Quantitative Finance Experiment - Part 4 - Stock Volatility Simulation

Notebook:	QuantFin		
Created:	12/7/2018 8:07 PM	**Updated:**	1/6/2019 12:38 PM
Author:	Vladimir		

Problem

Models that based on the assumptions about the normal distribution of market volatility systematically underestimates the probability of intense volatility periods.

Goal

Acquire historical market data for at least ten years for major public trading companies. Based on the stock performance generate empirical distribution and fit different probability density functions. Then use the best-fitted function in the hierarchical model for Markov Chain Monte Carlo volatility simulations. Samples from the stimulation could be used to estimate the cumulative distribution function for price volatility.

Data Source

The data source I used for this research is quandl.com, an open online database for economic and finance data. Historical quotes were loaded directly using R API without any changes and preprocessing.

Abstract

After realizing that fitting a linear model to a pure fundamental data is not the best idea regarding price predictability, I decided to switch back to risk analysis. If we know the legitimate risk estimation we can use it to support design making about which stock is safe to buy, when is the best time to buy/sell and what is the fair price for the option for this stock.

First of all, we need to how we define risk? What is the difference between risk, uncertainty, and volatility?

Risk is basically the uncertainty that could be measured. Volatility is a difference between the minimum and maximum price in the defined period. If we know the initial stock price and monthly volatility, we can estimate the risk of losing money. In another world, the probability that stock's price at the end of the month will be lower than at the beginning.

$$Risk = E(price_a < price_b)$$

Where "price_a" is simulated current/ending price of the period and "price_b" is a starting price of the period. I'll describe a definition of "simulated" later in the text.

Then we can set up the "risk thresholds" that could be identified as: "If an asset has more than n% risk of negative return, we should consider it as non-inestimable" or "If an asset has more than n% risk of negative return we won't open a short option position on it"

"Simulated" price means a theoretical price drown form the approximate distribution of asset volatility. If we know the distribution(or at least the most appropriate one) we can use Monte Carlo simulations to "look to the future" and estimate where price could be after n-periods of time.

Distribution approximation

Let's consider an example of Apple stock price. One of the most popular assets in the stock market. It gave an enormous return for their investors and a distinct "outlier" in terms of performance. Here's a visualization of its stock price change over the last few years:

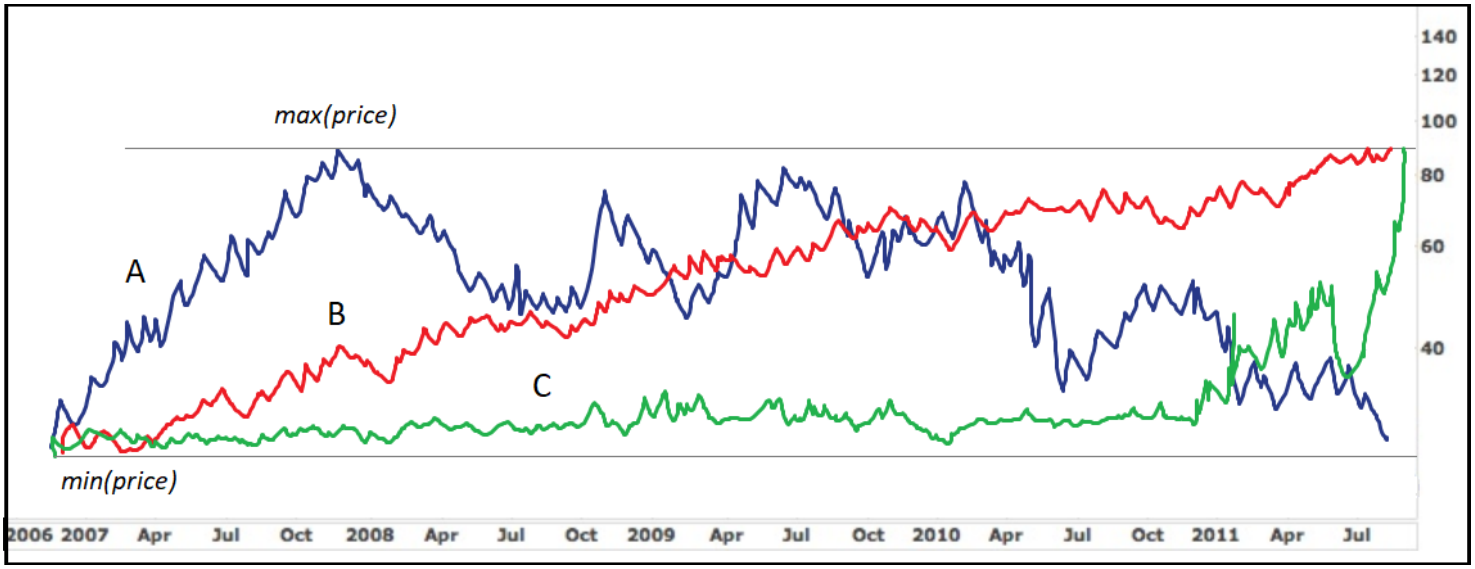
source: Yahoo Finance

With the average annual stock market return in 10%, Apple shows strong 33%, at least three times more.

To calculate monthly volatility, I got historical market data from yahoo finance, aggregated it by month and estimated relative volatility with:

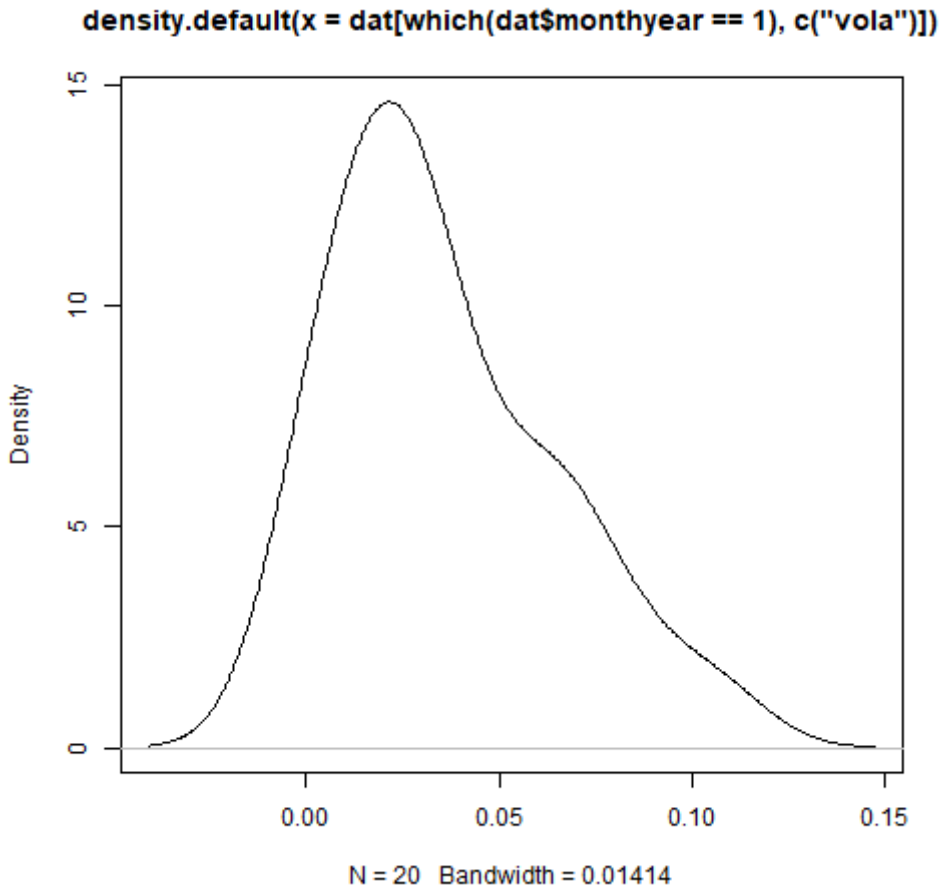
$$Vol = \frac{max(price) - min(price)}{min(price)}$$

It's important to mention that this formula is allowed only for intraday behavior. Applied to monthly/yearly aggregated data it won't represent the risk of investment because it doesn't consider hidden action of the asset. As an example, let's see the behavior of three different hypothetical stocks:



As we see, all - A, B and C stock have the same price range in a selected interval, but all have very different behavior and thus different implied risk.
We need to consider one volatility measure that takes into account stock's inner behavior.

One of the ways to describe volatility is to use a standard deviation of mean daily volatility. Let's look at the chart of a mean volatility distribution of the specific month.

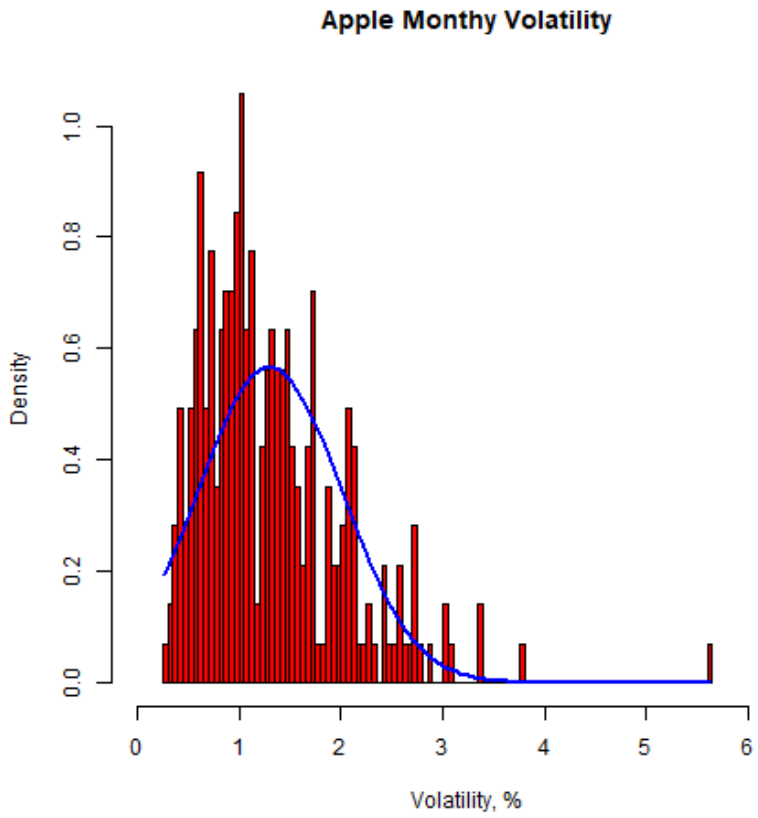


The mean of this plot is the average daily volatility of the specific month. Depending on stock's behavior, the density plot will be different for each volatility profile. That means if we use density bandwidth as a volatility measurement, we can catch the difference of intraperiod performance.

To estimate stock's monthly volatility we need to find it's mean daily volatility, calculate the variance for each day observation and then sum it over.
Assuming that each volatility observation is independent (we'll come back to this later), the monthly volatility can be expressed from daily volatility using the following expression:

$$\widehat{\mathrm{Var}}(x_{t+1} + \ldots + x_{t+K}) = \widehat{\mathrm{Var}}(x_{t+1}) + \ldots + \widehat{\mathrm{Var}}(x_{t+K}).$$

Here's a density plot of the monthly stock volatility and fitted normal distribution function:

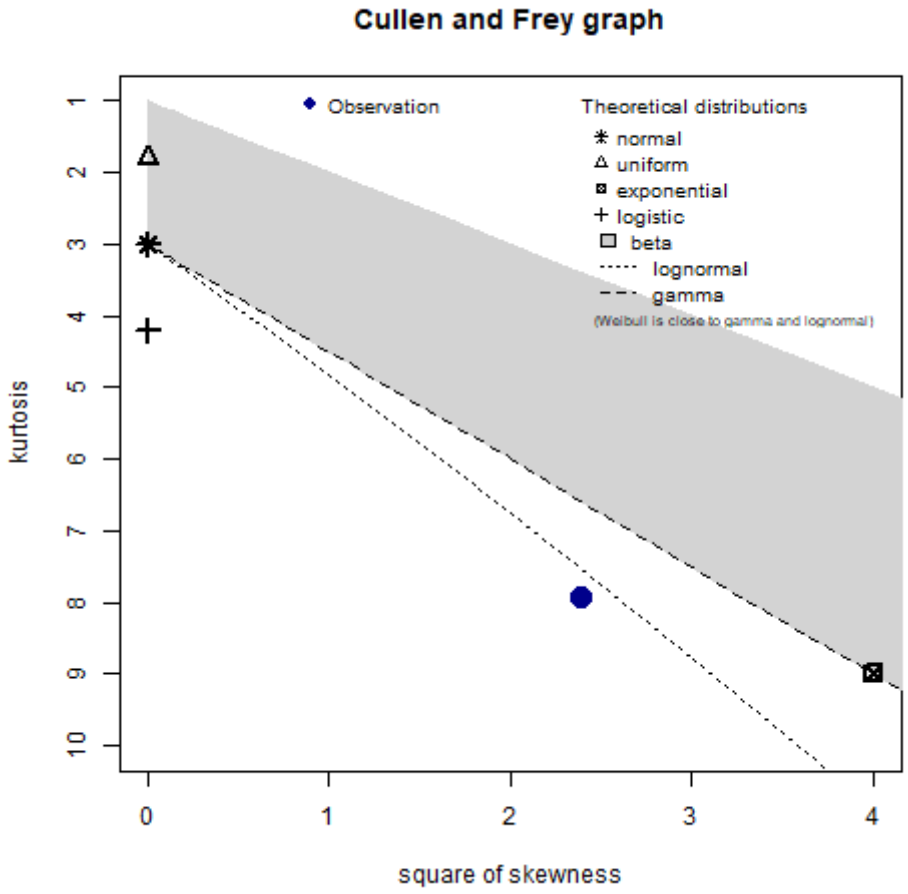


As we see, the most decency mass is concentrated in 0%-4% range. We also recognize that there was couple month with extremely high volatility when stock returned around 6% ROI.

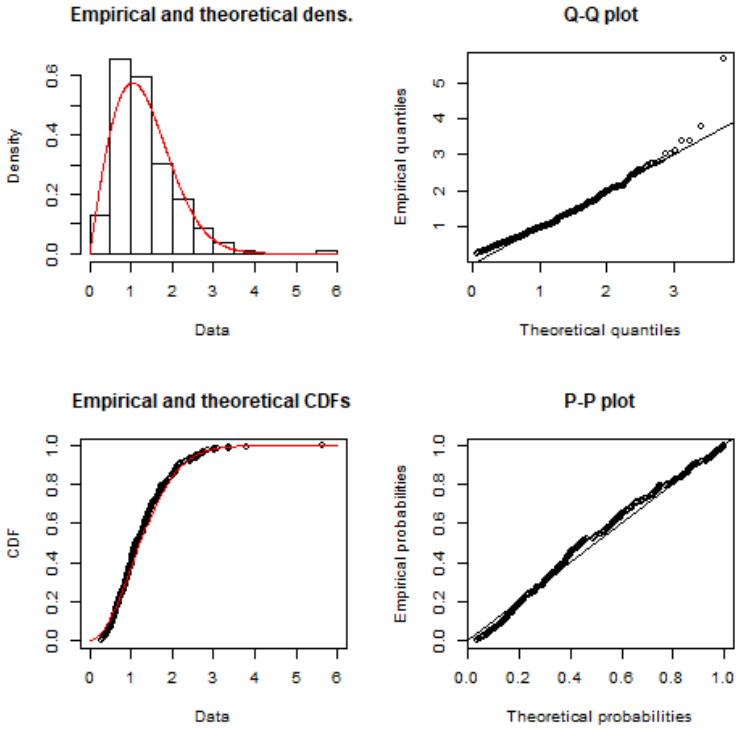
It's also clear that the Normal distribution cannot be used for volatility approximation. The normal distribution underestimates the probability of medium and high volatility. We need to consider using more "fat tail" like model.

Now, let's try to fit different probability density functions and see which one will work the best. To find it out we need to find parsimonious approximate descriptions.

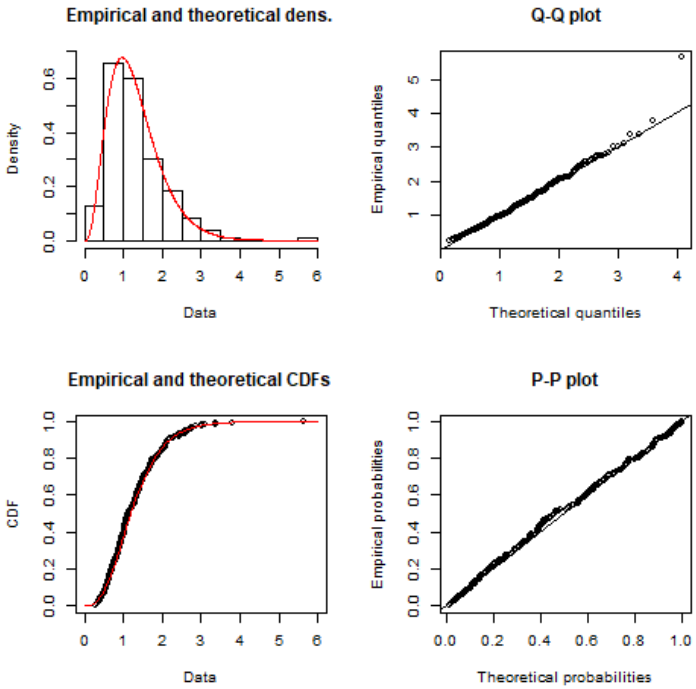
Here's a plot of the kurtosis and squared skewness of our samples:



It gives some basic estimations about potential candidates for theoretical distribution.
We see that the closest estimation will be a lognormal distribution.
Here are q-q and p-p plot of a lognormal distribution:



And for a Gamma distribution:



The AIC statistics for weibull, normal, gamma and log-normal are the following:

```
>aic_df
   weibull    gamma     norm    lnorm
1 550.6034 525.3154 609.4202 520.4791
```

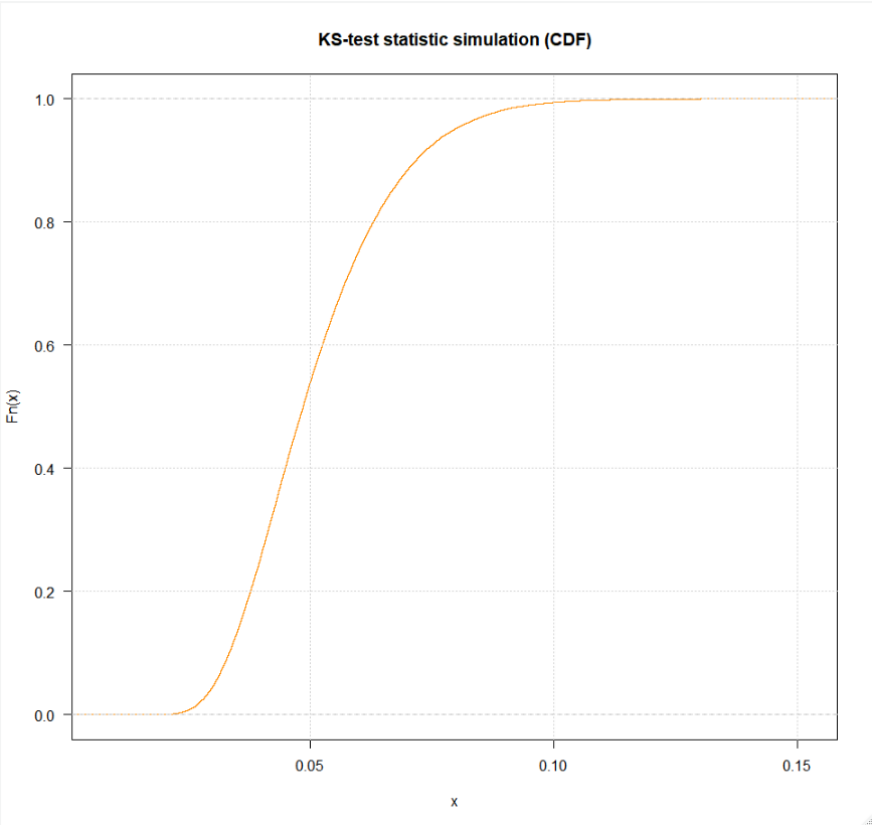
As we see, both gamma and log-normal distribution have fit the data well.

I'll test both gamma and lnorm models separately and see which gives a better return on the backtesting stage. Let's start with a gamma distribution. Because it has higher kurtosis, it better represents the underlying asset behavior.

Solving Maximum Likelihood estimation for the Gamma distribution, we can get the best fitted (Shape, Rate) parameters:

```
> fit_stat[[1]]$gamma$estimate
   shape     rate
3.848280 2.952326
```

Then, using these parameters, we can create a simple hierarchical Bayesian model to sampling monthly stock volatility. To create MCMC model, we'll use JAGS, and it's R API. String representation of the model looks following:
Using these parameters we drown random vector form Gamma (shape, rate) using founded parameters. After that, we can calculate the Kolmogorov-Smirnov test to identify what is the probability that simulated and observed data points are from the same Gamma distribution.



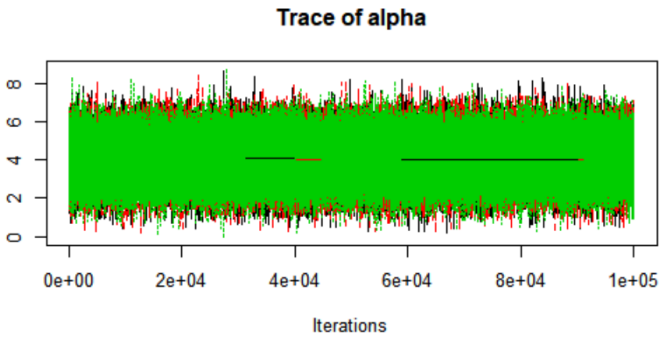
The Kolmogorov–Smirnov statistic is 0.041. Hence, for the significance level 0.01 level, the p-value is 0.072, we can strongly assume that the observed samples are drowned from a Gamma distribution.

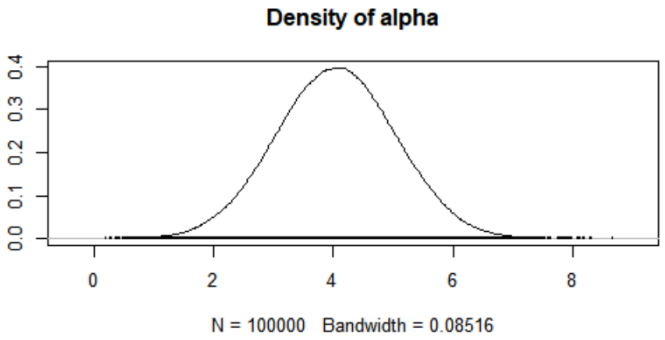
We can now use the obtained distribution as a prior distribution for a Bayesian model for volatility simulation.
For simulations, we'll use JAGS, and it's R API. Here's a model's string for rjags:

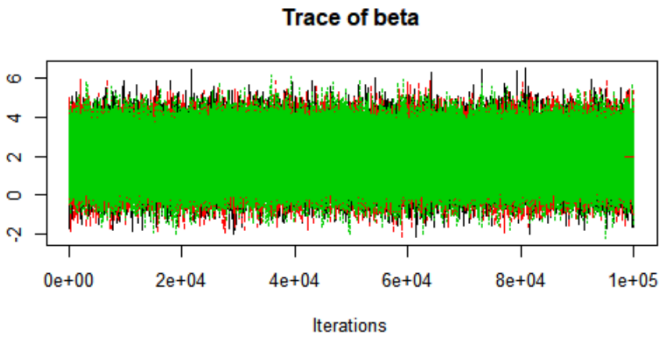
```
"model {
  #Likelihood
  for(i in 1:n) {
    y[i] ~ dgamma(alpha,beta)
  }
  #Prior
  alpha ~ dnorm(a_mu,1.0/sig_sq)
  beta ~ dnorm(b_mu, 1.0/sig_sq)
  "
```

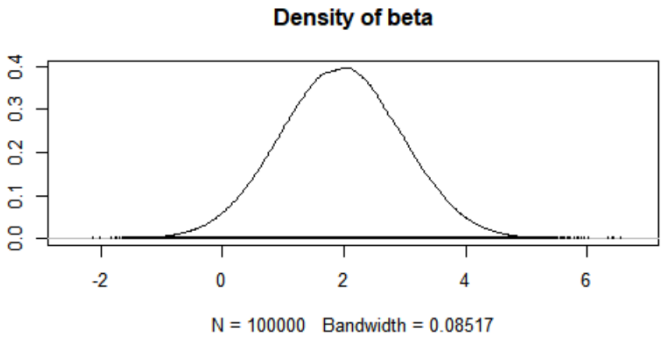
Datapoints are draws from Gamma distribution with alpha and beta parameters. Alpha and beta for their part are drowned from a normal distribution with mu and 1/sig^2 square parameters. Mu parameter is set up as an MLE from the KS test, while sig^2 is remaining free.

The simulations were run in three chains with 1e5 iterations each. The burn-in period was set up to 1e3 iterations.
Here's a trace plot for the simulations:
The trace plot for the model looks the following:

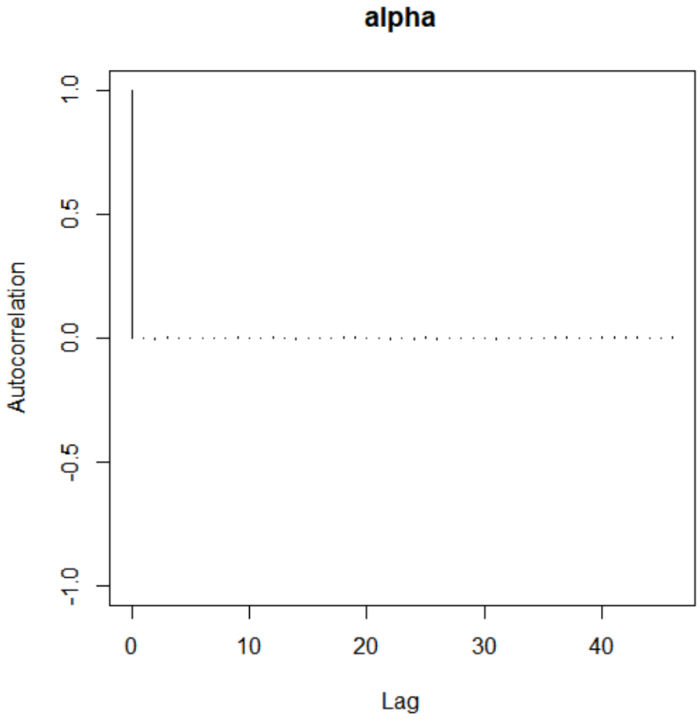


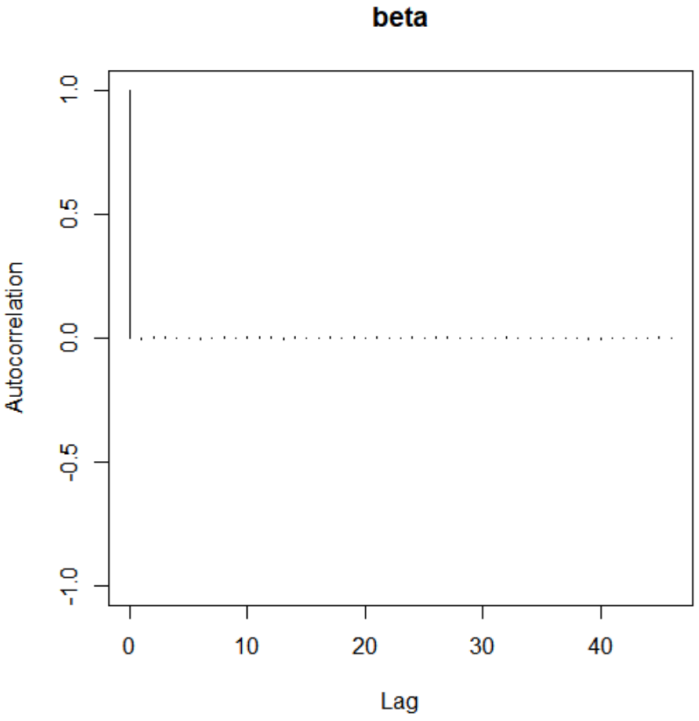






Autocorrelation levels and Gelman-Rubin diagnostic looks good:






```
>autocorr.diag(mod_sim)
               alpha         beta
Lag 0   1.0000000000 1.0000000000
Lag 1   0.0020603758 0.0017301907
Lag 5   0.0003452382 0.0004947912
Lag 10  0.0007025614 0.0007559833
Lag 50 -0.0005045586 0.0023457908
```

```
>gelman.diag(mod_sim)
Potential scale reduction factors:

      Point est. Upper C.I.
alpha          1          1
beta           1          1

Multivariate psrf

1
```

We can reasonably assume that the chain converted and exploring stationary distribution.

Now, let's tweak the model to find the best fit for the real-life patterns. Using the free sigma parameter, we can control the maximum "allowed" size of outliers in volatility and probability of high volatility periods.
Here's a result of ten simulations with gradually decreasing sigma from 80 to 0.07:

```
>results
       Sigma Mean.Volatility Max.Volatility High.Volatility.Probability
1  3.200000e+02      0.01474138     0.09212599                 0.004666667
2  1.600000e+02      0.01548248     0.10307916                 0.011666667
3  8.000000e+01      0.01668404     0.11868926                 0.021166667
4  4.000000e+01      0.01837978     0.14249831                 0.046666667
5  2.000000e+01      0.02086177     0.21776297                 0.086000000
6  1.000000e+01      0.02376335     0.26530006                 0.123666667
7  5.000000e+00      0.02763875     0.37926462                 0.163166667
8  2.500000e+00      0.03294886     0.49605275                 0.212500000
9  1.250000e+00      0.03974472     0.68206678                 0.252000000
10 6.250000e-01      0.04802832     0.68224880                 0.293166667
11 3.125000e-01      0.05795910     0.95297522                 0.324333333
12 1.562500e-01      0.06924015     1.31670261                 0.352666667
13 7.812500e-02      0.08127402     1.42828086                 0.378333333
14 3.906250e-02      0.09413590     1.77476913                 0.402000000
15 1.953125e-02      0.11138133     1.85243489                 0.447333333
16 9.765625e-03      0.14337547     2.53578841                 0.537000000
17 4.882812e-03      0.23243140     2.85253951                 0.734166667
18 2.441406e-03      0.46777434     3.26585908                 0.954666667
19 1.220703e-03      0.77102234     3.76200429                 0.996833333
20 6.103516e-04      0.99263209     4.05710751                 0.999500000
21 3.051758e-04      1.13428103     4.42465701                 0.999833333
```

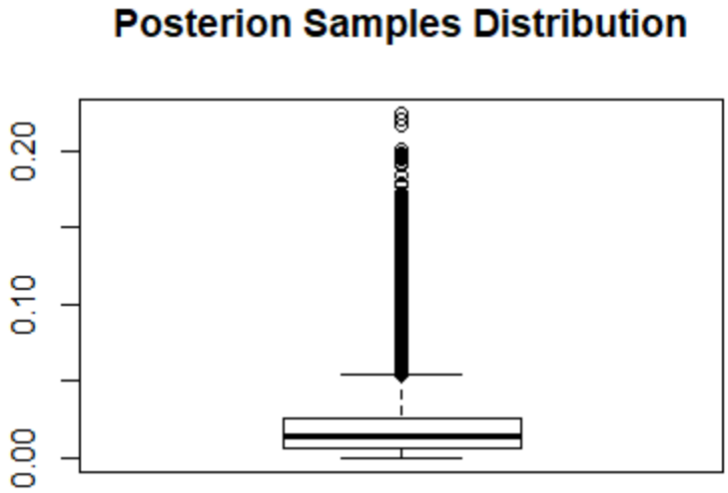
And the summary from observed data:

```
> mean(empirical_distribution> 3*mean(empirical_distribution))
[1] 0.003521127
> summary(empirical_distribution)
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
0.002717 0.008045 0.011309 0.013035 0.016699 0.056315
>
```

High.Volatility.Probability column is defined as a probability that monthly volatility will be more than three times higher than average.

We see that the best fit to empirical data will be a model with sigma parameter equal to 40.

The boxplot for the result distribution is the following:




```
>mean(y_hat)
[1] 0.06417608
```

We see that most of the distribution mass lay around 6% with high volatility outliers. Indeed, AAPL stock price had 20% volatility in late 2018 after more than ten years stable growth.