Documentation

Joris Baiutti & Claudio Polo

June 4, 2018

Load stuff

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

Api key: 1ywuwWFWbtaziA2GcaRU

```
Quandl.api_key('1ywuwWFWbtaziA2GcaRU')
```

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

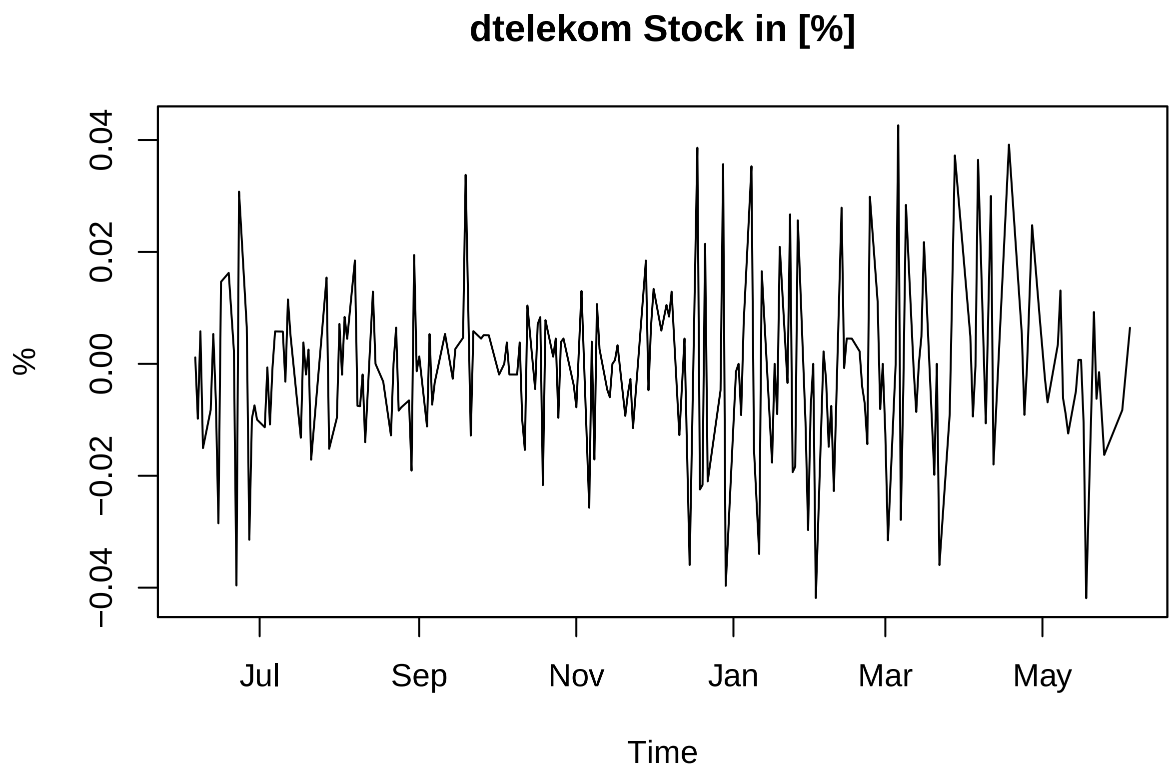
For the standard deviation we collect the percentuel rate values. These values can be optained directly with quandle. Then we can calculate standard deviation and mean of those rate values.

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',trans
plot(dtelekomclosing,ylab="%",xlab="Time",main="dtelekom Stock in [%]")
```



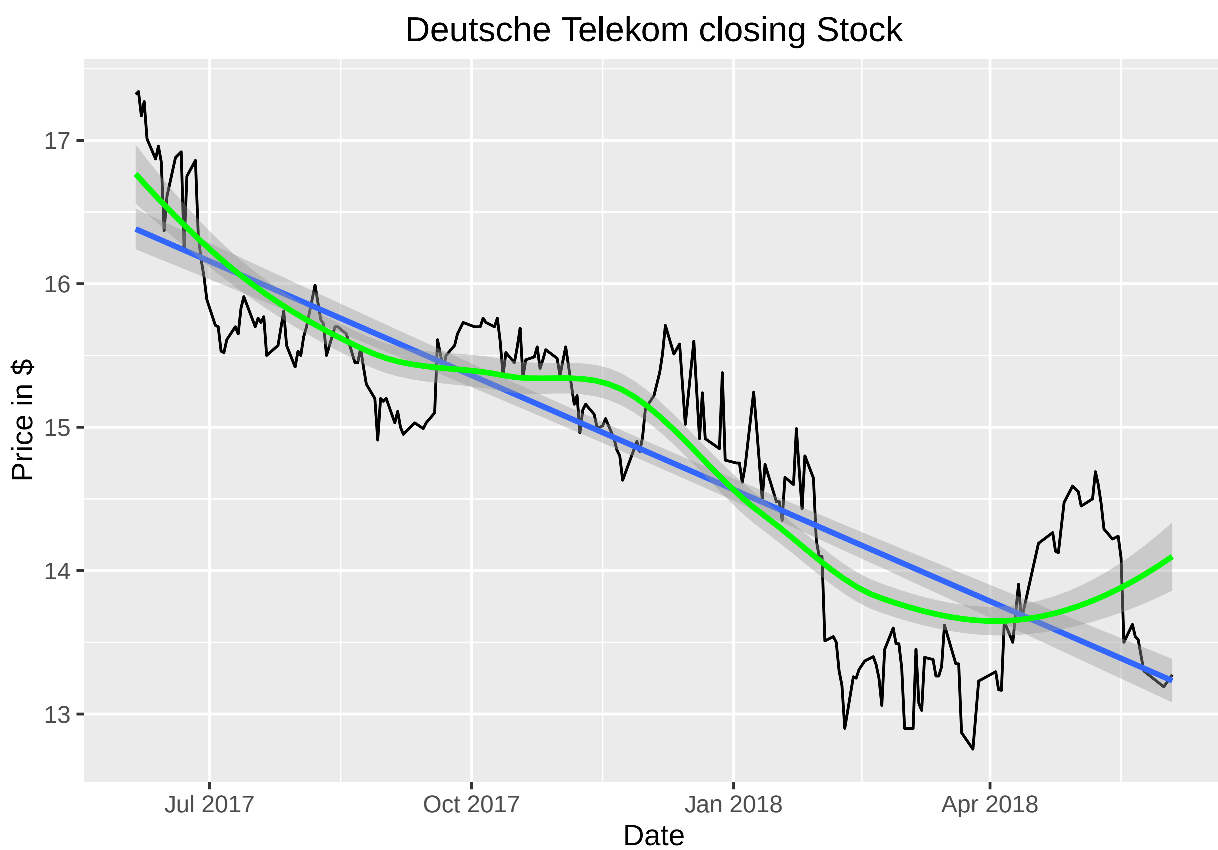

```
dtelekomsd = sd(dtelekomclosing)
dtelekommean = mean(dtelekomclosing)
```

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
```

```
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 = "l
```



Brownian Motion

```
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)
```

```
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
}
```

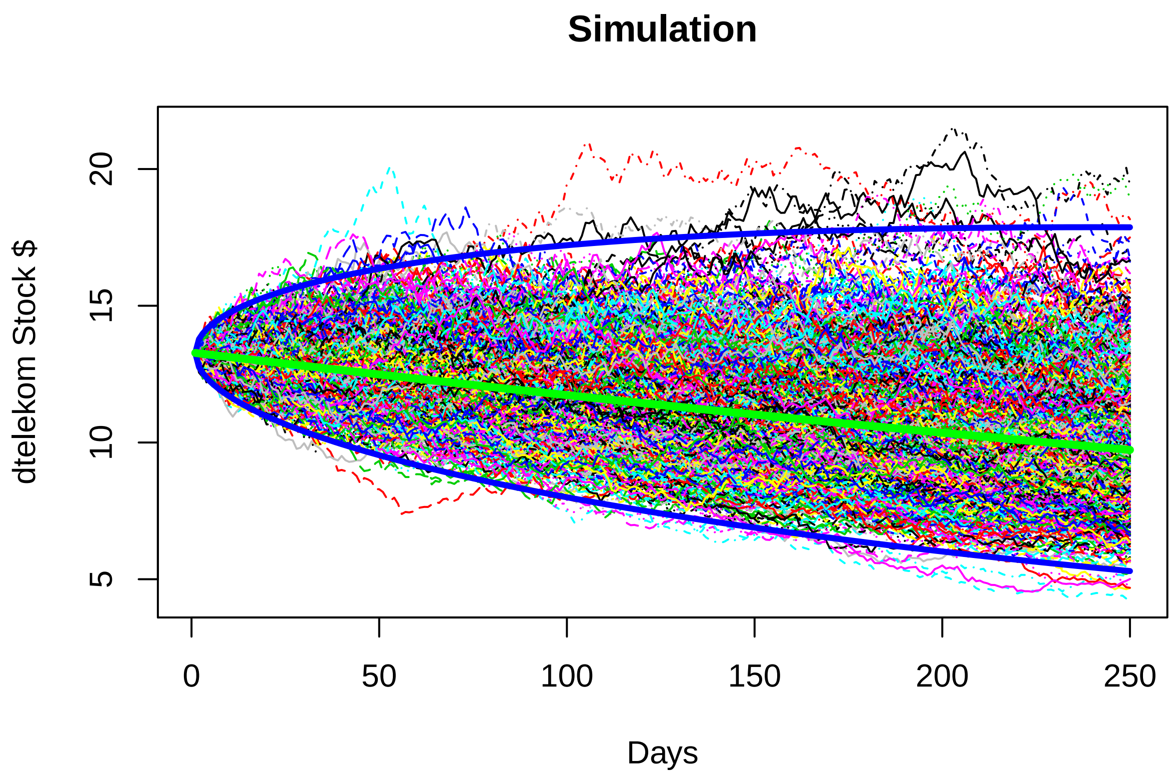
Calculation of SimulationS bOundaries

```
mean = dtelekommean * nDays
sigma = 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 =
for(i in 0:nDays){
  meanBnd[i] = S0*exp((( mean-sigma^2/2 ))*t[i])
  upBnd[i] = S0*exp((( mean-sigma^2/2 ))*t[i] + (sigma * 2.5 * sqrt(t[i])))
  loBnd[i] = S0*exp((( mean-sigma^2/2 ))*t[i] + (sigma * -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 10.04514
```

Standard deviation over a year over all Simulations

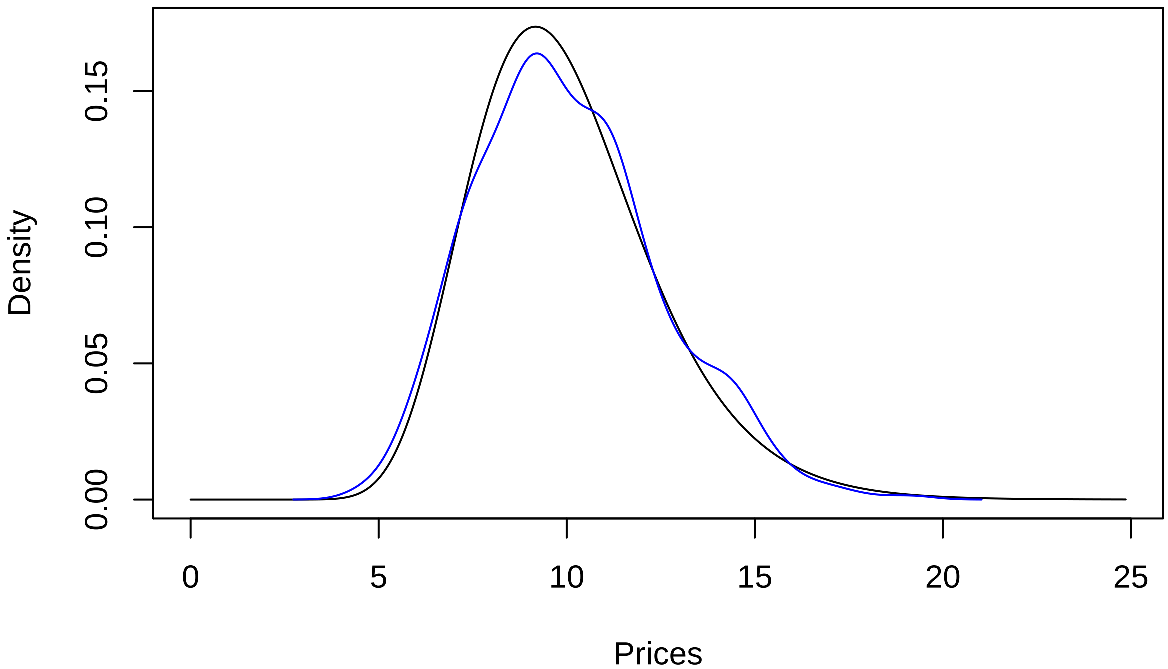
```
cat('Standard Deviation',lnSD,sd(S[nDays,]))
```

```
## Standard Deviation 2.473423 2.460196
```

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')
```

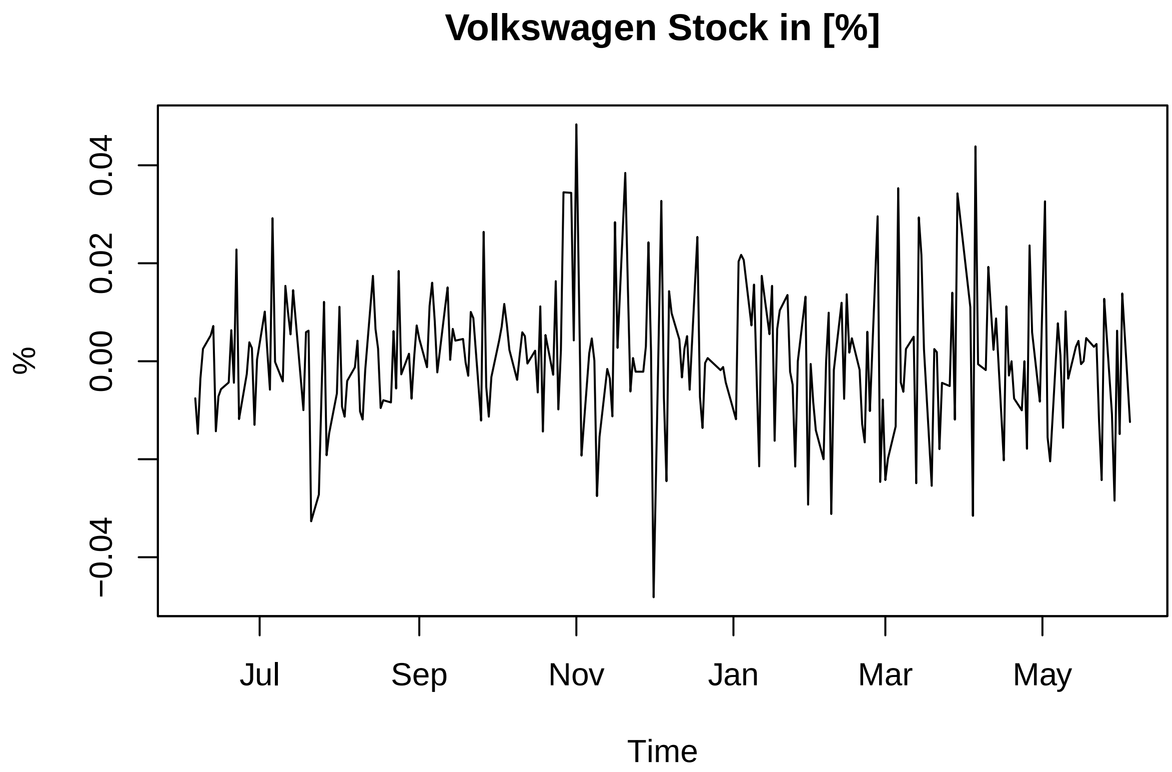


Volkswagen

Percentual rates

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

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



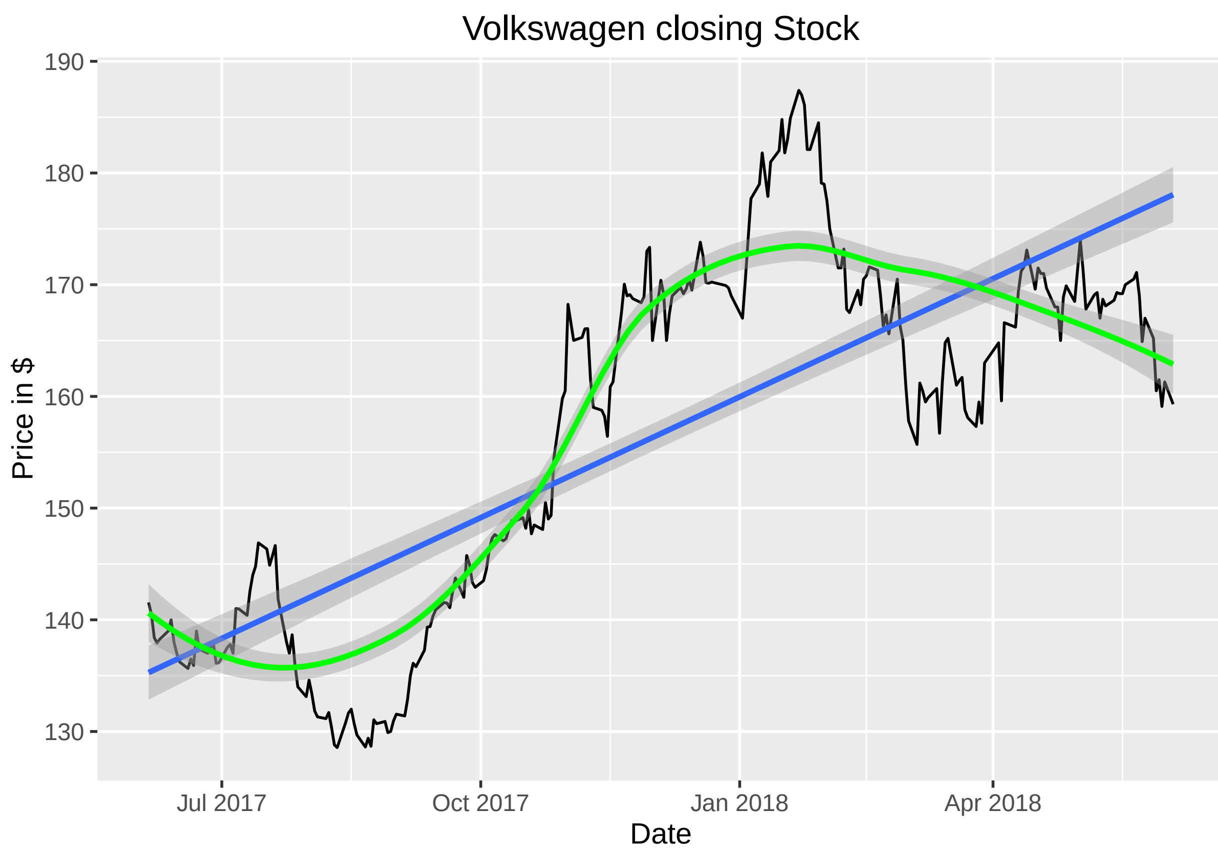

```
volkswagensd = sd(volkswagenclosing)
volkswagenmean = mean(volkswagenclosing)
```

Arithmetic mean and standard deviation

```
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 =
```



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
}
```

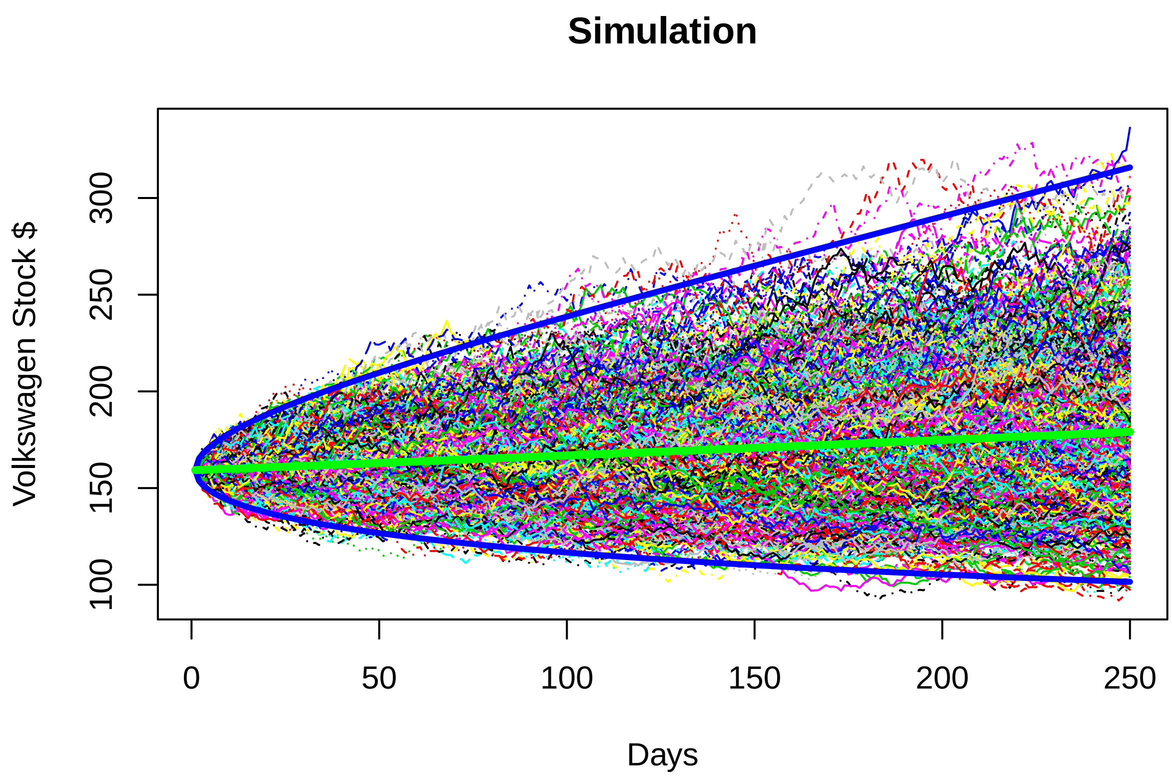
Calculation of SimulationS bOundaries

```
mean = volkswagenmean * nDays
sigma = 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 =
for(i in 0:nDays){
  meanBnd[i] = S0*exp((( mean-sigma^2/2 ))*t[i])
```

```
  upBnd[i] = S0*exp((( mean-sigma^2/2 ))*t[i] + (sigma * 2.5 * sqrt(t[i])))
  loBnd[i] = S0*exp((( mean-sigma^2/2 ))*t[i] + (sigma * -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

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

```
## Mean 182.6009
```

Standard deviation over a year over all Simulations

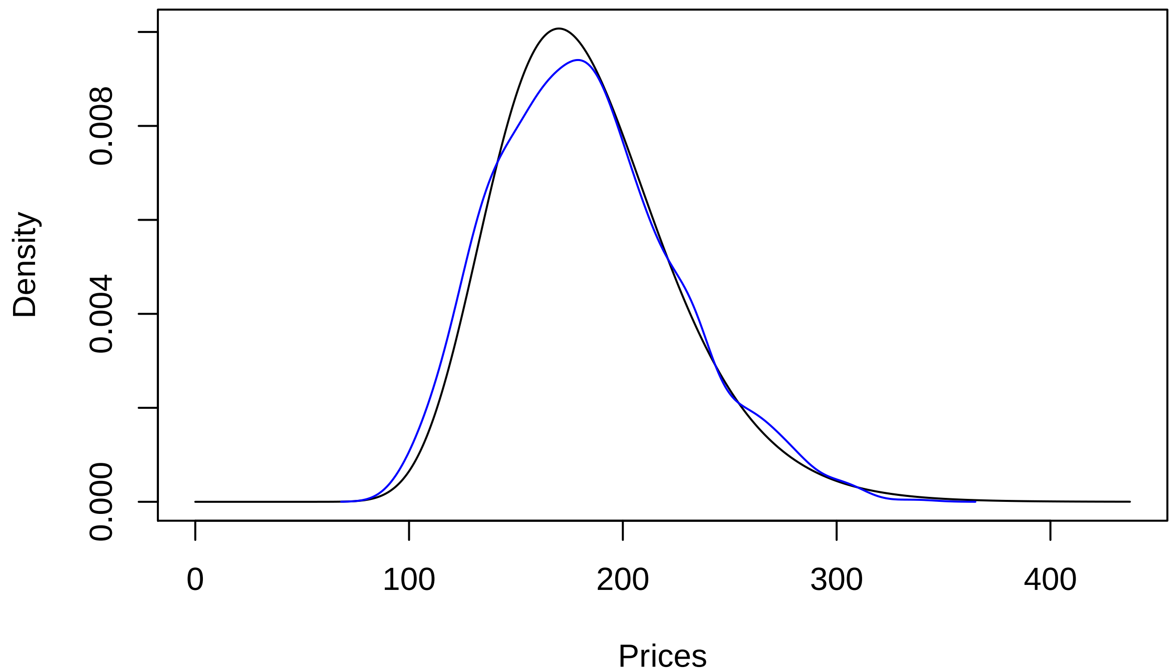
```
cat('Standard Deviation',lnSD,sd(S[nDays,]))
```

```
## Standard Deviation 42.24418 42.0243
```

Presentation of the longnormal distribution of theoretical density and empirical density

```
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')
```

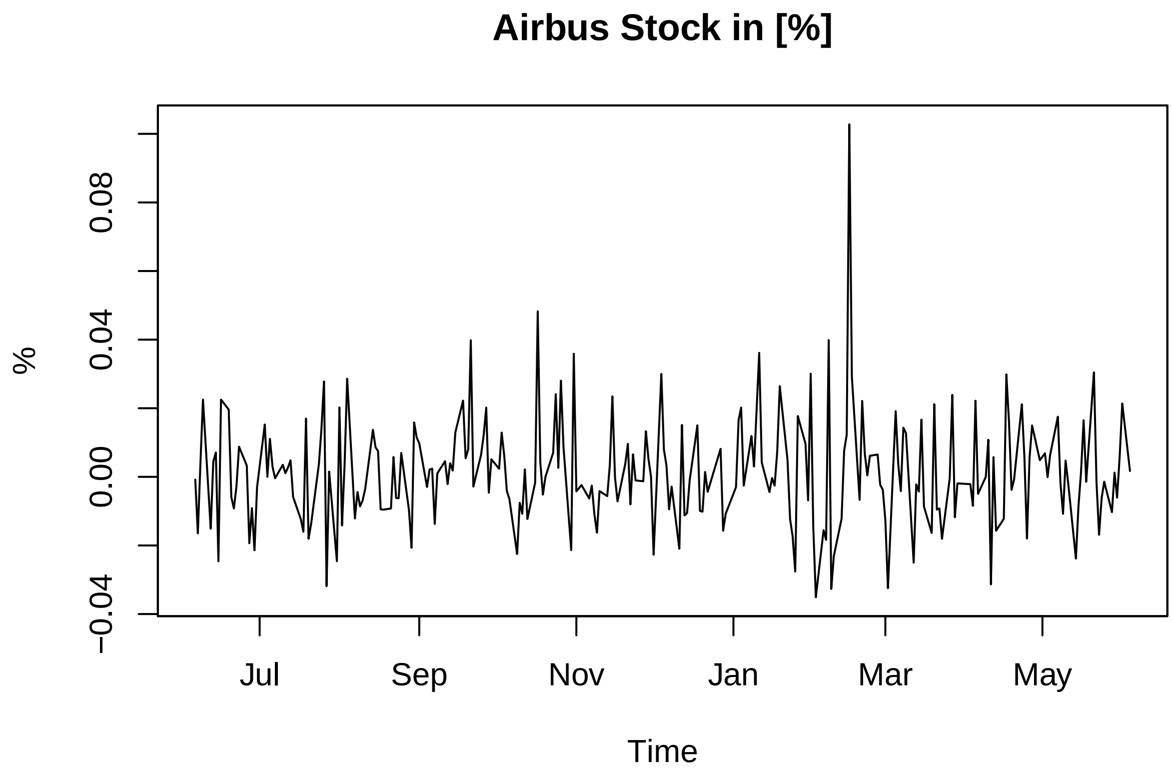


Airbus

Percentual rates

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

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



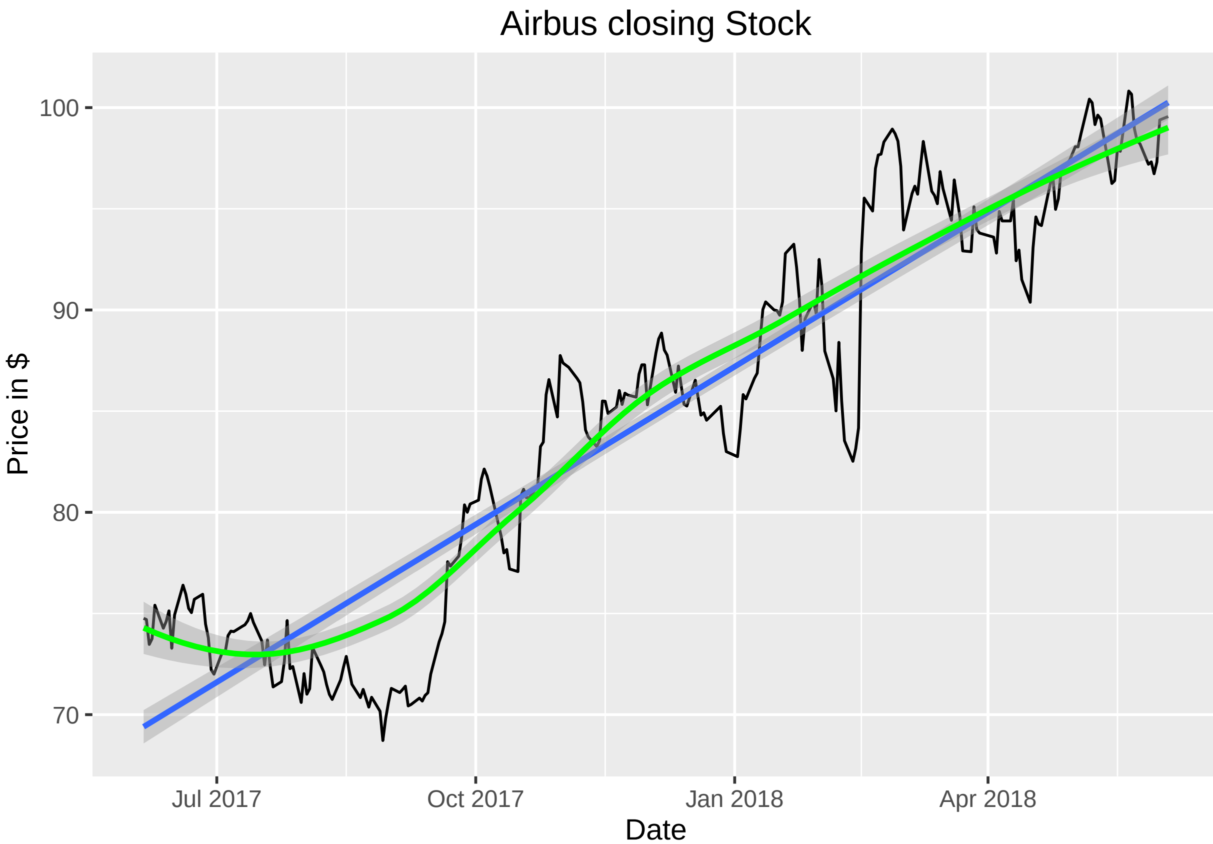

```
Airbussd = sd(Airbusclosing)
Airbusmean = mean(Airbusclosing)
```

Arithmetic mean and standard deviation

```
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
```

```
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 = "loes
```



Brownian Motion

```
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)
```

```
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
}
```

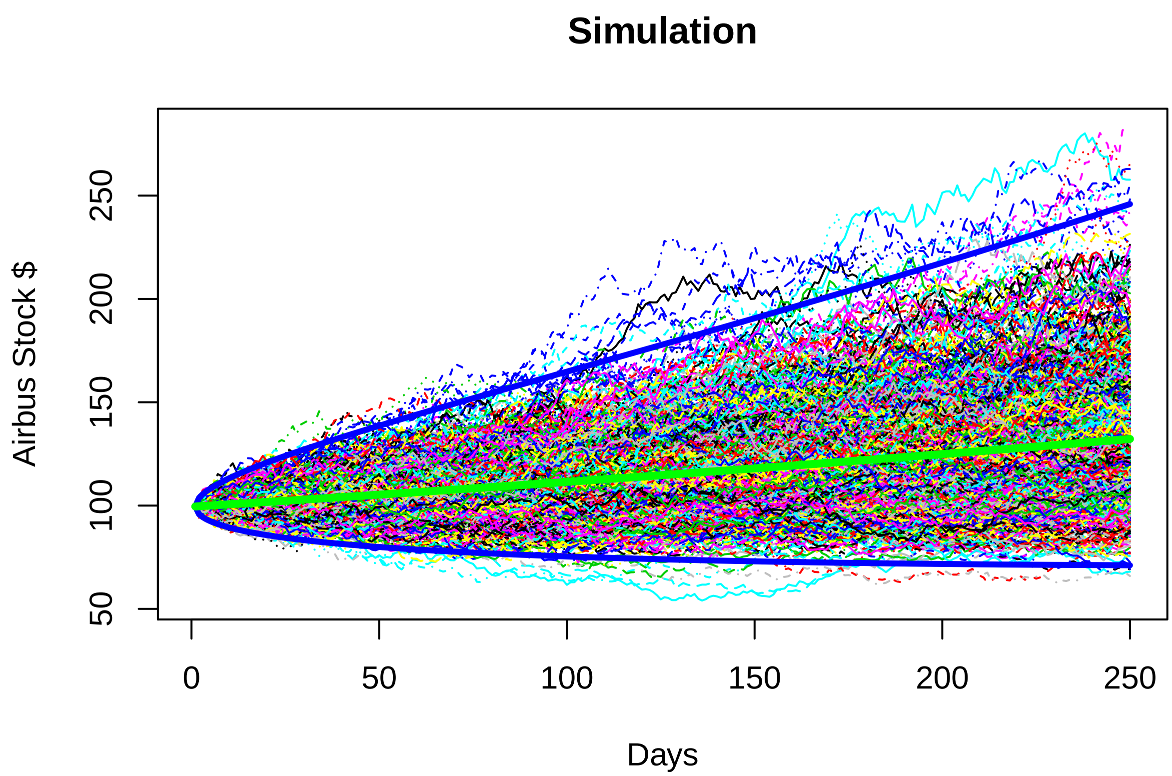
Calculation of SimulationS bOundaries

```
mean = Airbusmean * nDays
sigma = 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 =
for(i in 0:nDays){
  meanBnd[i] = S0*exp((( mean-sigma^2/2 ))*t[i])
```

```
  upBnd[i] = S0*exp((( mean-sigma^2/2 ))*t[i] + (sigma * 2.5 * sqrt(t[i])))
  loBnd[i] = S0*exp((( mean-sigma^2/2 ))*t[i] + (sigma * -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

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

```
## Mean 135.7192
```

Standard deviation over a year over all Simulations

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

```
## Standard Deviation 34.36564 33.77308
```

Presentation of the longnormal distribution of theoretical density and empirical density

```
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')
```