Risk Management and Regulation

Term Project

qq2112 Quan Qian

ts2956 Tianyi Shao

sz2533 Sumy Zhang

Fall 2016

Contents

1. Project description

1.1 Executive Summary

This is a review of the risk calculation system developed by Quan Qian, Tianyi Shao, and Sumi Zhang. The system takes an arbitrary portfolio of stocks and European vanilla options as input and calculates the portfolio's historical, Monte Carlo and parametric VaRs. The system also produces backtesting results for the VaR numbers calculated against historical returns. For options, prices are calculated using the Black-Scholes model without taking into account the volatility skew.

The VaR numbers calculated by the system can be used to estimate the risk level of the input portfolio, and hence facilitate the user's risk management process.

We find the system has the following strengths:

- The system is computationally inexpensive due to choice of Black-Scholes model.
- Equity spot risk is well accounted for.
- The system is capable of computing three types of different VaR, making comparison between different VaR approaches convenient.

We also identify the following weaknesses:

- The model is subject to many assumptions, including the traditional Black-Scholes assumptions for option pricing, i.e. constant interest rates, equity follows geometric Brownian motions, constant equity volatility, frictionless trading, etc.
- The model does not account for interest rate, dividend and equity volatility risks as they are assumed constant, while in practice they show stochastic nature.

Overall, we consider the weaknesses to be reasonable as far as this project is concerned.

1.2 Introduction

This is a review of the risk calculation system. We cover the computation of historical VaR, parametric VaR and Monte Carlo VaR for a portfolio consisting of some arbitrary stocks and European options. We will document the methodologies of calculation for all three approaches and analyze the suitability of the model for our intended risk calculations.

1.3 Product Description

Value at Risk (VaR) is a common metric for assessing the risk of a certain portfolio. Despite the criticisms VaR has received, it still remains as one of the most crucial metrics used by banks and investment firms. Its usage is always accompanied by the concept of significance level and horizon. Mathematically, Let V be the X-day Y% VaR, we have $\mathrm{P}(L < V) = 1 - Y\%$, where L is the X-day loss on the portfolio. In this document, the portfolio is restricted to only contain stocks and European options.

1.4 Model Description

Model Dynamics for Parametric and Monte Carlo VaR

The model assumes the value of the entire portfolio to follow a geometric Brownian motion process. Let the portfolio value at time t be P_t, then

$$dP_t = \mu P_t dt + \sigma P_t dW_t$$

Where μ and σ are calibrated to the historical portfolio values with an exponentially weighted approach.

Parametric VaR

Since geometric Brownian motions have analytical solution, it is possible to directly calculate VaR in a parametric approach. The analytical solution to above geometric Brownian is

$$P_t = P_0 \exp\left(\left(\mu - \frac{\sigma^2}{2}\right)t + \sigma W_t\right)$$

After some algebra, we have the following formula for the parametric X-day Y% VaR:

$$\mathrm{VaR} = P_0\left(1 - exp\left(a_Y \sigma \sqrt{X/252} + \frac{X}{252}\left(\mu - \frac{\sigma^2}{2}\right)\right)\right)$$

Where α_Y is the number such that $\mathrm{P}(Z < \alpha_Y) = 1 - Y\%$. Z is a standard normal random variable.

Monte Carlo VaR

Instead of analytically computing the VaR, the model generates N random samples of portfolio values after X days. This is easily achievable given the analytical solution of geometric Brownian motion and the availability of Gaussian random number generators. For each portfolio value simulated, the corresponding loss is computed. The X-day Y% VaR is then simply the $1 -$ Y% quantile of the N losses generated.

Historical VaR

The historical VaR approach is straight forward. Let $L_1, L_2, \cdots, L_N$ be the X-day losses of the portfolio (N is determined by the length of historical data window). The X-day Y% VaR is simply the $1 -$ Y% quantile of the N X-day losses. This approach does not make any assumptions for the underlying dynamics of the portfolio.

1.5 Backtesting

The event where realized portfolio loss exceeds the VaR calculated is called a VaR breach. If the VaR calculation is accurate and effectively captures the risk of the portfolio, we would expect the number of VaR breaches to be approximately 1-Y% of the observations, for a Y% VaR. Therefore, if we compare for each day the VaR number and the realized X-day loss, we could observe how effective the VaR estimates are throughout history.

1.6 Model Calibration

The only calibration used by the model is the calibration of μ and σ for the GBM used in parametric and Monte Carlo VaR. We first choose a window of historical data to be used, say 2 years. Then we apply an exponential weight to the historical portfolio values, making near current data points more influential than old data points. The μ and σ are obtained from the weighted average and standard deviations.

1.7 Numerical Model Implementation

The parametric and historical VaR approaches do not involve numerical methodologies such as PDE grid solver of Monte Carlo simulation. The Monte Carlo VaR is implemented by a Gaussian number generator for the standard Brownian motions. The simulated Brownian motion terminal values are then substituted into the GBM analytical solution to obtain the simulated portfolio values and losses.

1.8 Model Assumptions and Analysis

We identify the following assumptions for all three VaR approaches:

- Option prices is accurately priced with Black-Scholes model without taking into account the volatility skew. We do not consider volatility skews here due to the limited data availability. Calibration of implied volatility surfaces requires large amount of market option data input and intensive computation power.

We identify the following assumptions for parametric VaR approach:

- The portfolio value as a whole follows geometric Brownian motion. This assumption enables fast and simple analytical solutions for parametric VaR. Other approaches, such as assuming GBM for each individual stock, would result much more complex calculation (higher dimensions, calibration of correlations, etc.), especially given that we also have options in the portfolio.

We identify the following assumptions for Monte Carlo VaR approach:

- The portfolio value as a whole follows geometric Brownian motion. This assumption enables fast and simple analytical solutions for Monte Carlo VaR. Other approaches, such as assuming GBM for each individual stock, would introduce huge numerical complexity (discretization schemes, calibration of correlations, etc.).
- Enough convergence is achieved with the chosen MC sample size. We currently use 10000 paths and consider the number to be reasonable. Essentially we need to achieve a balance between computational cost and accuracy.

We identify the following assumptions for historical VaR approach:

- All historical data points are equally relevant for estimating VaR. The simplest approach gives equal weight to all the historical data points in the used window, while in reality further away data points are usually less relevant. This is one of the reasons that we use exponential weight for parametric and Monte Carlo VaR approaches.

1.9 Conclusions

We conclude that our approaches are adequate for VaR calculations as far as this project is concerned. Its biggest weaknesses are the number of assumptions the model is based on and the fact that only equity spot risk is captured. However, given the scope of this project, we consider the weaknesses to be acceptable. The model is capable of giving fast estimates in three different approaches and also the backtesting results, making it a useful tool for quick risk calculation and analysis.

2. Software design documentation

2.1 Detailed Description of Software Architecture

(1) Use the Black-Scholes equation to compute option price corresponding to each day's stock price
(2) Put prices of stocks and options together as a combined portfolio
(3) Do GBM to the portfolio and compute the parametric VaR
(4) Compute historical VaR of the portfolio for each date based on the drift and vol estimates for each date in history using relative change
(5) Compute Monte Carlo VaR based on fitting GBM parameters for the portfolio itself for each date
(6) Backtest: compare actual P&L to VaR and sum up the exceptions
(7) Plot all the outcomes

2.2 Module Documentation

a. Purpose

The purpose of the module is to develop a risk calculation system, where we can take an arbitrary set of stocks and option position as input and calculate the portfolio's Monte Carlo, historical and parametric VaR. The system is able to both calibrate to historical data and take parameters as input with user's choice of relevant variables, including VaR percent, year window and days of VaR and backtest the computed VaRs against history to validate its correctness.

b. Assumptions

- We assume that the portfolio follows GBM
- Fixed variables in Black Scholes formula like rate (1%) and volatility (20%).

c. Interfaces

Platform: Matlab R2015a

Input: Close price of all stocks involved in the portfolio (the price input of the model); Strike Price of each option (the options input of the model); number of shares of stocks and options(the port input in the model). For example, one fund manager purchased 500 shares of apple, 300 shares of IBM, 50 shares of $120 apple call option of apple, 20 shares of $90 put option, 30 share of $75 IBM call option, 99 shares of $50 IBM put option. Then price should be the historical prices of apple and IBM, port=[500 300 50 30 20 99], options = [120 90; 75 50].

The input percent is the percent of VaR, so are window and day.

d. Data structures

A data structure is a particular way of collecting and organizing data and key to designing efficient software. It ensures the efficiency to perform operations on large amounts of data.

3. Test plan

3.1 Documentation of Purpose, Design and Execution of Test Plans

We Use VaR Backtesting methodology to backtest the computed Monte Carlo,

historical and parametric VaR. We use 5 day holding period calibration to compute 99% VaR, and then compare the realized P&L to VaR to confirm the accuracy of VaR. The test exceptions are the days that the actual portfolio P&L exceeds 99% VaR. We expect exceptions in $(1 - p)\%$ of the cases for p percentile VaR, and compute the total number of exceptions daily over last 252 days.

3.2 Summary of Results with Commentary and Evaluation

We use Zone system to evaluate the model:

Green zone: ≤ 4 exceptions, acceptable

Yellow zone: 4-9 exceptions, investigate

Red zone: ≥ 10 exceptions, requires immediate work to improve model

And the model should also meet the following requirements:

- No more than 12 99% exceptions/year
- No more than 30 97.5% exceptions/year

3.3 Detailed Analysis of Informative Samples

We choose the following 10 stocks to test the model: Apple Inc. (*AAPL*), Advanced Micro Devices, Inc. (*AMD*), Bank of America Corporation (*BAC*), General Electric Company (*GE*), Intel Corporation (*INTC*), Microsoft Corporation (*MSFT*), Pfizer Inc. (*PFE*), Rite Aid Corporation (*RAD*), Wells Fargo & Company (*WFC*) and Yahoo! Inc. (*YHOO*). Each of them consists 20 years of historical stock price. Then we use Black-Scholes formula to compute the prices of put option and call option for corresponding stocks maturing in 1 year with certain strike, and combine the stocks and options together as a portfolio. We compute the VaRs of the portfolio and calculate the total number of exceptions, then compare the test results to see if the model meet the requirements above.

4. Software

4.1 Code

```
function[parametric,parametric2,historical,monte] = VaR(price,port,percent,day,window,options)
%compute all types of VaR given historical data; exponantional weighting
%Assume GBM
%price is the historical price of stocks, which is x*y
%port is the shares of different equities in portfolio, include normal stock, call options and put options,
which is 1*y
%percent is the percent of VaR, which is a number between 0 and 1
%day is the days to calculate VaR, window is the year window
%options data only includes the strike price of each option corresponding to each stock in price data


%Use Black-Scholes to simulate option price. Assume rate, maturity time and volatility to be constant
[x,y] = size(price);
Call = zeros(x,y);
Put = zeros(x,y);
for i = 1:x
  for j = 1:y
    Strike = options(j);
    Rate = 0.01;
    Time = 1;
    Volatility = 0.2;
    [Call(i,j), Put(i,j)] = blsprice(price(i,j), Strike, Rate, Time, Volatility);
  end
end

%Combine stock and option price together
price = [price,Call,Put];
[m,n] = size(price);
%calculate portfolio value
value = zeros(m,1);
for i = 1:m
  for j = 1:n
    value(i) = value(i) + price(i,j)*port(j);
  end
end

%calculate log return
lreturn = zeros(m-1,1);
for i = 1:m-1
  lreturn(i)= log(value(i)/value(i+1));
end
%square log return
slreturn = lreturn.^2;
```

```
%calculate lambda with a weight of 20%
lambda = nthroot(0.2,252*window);
%construct lambda array
Lambda = zeros(m,1);
for i = 1:m
  Lambda(i) = lambda^i;
end
%exponantial weighting lambda
weighted = zeros(m,1);
for i = 1:m
   weighted(i) = Lambda(i)/sum(Lambda(1:(window*252)));
end

%compute sigma and mu
a = m-window*252;
sigma = zeros(a,1);
mu = zeros(a,1);

for i = 1:a
   b = sum(slreturn(i:(i+252*window-1)).*weighted(1:window*252));
   c = sum(lreturn(i:(i+252*window-1)).*weighted(1:window*252));
   sigma(i) = sqrt(b - c^2)*sqrt(252*window);
end

for i = 1:a
   d = sum(lreturn(i:(i+252*window-1)).*weighted(1:window*252));
   mu(i) = d*252 + sigma(i)^2/2;
end

%compute the initial capital (portfolio value of the first day in trading history)
v0 = sum(price(m,:).*port);

%compute exponential weighting parametric VaR
parametric = zeros(a,1);
for i = 1:a
   e = norminv(1-percent,0,1);
   parametric(i) = v0*(1-exp(sigma(i)*sqrt(day/252)*e+(mu(i)-sigma(i)^2/2)*day/252));
end


for i = 1:a
   e = norminv(1-percent,0,1);
   parametric(i) = v0*(1-exp(sigma(i)*sqrt(day/252)*e+(mu(i)-sigma(i)^2/2)*day/252));
end

%compute windowed parametric VaR
parametric2 = zeros(a,1);
```

```
sigma2 = zeros(a,1);
mu2 = zeros(a,1);
for i = 1:a
   sigma2(i) = std(lreturn(i:i + 252*window - 1))*sqrt(252);
end
for i = 1:a
   mu2(i) = mean(lreturn(i:i+window*252-1))*252 + sigma(i)^2/2;
end
for i = 1:a
   e = norminv(1-percent,0,1);
   parametric2(i) = v0*(1-exp(sigma2(i)*sqrt(day/252)*e+(mu2(i)-sigma2(i)^2/2)*day/252));
end


%compute historical VaR, relative changes
historical = zeros(m-1,1);

for i = 1:m-1
   %construct historical samples
   hsample = zeros(m-i,1);
   for j = 1:m-i
      hsample(j) = value(i)*lreturn(j);
   end
   %compute relative losses
   loss = zeros(m-i,1);
   for j = 1:m-i
      loss(j) = value(j) - hsample(j);
   end
   historical(i) = prctile(loss,100*percent);
end


%compute Monte Carlo VaR using GBM parameters computed above
w=zeros(1,10000);
P=zeros(a,10000);
monte=zeros(a,1);

for i=1:a
   w(1,:) = mvnrnd(0,1,10000);
   P(i,:) = v0*exp((mu(i)-sigma(i)^2/2)*(day/252)+sigma(i)*w(1,:)*sqrt(day/252));
   monte(i) = quantile(10000-P(i,:),percent);

end

figure
subplot(4,1,1)      % add first plot in 4 x 1 grid
plot(parametric)
```

```
title('exponentially weighted Parametric VaR')

subplot(4,1,2)      % add first plot in 4 x 1 grid
plot(parametric2)
title('unweighted Parametric VaR')

subplot(4,1,3)      % add second plot in 4 x 1 grid
plot(historical)
title('Historical VaR')

subplot(4,1,4)      % add second plot in 4 x 1 grid
plot(monte)
title('Monte Carlo VaR')

%backtest
portreturn = zeros(m-day,1);
longportloss = zeros(m-day,1);
for i = 1:m-day
   portreturn(i) = (value(i) - value(i+day))/value(i+day);
end

for i = 1:m-day
   longportloss(i) = v0 - v0*(1+portreturn(i));
end

pexception = zeros(a,1);
p2exception = zeros(a,1);
hexception = zeros(a,1);
mexception = zeros(a,1);
for i = 1:a
   if longportloss(i)>parametric(i)
      pexception(i) = 1;
   end
   if longportloss(i)>parametric2(i)
      p2exception(i) = 1;
   end
   if longportloss(i)>historical(i)
      hexception(i) = 1;
   end
   if longportloss(i)> -monte(i)
      mexception(i) = 1;
   end
end

pexceptionpy = zeros(a-252,1);
p2exceptionpy = zeros(a-252,1);
hexceptionpy = zeros(a-252,1);
```

```
mexceptionpy = zeros(a-252,1);
for i = 1:a-252
   pexceptionpy(i) = sum(pexception(i:i+252));
   p2exceptionpy(i) = sum(p2exception(i:i+252));
   hexceptionpy(i) = sum(hexception(i:i+252));
   mexceptionpy(i) = sum(mexception(i:i+252));
end

figure
subplot(2,1,1)      % add first plot in 3 x 1 grid
plot(pexceptionpy)
title('Equivalent weighted Parametric VaR Exception')

subplot(2,1,2)      % add second plot in 3 x 1 grid
plot(p2exceptionpy)
title('unweighted Parametric VaR Exception')


xlswrite('weighted parametric VaR',parametric)
xlswrite('windowed parametric VaR',parametric2)
xlswrite('historical VaR',historical)
xlswrite('Monte Carlo VaR',monte)
xlswrite('weighted parametric exception',pexceptionpy)
xlswrite('windowed parametric exception',p2exceptionpy)
xlswrite('Acual Loss',longportloss)
```

4.2 Example

To show our results, we computed a series of 5-day 2 years VaR as an example.

Parametric VaRs

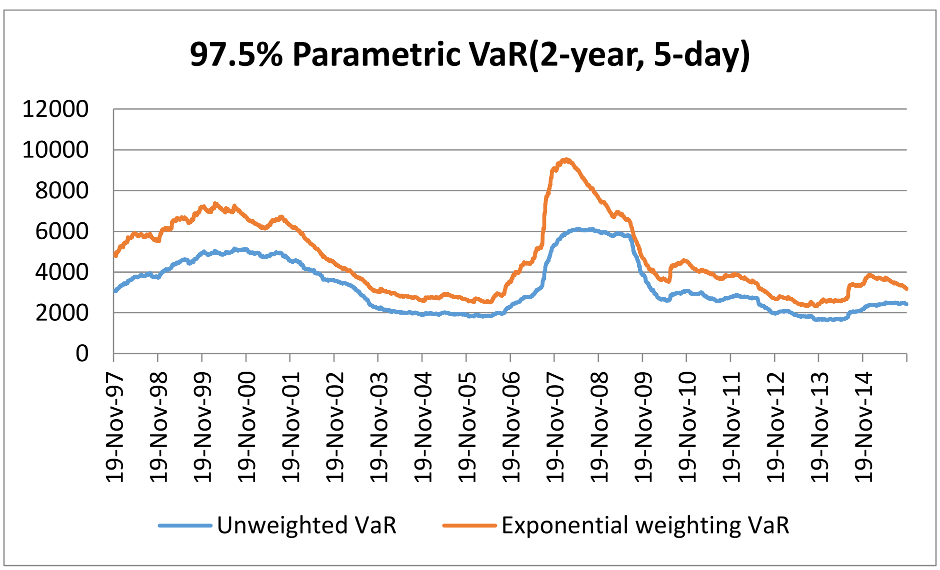

Fig 4.2.1 5-day 2-year windowed and exponentially weighted 97.5% Parametric VaR

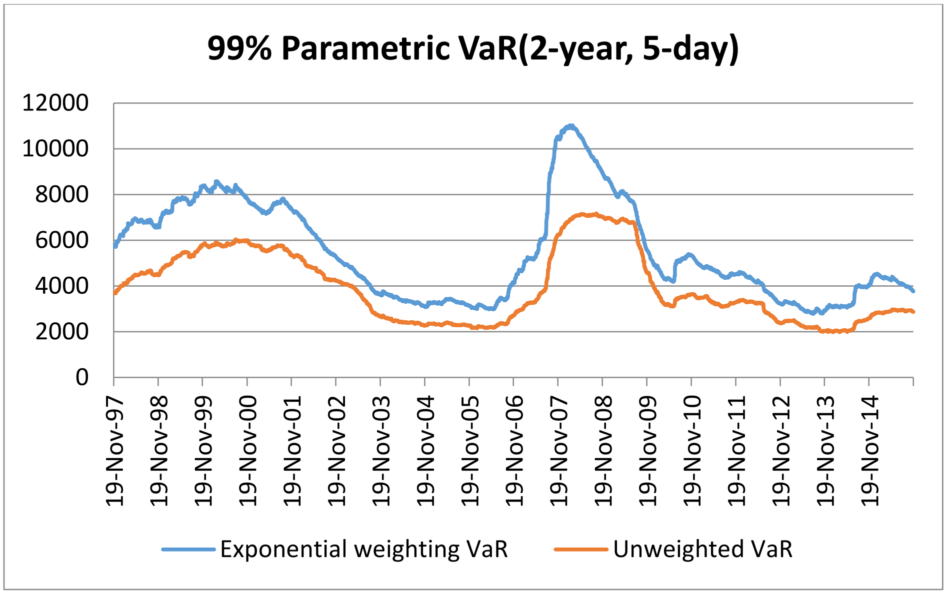

Fig 4.2.2 5-day 2-year windowed and exponentially weighted 99% Parametric VaR

Monte Carlo VaRs

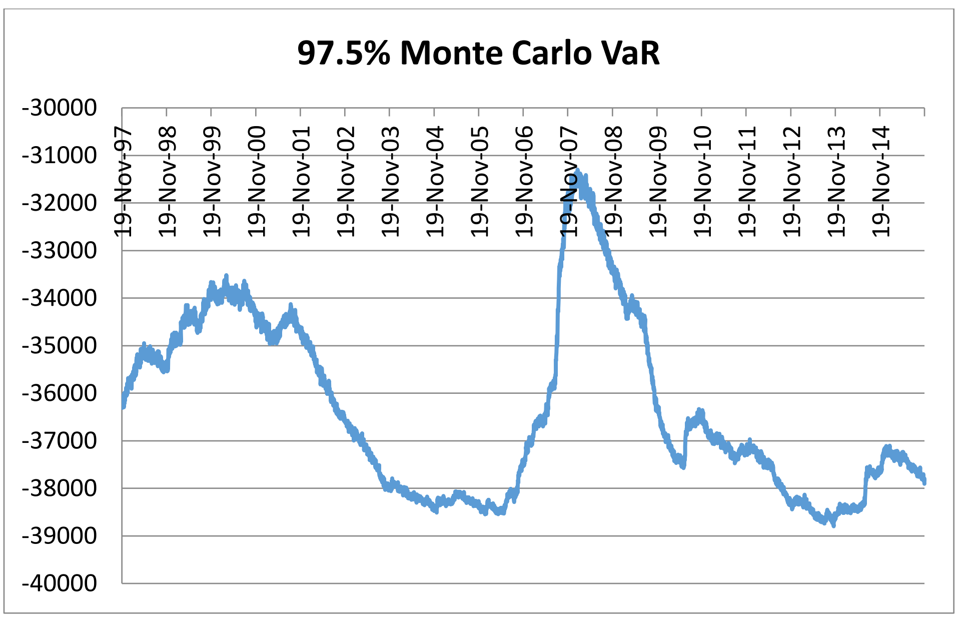

Fig 4.2.3 5-day 2-year 97.5% Monte Carlo VaR

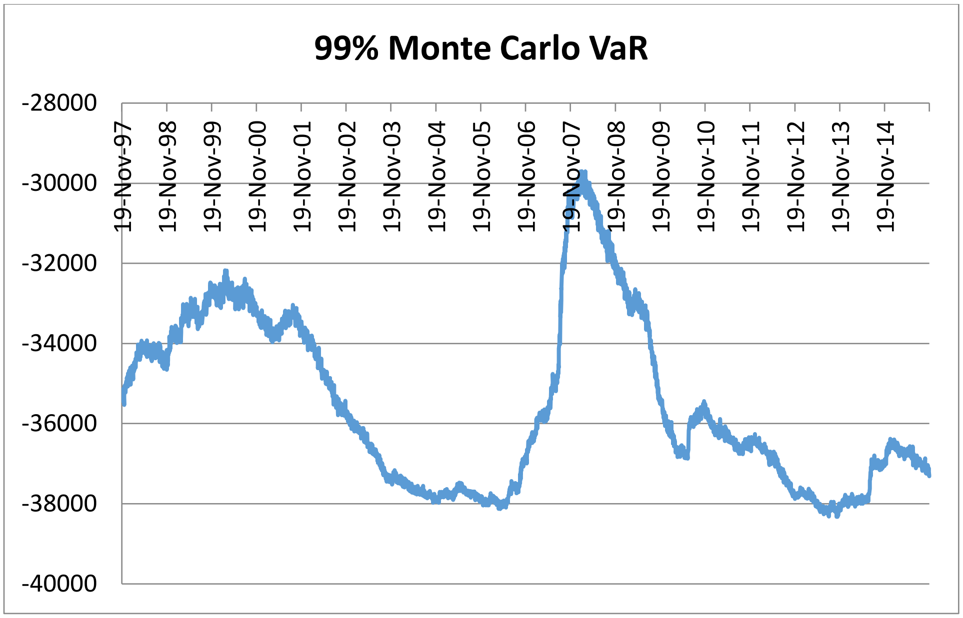

Fig 4.2.4 5-day 2-year 99% Monte Carlo VaR

Historical VaRs

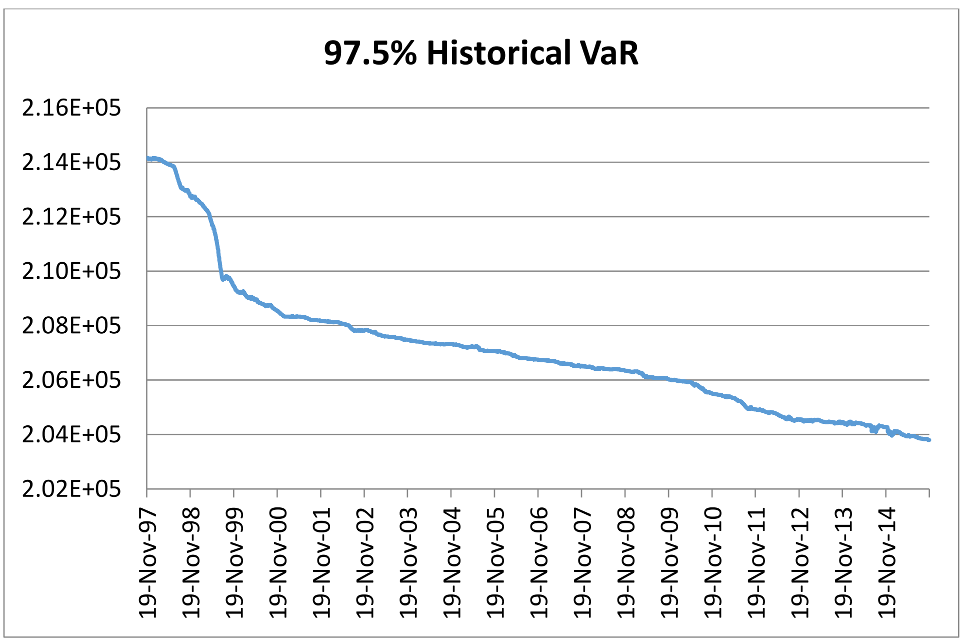

Fig 4.2.5 5-day 2-year 97.5% Historical VaR

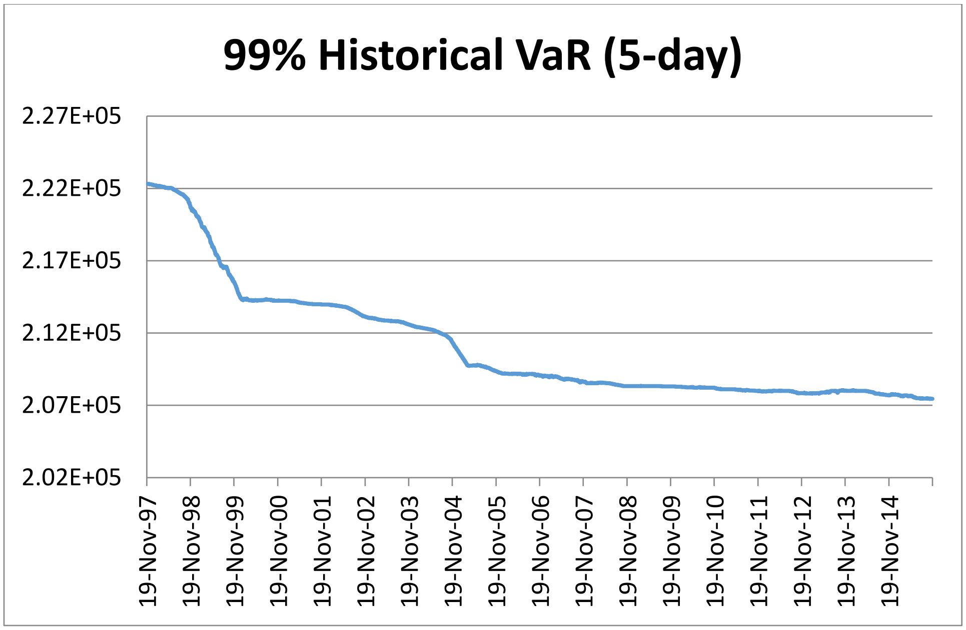

Fig 4.2.6 5-day 2-year 97.5% Historical VaR

4.3 Analysis

We exported our results to excel in order to make the plots clearer. We computed both 97.5% and 99% VaRs by using 3 different methods. Plots of parametric VaR and Monte Carlo VaR are almost same, except that of Monte Carlo VaR is noisier, which is verified by we have learned previously.

5. Test results

5.1 Backtesting result example

We used the code to calculate the exceptions of 99% and 97.5% VaR, setting 2 year window for example, 5 year and 10 year window can be calculated in the same way. Since Monte Carlo VaR turns out to have zero exception, we took Parametric VaR for further discussion.

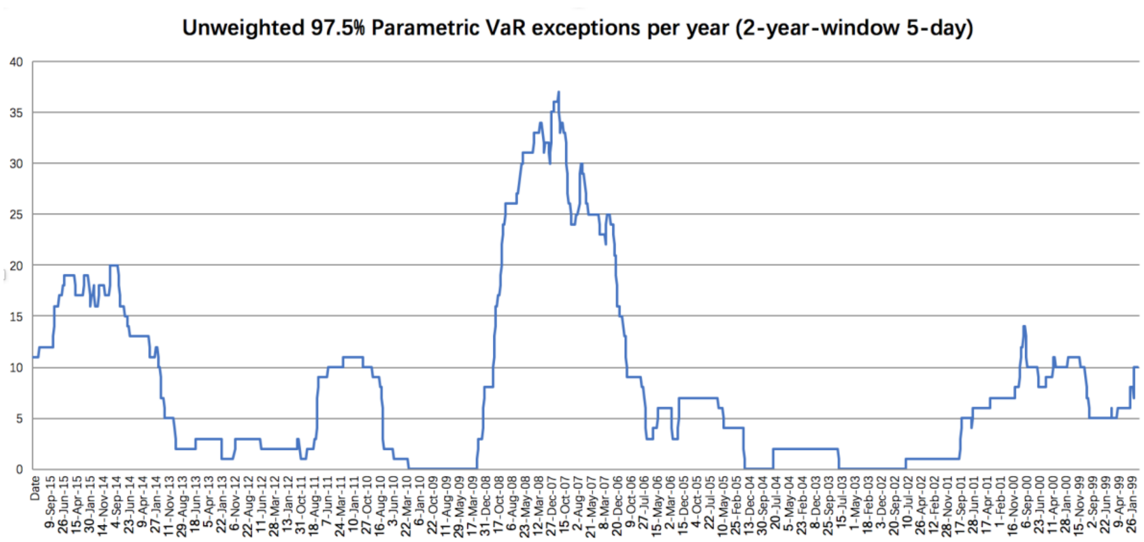

Fig 5.1.1 5-day 2-year widowed 97.5% VaR exceptions

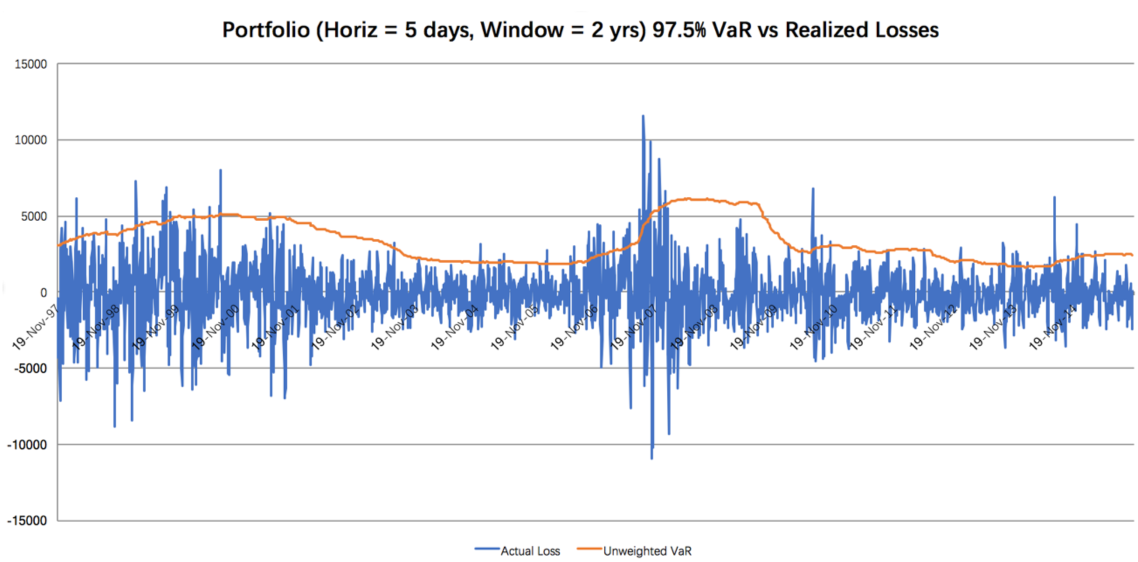

Fig 5.1.2 5-day 2-year widowed 97.5%VaR vs Realized Losses

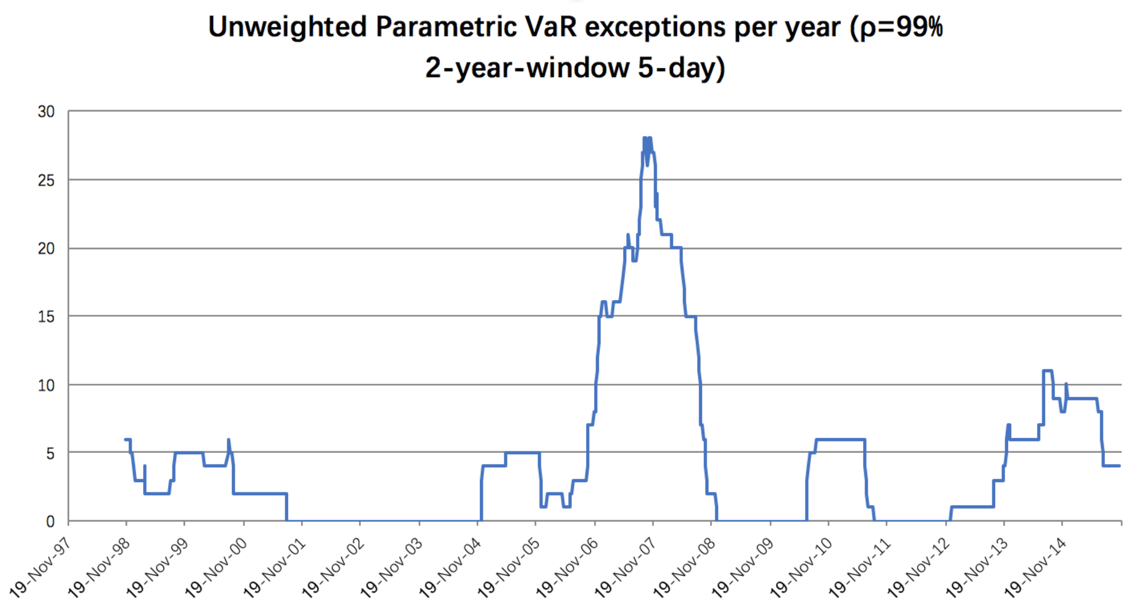

Fig 5.1.3 5-day 2-year widowed 99% VaR exceptions

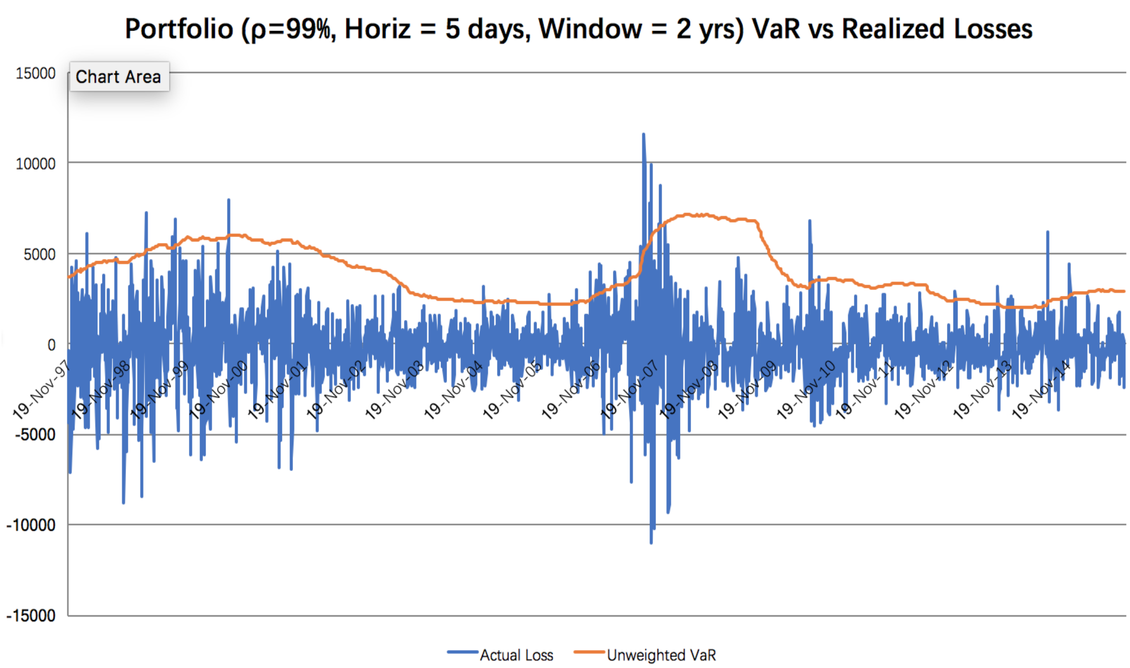

Fig 5.1.4 5-day 2-year widowed 99%VaR vs Realized Losses

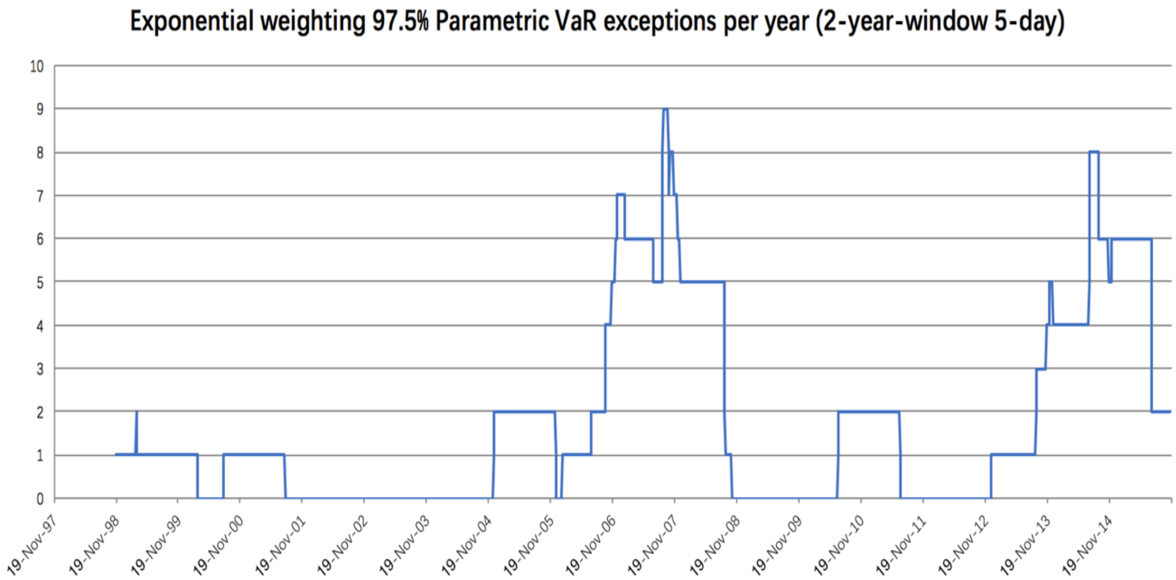

Fig 5.1.5 5-day 2-year exponentially weighted 99% VaR exceptions

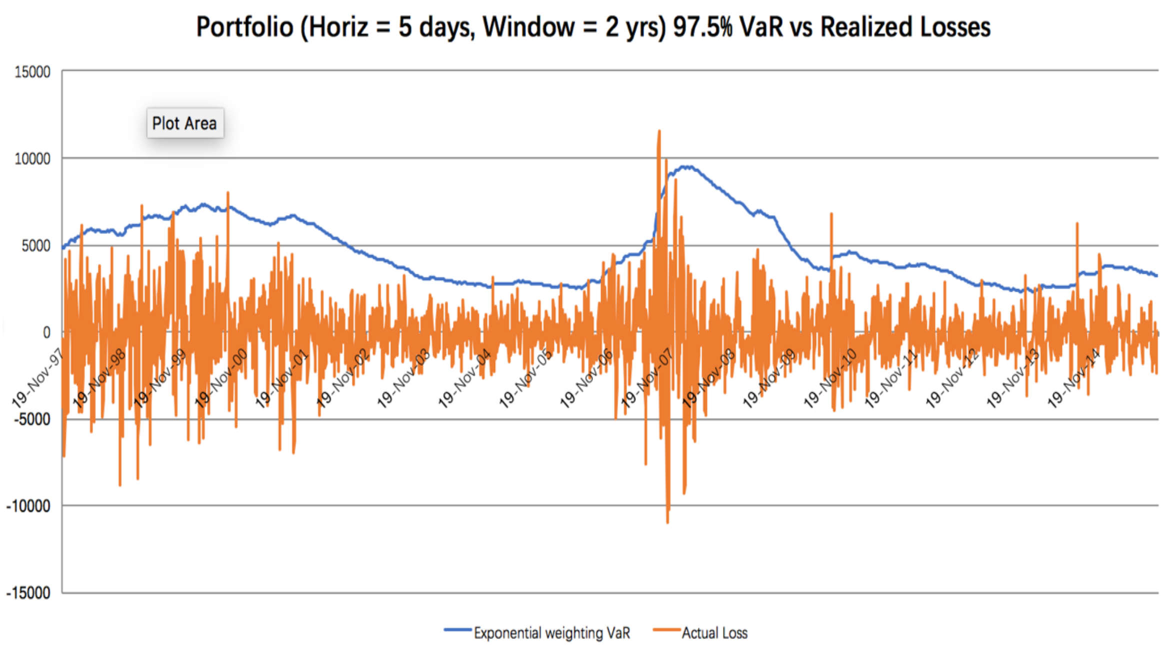

Fig 5.1.6 5-day 2-year exponentially weighted 99% VaR vs Realized Losses

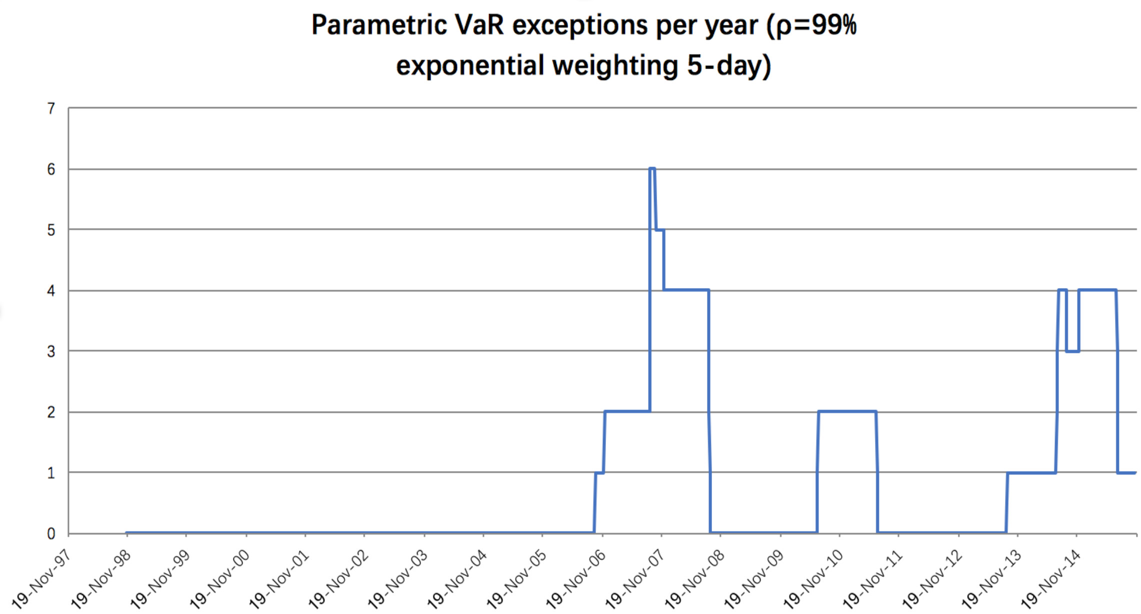

Fig 5.1.7 5-day 2-year exponentially weighted 99% VaR exceptions

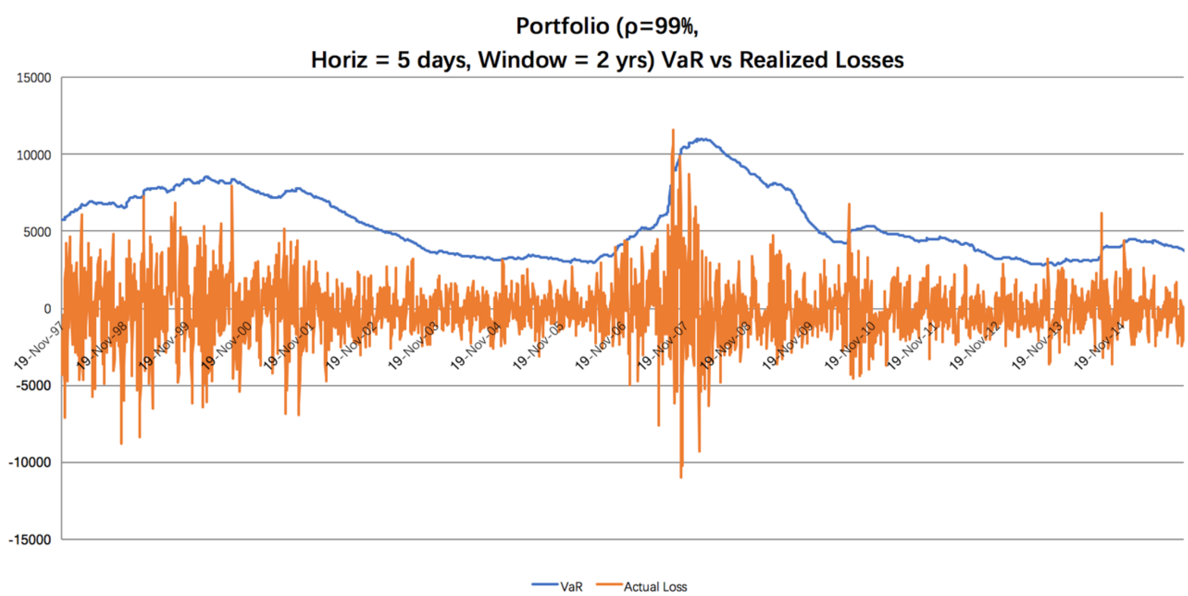

Fig 5.1.8 5-day 2-year exponentially weighted 99% VaR vs Realized Losses

5.2 Analysis

For 4277 dates in total that we compute exceptions, we have the following results:

The exponential weighting tends to yield fewer exceptions

For the unweighted 99% VaR results, 2781 of the dates are in the green zone, 1009 of the dates are in the yellow zone, and 487 of the dates are in red zone. 3848 of the dates meet the requirement that have no more than 12 exceptions per year

For the unweighted 97.5% VaR results, the exceptions tend to be more than 99% VaR results. 1930 of the dates are in the green zone, 954 of the dates are in the yellow zone and 1393 of the dates are in the red zone. 4111 of the dates meet the requirement that have no more than 30 exceptions per year

For the exponential weighting 99% VaR results, 4215 of the dates are in the green zone and the rest of them are in yellow zone. No red zone. All of the dates have no more than 12 exceptions per year

For the exponential weighting 97.5% VaR results, 3551 of the dates are in the green zone and the rest of them are in the yellow zone. No red zone. All of the dates have no more than 30 exceptions per year

Above all, we can conclude that our model tests well, since the number of exceptions meets the test requirement in most of the dates.