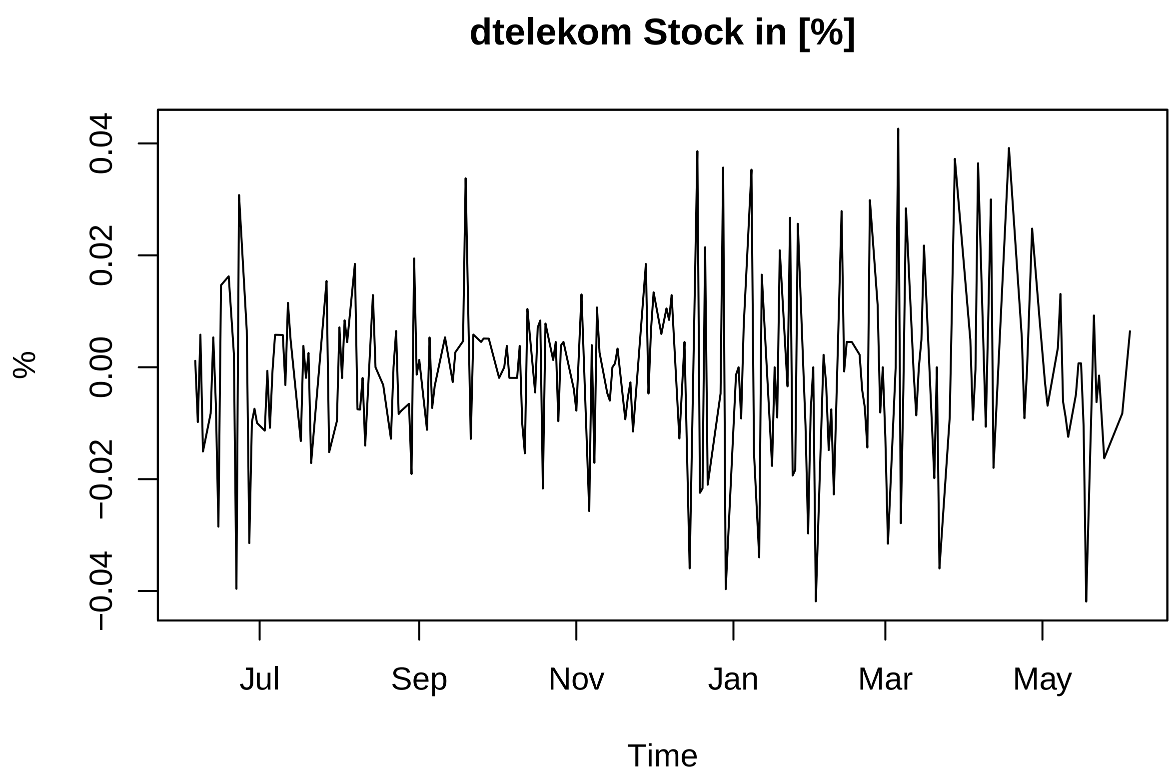
For the standard deviation we collect the percentual rate values. These values can be optained directly with quandle. Then we can calculate standard deviation and mean of those rate values.

Deutsche Telekom

Percentual rates

In this graph we can see the percentual daily rates over a year.

```
dtelekomclosing = Quandl('EURONEXT/DTEL',start_date='2017-06-04', type='zoo',
                             end_date='2018-06-04',transform = "rdiff")[,"Last"]
plot(dtelekomclosing,ylab="%",xlab="Time",main="dtelekom Stock in [%]")
```



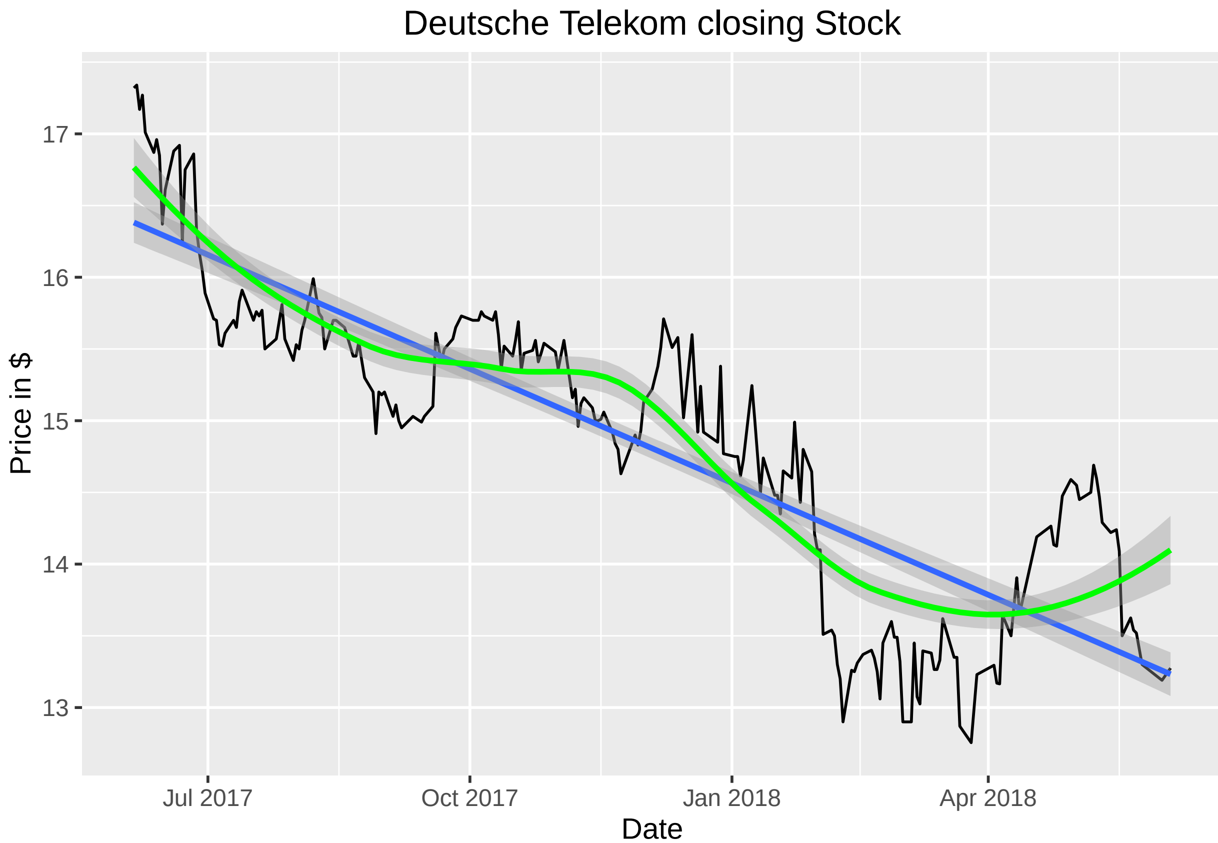

```
dtelekomsd = sd(dtelekomclosing)
dtelekommean = mean(dtelekomclosing)
```

Arithmetic mean and standard deviation

```
dtelekommatrix = matrix(c(dtelekomsd,dtelekommean),ncol=2,byrow=TRUE)
colnames(dtelekommatrix) = c("Standard deviation","mean")
print.table(dtelekommatrix)
```

```
##      Standard deviation         mean
## [1,]        0.015379015 -0.001124455
```

```r
dtelekom = Quandl('EURONEXT/DTEL', start_date='2017-06-04', end_date='2018-06-04')
ggplot(dtelekom, aes(Date, Last,)) + geom_line() + geom_smooth(method = "lm") +
geom_smooth(method = "loess", color="Green") +
labs(title="Deutsche Telekom closing Stock", y="Price in $", caption="Source: EURONEXT")
```



Brownian Motion

```r
nSim = 1000
nDays = 250
S0 = head(Quandl('EURONEXT/DTEL', start_date='2017-06-04', end_date='2018-06-04'), n=1)
S0 = S0$Last
S = matrix(0,nrow=nDays,ncol=nSim)
```

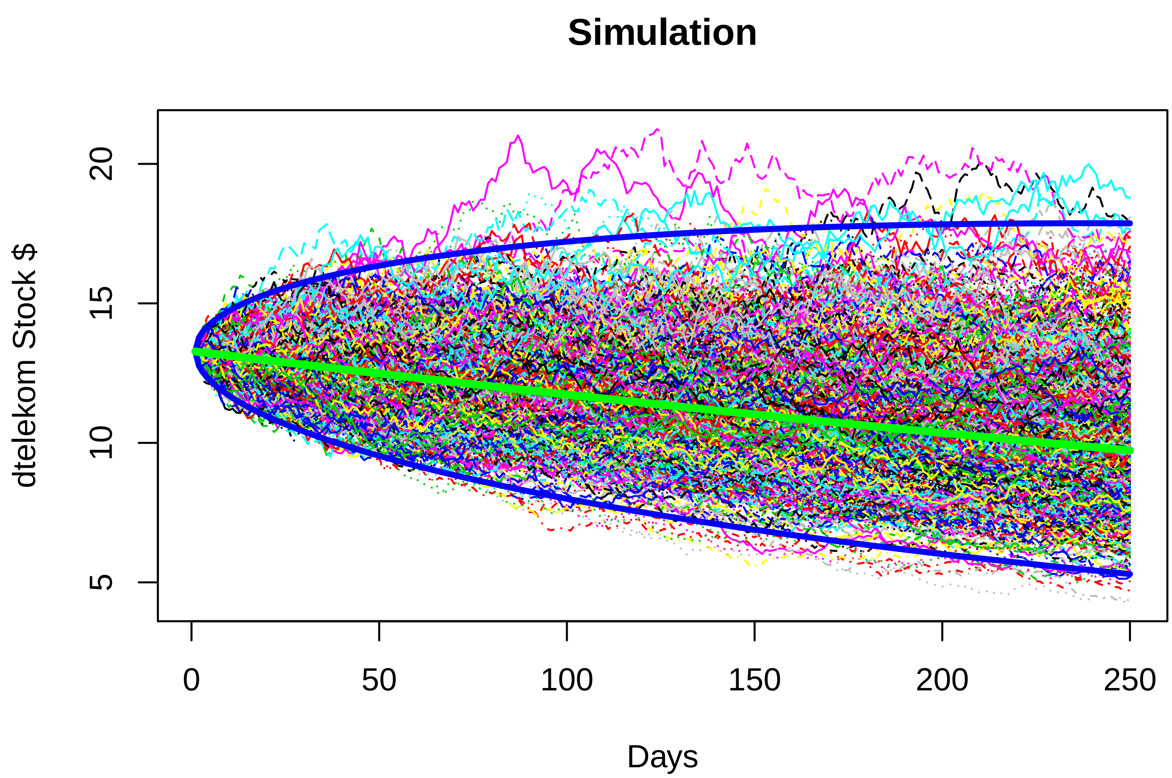
```r
for (i in 1:nSim) {
  SVec = rep(0, nDays)
  SVec[1] = S0
  for(j in 2:nDays) {
    pastDay = SVec[j-1]
    DeltaS = dtelekommean*pastDay + dtelekomsd*pastDay*rnorm(1)
    SVec[j] = pastDay+DeltaS
  }
  S[,i] = SVec
}
```

Calculation of SimulationS bOundaries

```
mean = dtelekommean * nDays
dtelekomsigma = dtelekomsd * sqrt(nDays)
t = seq(from=0,to=1,length.out=nDays)
meanBnd = vector(length = nDays)
upBnd = vector(length = nDays)
loBnd = vector(length = nDays)
test = vector(length = nDays)
for(i in 0:nDays){
  meanBnd[i] = S0*exp((( mean-dtelekomsigma^2/2 ))*t[i])
  upBnd[i] = S0*exp((( mean-dtelekomsigma^2/2 ))*t[i] + (dtelekomsigma * 2.5 * sqrt(t[i])))
  loBnd[i] = S0*exp((( mean-dtelekomsigma^2/2 ))*t[i] + (dtelekomsigma * -2.5 * sqrt(t[i])))
}
```

Presentation of the simulation.Green Line is the Mean of all simulations and the blue lines are the boundaries

```
matplot(S,type='l',col=1:100,ylab='dtelekom Stock $',xlab='Days',main='Simulation')
matlines(loBnd,lwd=3,col="blue")
matlines(upBnd,lwd=3,col="blue")
matlines(meanBnd,lwd=4,col="green")
```



Calculation of the Longnormal Distribution

```
lnMean = S0*exp(dtelekommean*nDays)
lnSD = S0*exp(dtelekommean*nDays)*sqrt(exp((dtelekomsd^2)*nDays)-1)
```

Mean over a year over all Simulations

```
cat('Mean',mean(S[nDays,]))
```

```
## Mean 10.08536
```

Standard deviation over a year over all Simulations

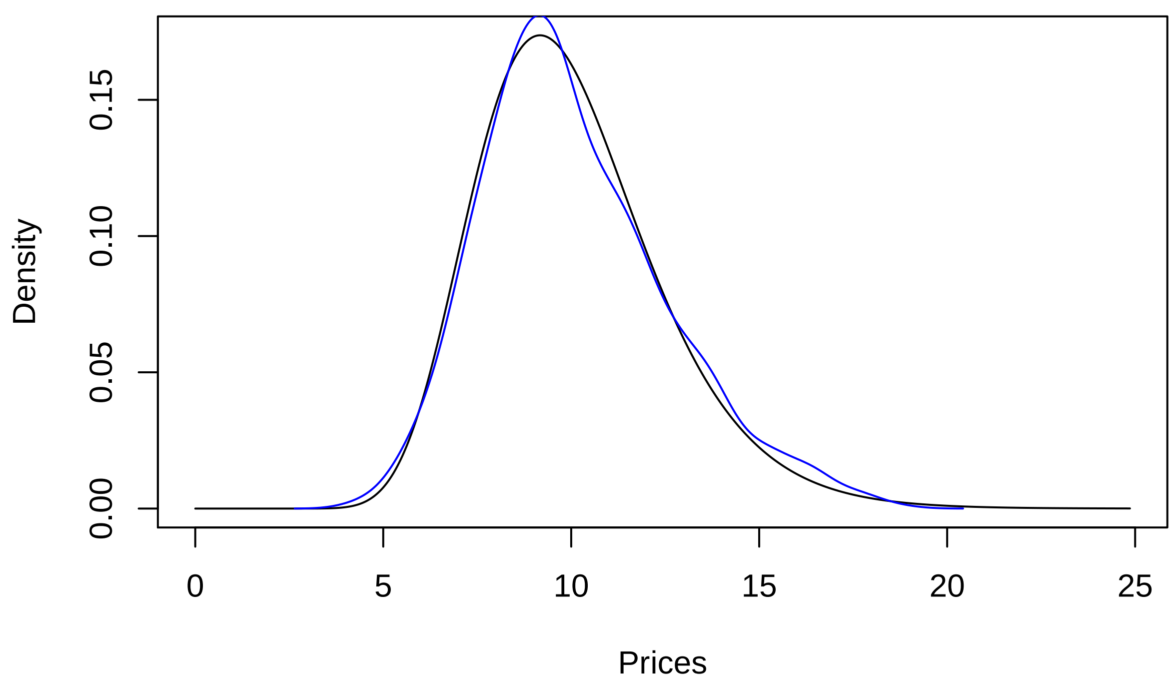
```
cat('Standard Deviation',lnSD,sd(S[nDays,]))
```

```
## Standard Deviation 2.473423 2.487804
```

Presentation of the longnormal distribution of theoretical density and empirical density

```
meanOfLog = log(S0) + (dtelekommean-(dtelekomsd^2)/2)*nDays
sdOfLog = dtelekomsd*sqrt(nDays)
priceGrid = seq(0,lnMean+6*lnSD,length=10000)
theoreticalDens = dlnorm(priceGrid,meanOfLog,sdOfLog)
empiricalDens = density(S[nDays,])

plot(priceGrid,theoreticalDens,type='l',xlab='Prices',ylab='Density')
lines(empiricalDens,col='blue')
```



Call Put Options price Black Scholes

```
openingS = head(Quandl('EURONEXT/DTEL', start_date='2017-06-04', end_date='2018-06-04'), n=1)
S = openingS$Open
strikecall = 15
optionCall = GBSOption(TypeFlag = "c", S, strikecall, 1/12, r = 0.03, b = 0.03, dtelekomsigma)
cat("OptionCall Price",optionCall@price)
```

```
## OptionCall Price 0.01811317
```

```r
strikeput = 12
optionPut = GBSOption(TypeFlag = "p", S, strikeput, 1/12, r = 0.03, b = 0.03, dtelekomsigma)
cat("OptionPut Price",optionPut@price)
```

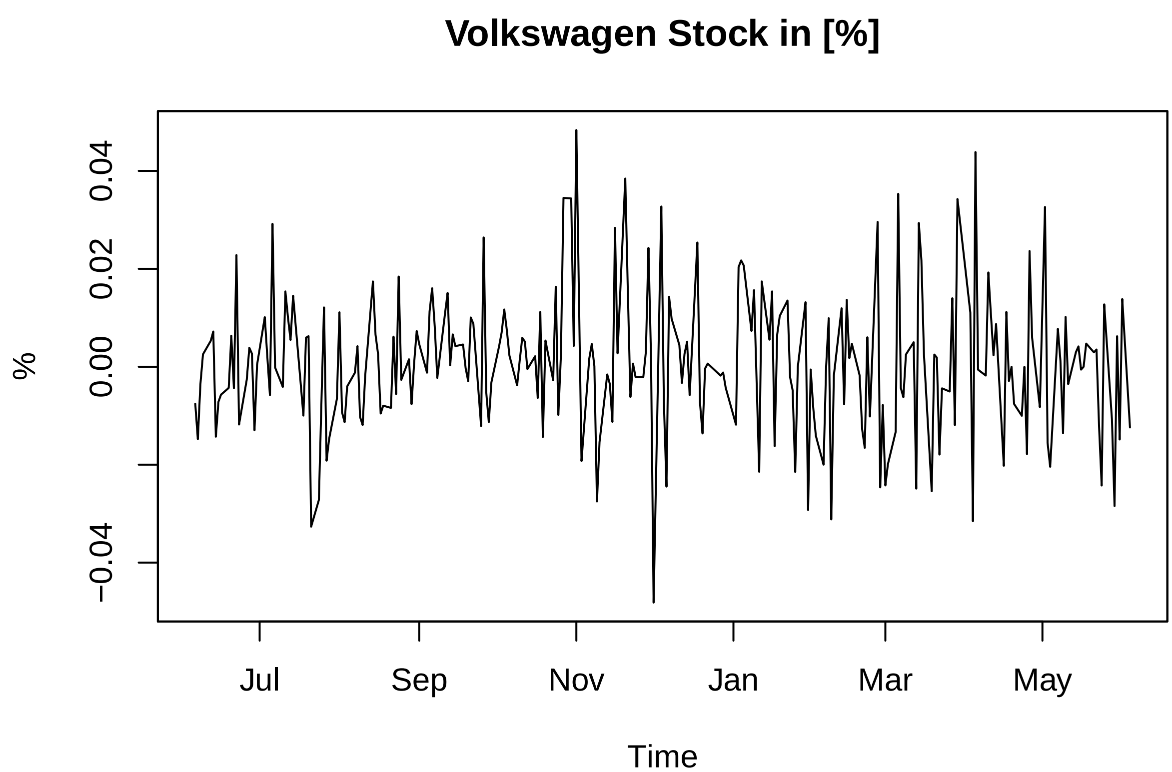
```
## OptionPut Price 0.027163
```

Volkswagen

Percentual rates

In this graph we can see the percentual daily rates over a year.

```r
volkswagenclosing = Quandl('EURONEXT/VWA',start_date='2017-06-04', type='zoo',end_date='2018-06-04',
                           transform = "rdiff")[,"Last"]
plot(volkswagenclosing,ylab="%",xlab="Time",main="Volkswagen Stock in [%]")
```



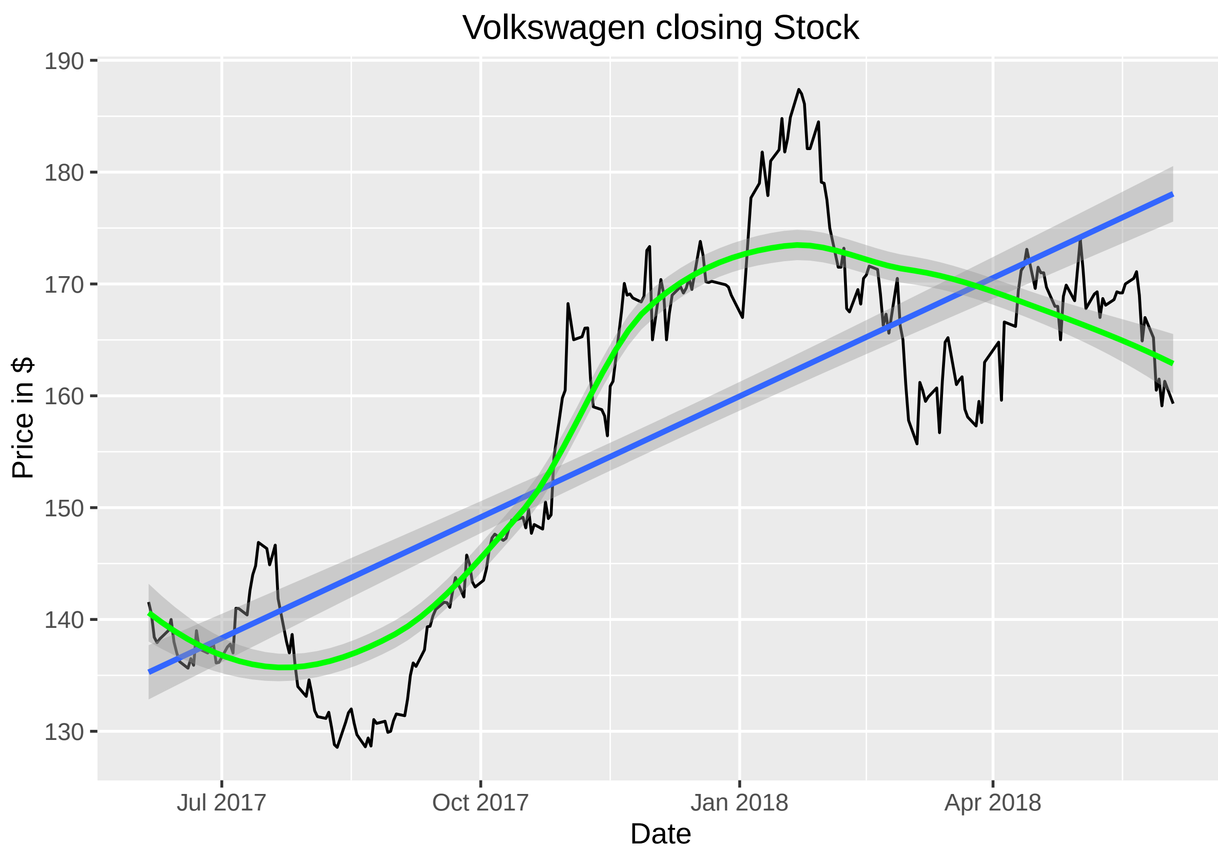

```r
volkswagensd = sd(volkswagenclosing)
volkswagenmean = mean(volkswagenclosing)
```

Arithmetic mean and standard deviation

```r
volkswagenmatrix = matrix(c(volkswagensd,volkswagenmean),ncol=2,byrow=TRUE)
colnames(volkswagenmatrix) = c("Standard deviation","mean")
print.table(volkswagenmatrix)
```

```
##      Standard deviation         mean
## [1,]       0.0143539191 0.0005709867
```

```
volkswagen = Quandl('EURONEXT/VWA', start_date='2017-06-04', end_date='2018-06-04')
ggplot(volkswagen, aes(Date, Last,)) + geom_line() + geom_smooth(method = "lm") +
geom_smooth(method = "loess", color="Green") + labs(title="Volkswagen closing Stock",
                                                 y="Price in $", caption="Source: EURONEXT")
```



Brownian Motion

```
nSim = 1000
nDays = 250
S0 = head(Quandl('EURONEXT/VWA', start_date='2017-06-04', end_date='2018-06-04'), n=1)
S0 = S0$Last
S = matrix(0,nrow=nDays,ncol=nSim)
```

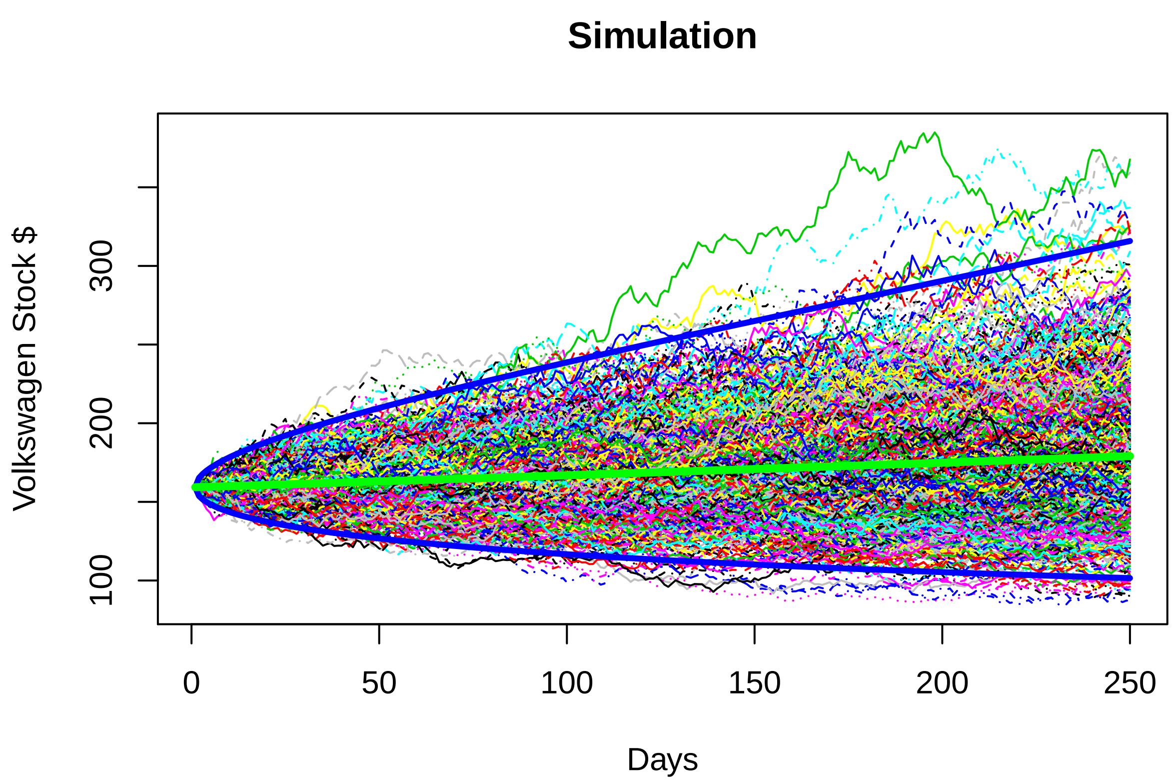
```
for (i in 1:nSim) {
  SVec = rep(0, nDays)
  SVec[1] = S0
  for(j in 2:nDays) {
    pastDay = SVec[j-1]
    DeltaS = volkswagenmean*pastDay + volkswagensd*pastDay*rnorm(1)
    SVec[j] = pastDay+DeltaS
  }
  S[,i] = SVec
}
```

Calculation of SimulationS bOundaries

```
mean = volkswagenmean * nDays
volkswagensigma = volkswagensd * sqrt(nDays)
t = seq(from=0,to=1,length.out=nDays)
meanBnd = vector(length = nDays)
upBnd = vector(length = nDays)
loBnd = vector(length = nDays)
test = vector(length = nDays)
for(i in 0:nDays){
  meanBnd[i] = S0*exp((( mean-volkswagensigma^2/2 ))*t[i])
  upBnd[i] = S0*exp((( mean-volkswagensigma^2/2 ))*t[i] + (volkswagensigma * 2.5 * sqrt(t[i])))
  loBnd[i] = S0*exp((( mean-volkswagensigma^2/2 ))*t[i] + (volkswagensigma * -2.5 * sqrt(t[i])))
}
```

Presentation of the simulation.Green Line is the Mean of all simulations and the blue lines are the boundaries

```
matplot(S,type='l',col=1:100,ylab='Volkswagen Stock $',xlab='Days',main='Simulation')
matlines(loBnd,lwd=3,col="blue")
matlines(upBnd,lwd=3,col="blue")
matlines(meanBnd,lwd=4,col="green")
```



Calculation of the Longnormal Distribution

```
lnMean = S0*exp(volkswagenmean*nDays)
lnSD = S0*exp(volkswagenmean*nDays)*sqrt(exp((volkswagensd^2)*nDays)-1)
```

Mean over a year over all Simulations

```r
cat('Mean',mean(S[nDays,]))
```

```
## Mean 182.7231
```

Standard deviation over a year over all Simulations

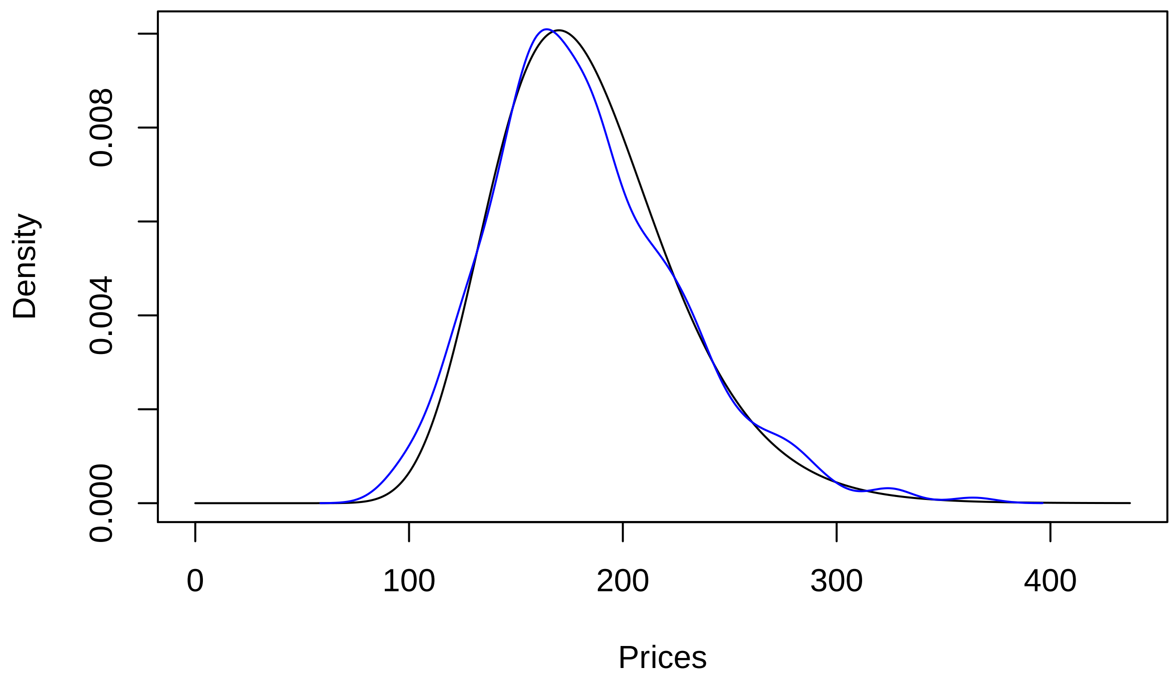
```r
cat('Standard Deviation',lnSD,sd(S[nDays,]))
```

```
## Standard Deviation 42.24418 44.45183
```

Presentation of the longnormal distribution of theoretical density and empirical density

```r
meanOfLog = log(S0) + (volkswagenmean-(volkswagensd^2)/2)*nDays
sdOfLog = volkswagensd*sqrt(nDays)
priceGrid = seq(0,lnMean+6*lnSD,length=10000)
theoreticalDens = dlnorm(priceGrid,meanOfLog,sdOfLog)
empiricalDens = density(S[nDays,])

plot(priceGrid,theoreticalDens,type='l',xlab='Prices',ylab='Density')
lines(empiricalDens,col='blue')
```



Call Put Options price Black Scholes

```r
openingS = head(Quandl('EURONEXT/VWA', start_date='2017-06-04', end_date='2018-06-04'), n=1)
S = openingS$Open
strikecall = 180
optionCall = GBSOption(TypeFlag = "c", S, strikecall, 1/12, r = 0.03, b = 0.03, volkswagensigma)
cat("OptionCall Price",optionCall@price)
```

```
## OptionCall Price 0.2723785
```

```r
strikeput = 150
optionPut = GBSOption(TypeFlag = "p", S, strikeput, 1/12, r = 0.03, b = 0.03, volkswagensigma)
cat("OptionPut Price",optionPut@price)
```

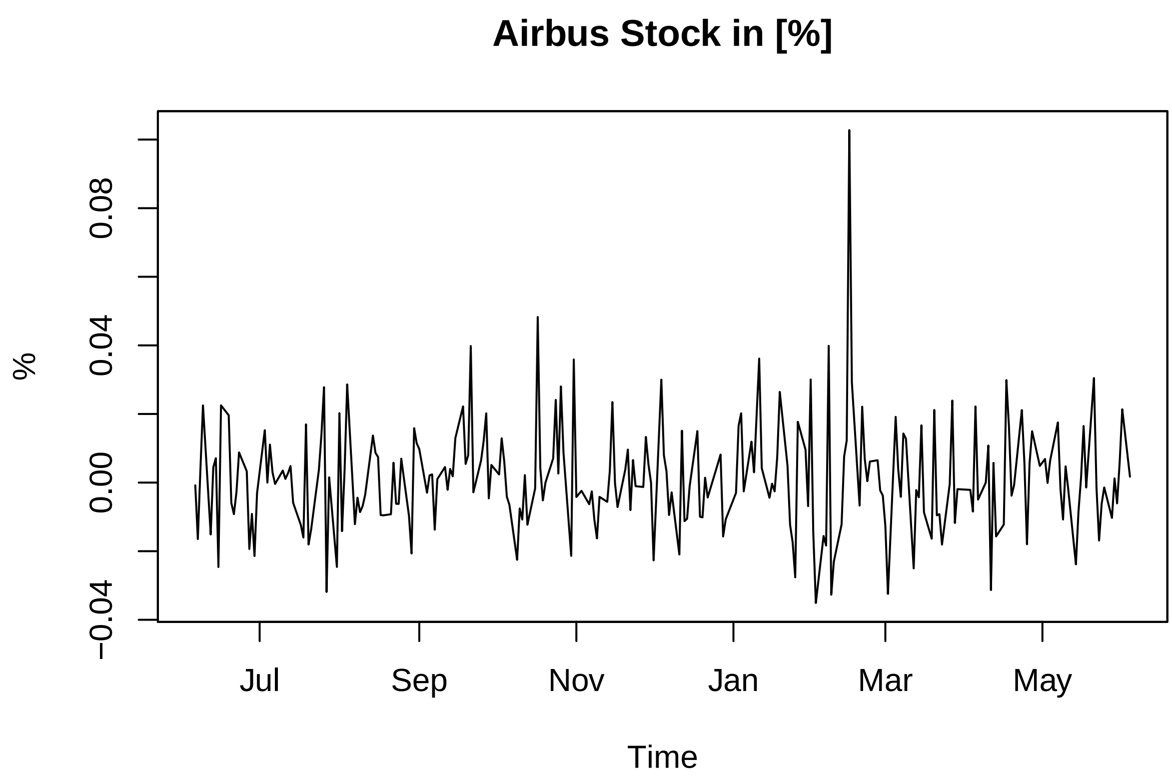
```
## OptionPut Price 0.5678754
```

Airbus

Percentual rates

In this graph we can see the percentual daily rates over a year.

```r
Airbusclosing = Quandl('EURONEXT/AIR',start_date='2017-06-04', type='zoo',
                       end_date='2018-06-04',transform = "rdiff")[,"Last"]
plot(Airbusclosing,ylab="%",xlab="Time",main="Airbus Stock in [%]")
```



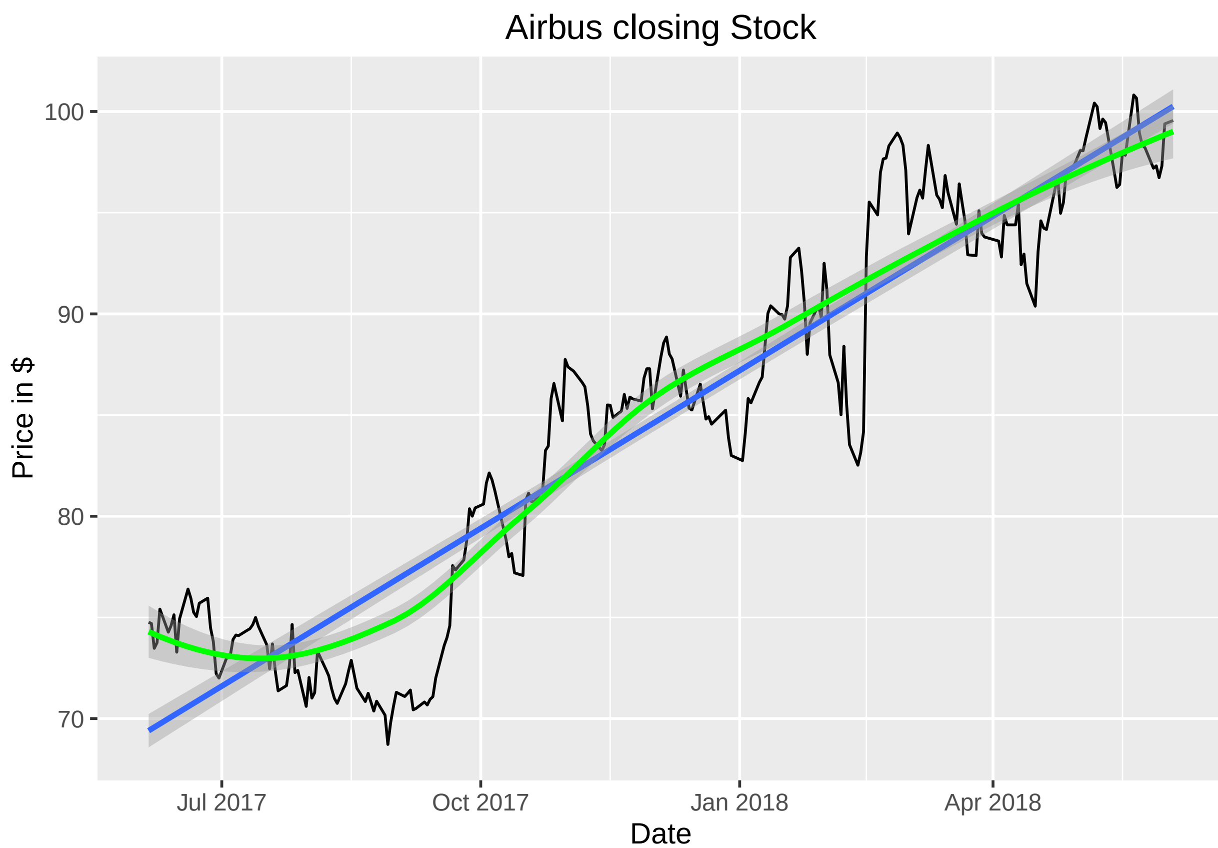

```r
Airbussd = sd(Airbusclosing)
Airbusmean = mean(Airbusclosing)
```

Arithmetic mean and standard deviation

```r
Airbusmatrix = matrix(c(Airbussd,Airbusmean),ncol=2,byrow=TRUE)
colnames(Airbusmatrix) = c("Standard deviation","mean")
print.table(Airbusmatrix)
```

```
##      Standard deviation        mean
## [1,]        0.015693199 0.001258491
```

```r
Airbus = Quandl('EURONEXT/AIR', start_date='2017-06-04', end_date='2018-06-04')
ggplot(Airbus, aes(Date, Last,)) + geom_line() + geom_smooth(method = "lm") +
geom_smooth(method = "loess", color="Green") + labs(title="Airbus closing Stock",
                                                    y="Price in $", caption="Source: EURONEXT")
```



Brownian Motion

```r
nSim = 1000
nDays = 250
S0 = head(Quandl('EURONEXT/AIR', start_date='2017-06-04', end_date='2018-06-04'), n=1)
S0 = S0$Last
S = matrix(0,nrow=nDays,ncol=nSim)
```

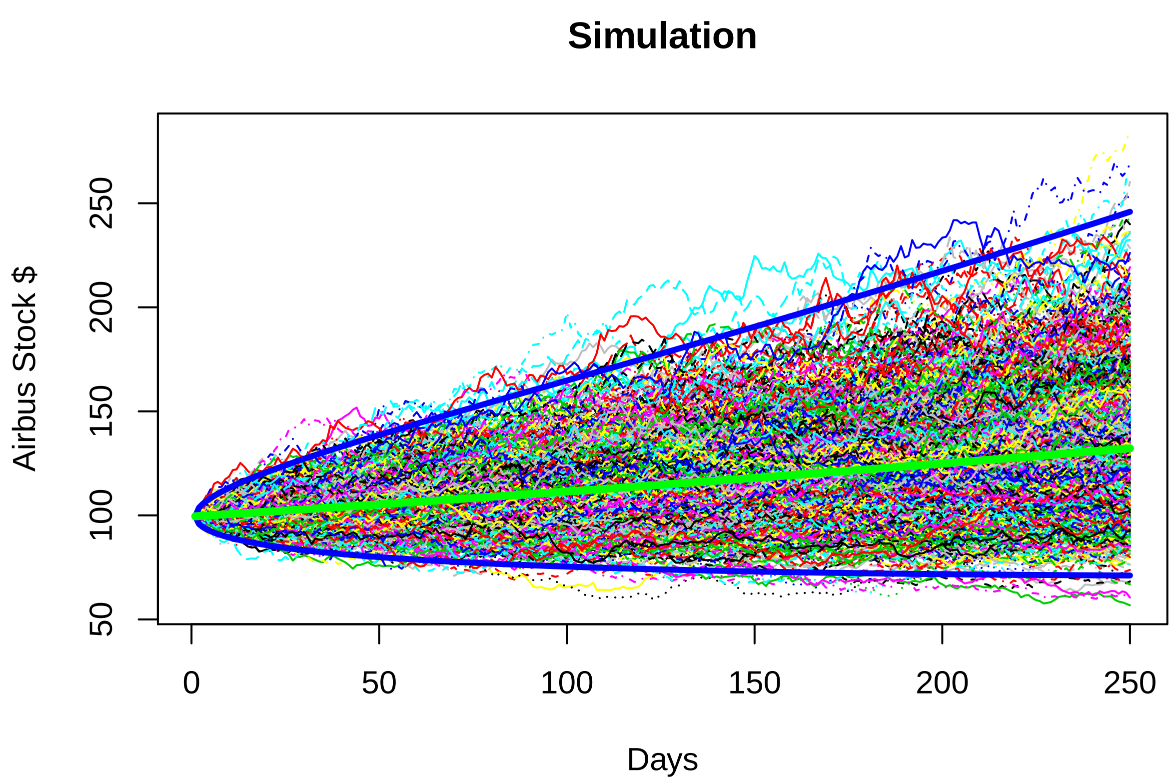
```r
for (i in 1:nSim) {
  SVec = rep(0, nDays)
  SVec[1] = S0
  for(j in 2:nDays) {
    pastDay = SVec[j-1]
    DeltaS = Airbusmean*pastDay + Airbussd*pastDay*rnorm(1)
    SVec[j] = pastDay+DeltaS
  }
  S[,i] = SVec
}
```

Calculation of SimulationS bOundaries

```
mean = Airbusmean * nDays
Airbussigma = Airbussd * sqrt(nDays)
t = seq(from=0,to=1,length.out=nDays)
meanBnd = vector(length = nDays)
upBnd = vector(length = nDays)
loBnd = vector(length = nDays)
test = vector(length = nDays)
for(i in 0:nDays){
  meanBnd[i] = S0*exp((( mean-Airbussigma^2/2 ))*t[i])
  upBnd[i] = S0*exp((( mean-Airbussigma^2/2 ))*t[i] + (Airbussigma * 2.5 * sqrt(t[i])))
  loBnd[i] = S0*exp((( mean-Airbussigma^2/2 ))*t[i] + (Airbussigma * -2.5 * sqrt(t[i])))
}
```

Presentation of the simulation.Green Line is the Mean of all simulations and the blue lines are the boundaries

```
matplot(S,type='l',col=1:100,ylab='Airbus Stock $',xlab='Days',main='Simulation')
matlines(loBnd,lwd=3,col="blue")
matlines(upBnd,lwd=3,col="blue")
matlines(meanBnd,lwd=4,col="green")
```



Calculation of the Longnormal Distribution

```
lnMean = S0*exp(Airbusmean*nDays)
lnSD = S0*exp(Airbusmean*nDays)*sqrt(exp((Airbussd^2)*nDays)-1)
```

Mean over a year over all Simulations

```r
cat('Mean',mean(S[nDays,]))
```

```
## Mean 137.0525
```

Standard deviation over a year over all Simulations

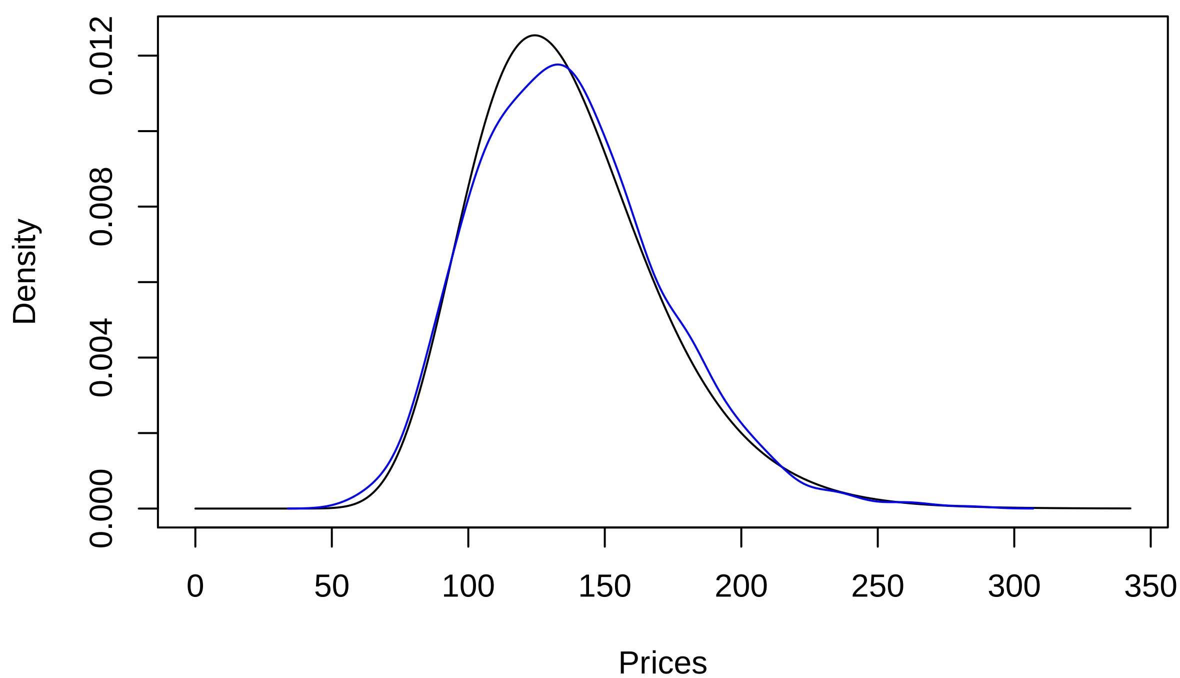
```r
cat('Standard Deviation',lnSD,sd(S[nDays,]))
```

```
## Standard Deviation 34.36564 34.11058
```

Presentation of the longnormal distribution of theoretical density and empirical density

```r
meanOfLog = log(S0) + (Airbusmean-(Airbussd^2)/2)*nDays
sdOfLog = Airbussd*sqrt(nDays)
priceGrid = seq(0,lnMean+6*lnSD,length=10000)
theoreticalDens = dlnorm(priceGrid,meanOfLog,sdOfLog)
empiricalDens = density(S[nDays,])

plot(priceGrid,theoreticalDens,type='l',xlab='Prices',ylab='Density')
lines(empiricalDens,col='blue')
```



Call Put Options price Black Scholes

```r
openingS = head(Quandl('EURONEXT/AIR', start_date='2017-06-04', end_date='2018-06-04'), n=1)
S = openingS$Open
strikecall = 110
optionCall = GBSOption(TypeFlag = "c", S, strikecall, 1/12, r = 0.03, b = 0.03, Airbussigma)
cat("OptionCall Price",optionCall@price)
```

```
## OptionCall Price 0.3445852
```

```r
strikeput = 90
optionPut = GBSOption(TypeFlag = "p", S, strikeput, 1/12, r = 0.03, b = 0.03, Airbussigma)
cat("OptionPut Price",optionPut@price)
```

```
## OptionPut Price 0.1961901
```