409-Risk Managment HW3

Ming-Hao Yu

2018-04-22

1 Choosing a VaR technique

Download the excel file which contains the time series of gains for a strategy from 8/8/2006 to 9/25/2008.

1. For each day in 2007-2008, compute historical VaR and exponential weighted 1-day 99%-VaR. Comment on the exceptions that happen with these two measures.

```
library(data.table)
library(lubridate)
```

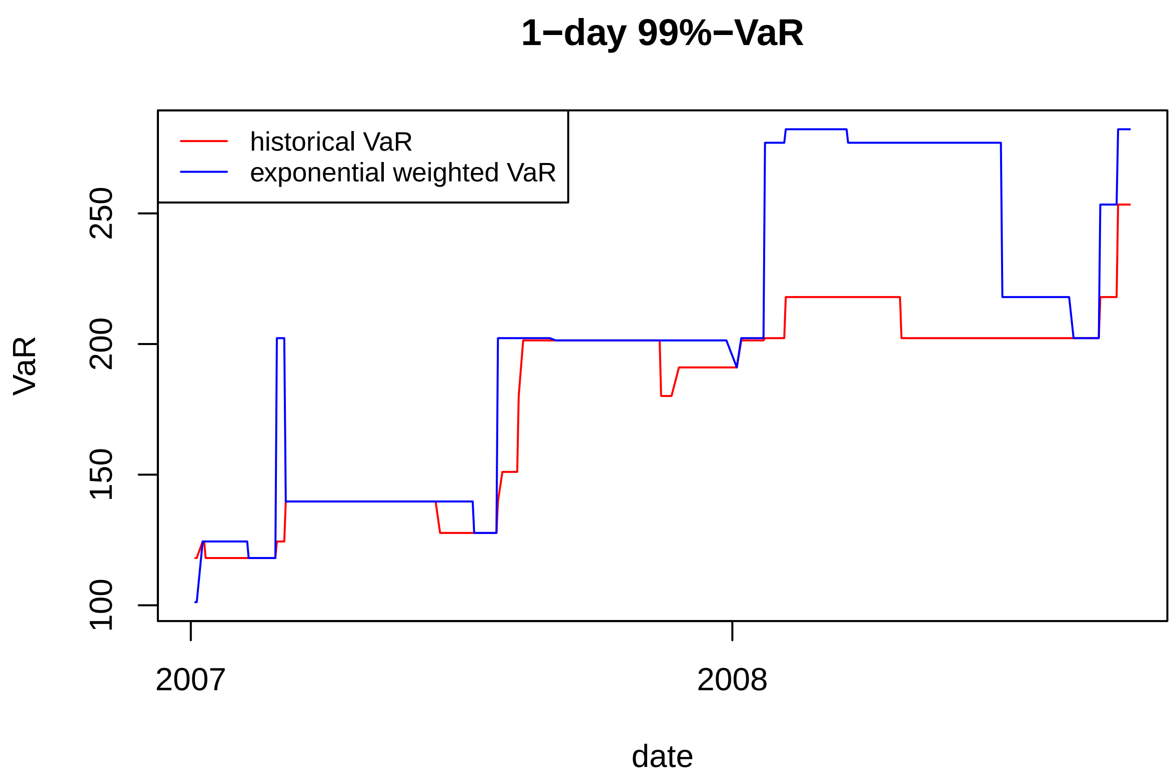
```
## Warning: package 'lubridate' was built under R version 3.4.3
##
## Attaching package: 'lubridate'
## The following objects are masked from 'package:data.table':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday,
##     week, yday, year
## The following object is masked from 'package:base':
##
##     date
```

```
library(tseries)
```

```
## Warning: package 'tseries' was built under R version 3.4.3
```

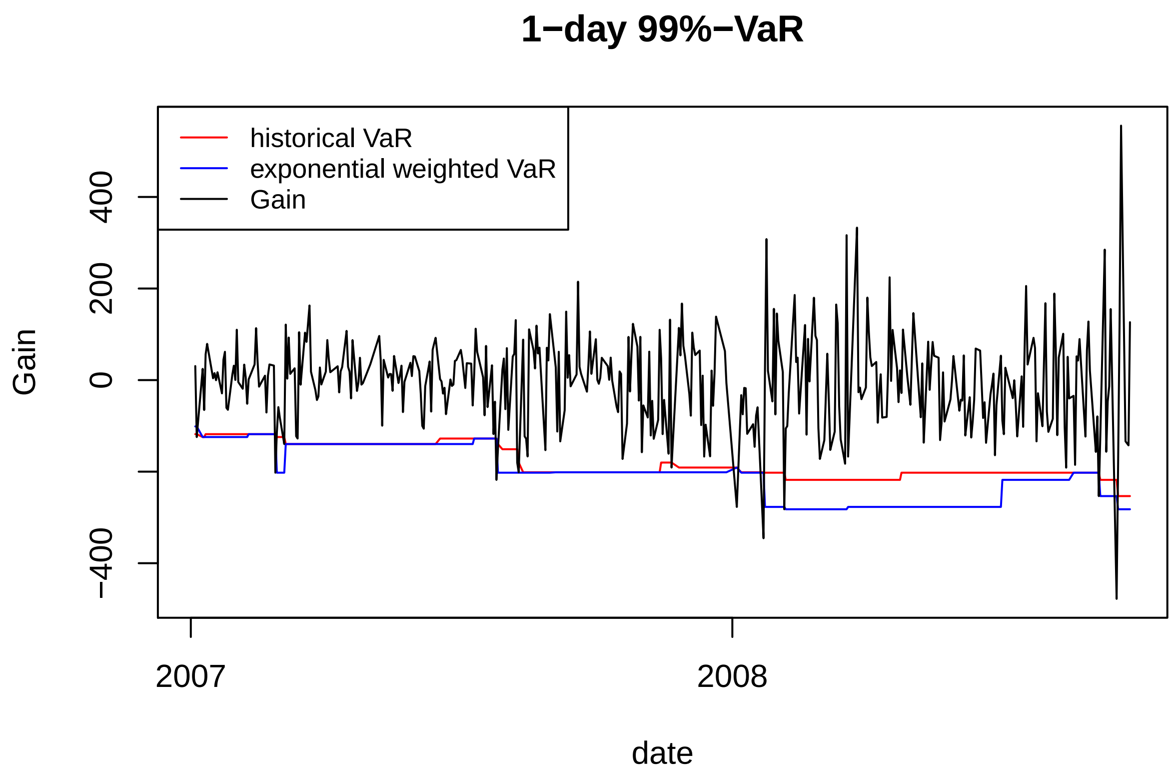
```
setwd("C:\\Users\\Ming-Hao\\Desktop\\MFE\\409-Financial Risk Measurement and Management")

if(!exists("hw3")){
    hw3 = fread(file="homework3_data.csv")
    hw3[, date:= as.Date(date, "%Y/%m/%d")]
}

#Q1
start = which(hw3$date >= as.Date("2007-01-01", "%Y-%m-%d"))[1]
VaR = VaRexp = VaRb = VaRexpb = vector()
upper = lower = upperb = lowerb = upperexpb = lowerexpb = vector()
j = 1

for(i in start:length(hw3$date)){
    data = data.table(hw3[1:(i-1), ])
    lambda = 0.995
    n = length(data$date)
    ii = seq(1,n)
    weight = lambda^(n-ii)*(1-lambda)/(1-lambda^n)
    data[, Weight := weight]
    VaR[j] = abs(sort(data$gain)[ceiling(0.01*i)])

    data = data[order(data$gain), ]
```

```
    ptr = which(cumsum(data$Weight)>=0.01)[1]
    VaRexp[j] = abs(data$gain[ptr])

    mu = mean(data$gain)
    sigma = sd(data$gain)
    x = mu+qnorm(0.01)*sigma
    f = exp(-(x-mu)^2/2/sigma^2)/sigma/sqrt(2*pi)
    StdDev = 1/f*sqrt(0.99*0.01/i)

    upper[j] = VaR[j] + 2*StdDev
    lower[j] = VaR[j] - 2*StdDev
    j = j+1
}

ylim=c(min(VaR, VaRexp), max(VaR,VaRexp))
plot(x=hw3$date[which(hw3$date >= as.Date("2007-01-01", "%Y-%m-%d"))],
     y=VaR, xlab="date", ylab="VaR", main="1-day 99%-VaR", type="l", col="red", ylim=ylim)
points(x=hw3$date[which(hw3$date >= as.Date("2007-01-01", "%Y-%m-%d"))],
       y=VaRexp, col="blue", type="l")
legend("topleft", c("historical VaR", "exponential weighted VaR"),
       col=c("red", "blue"), cex=0.8, lwd=1)
```

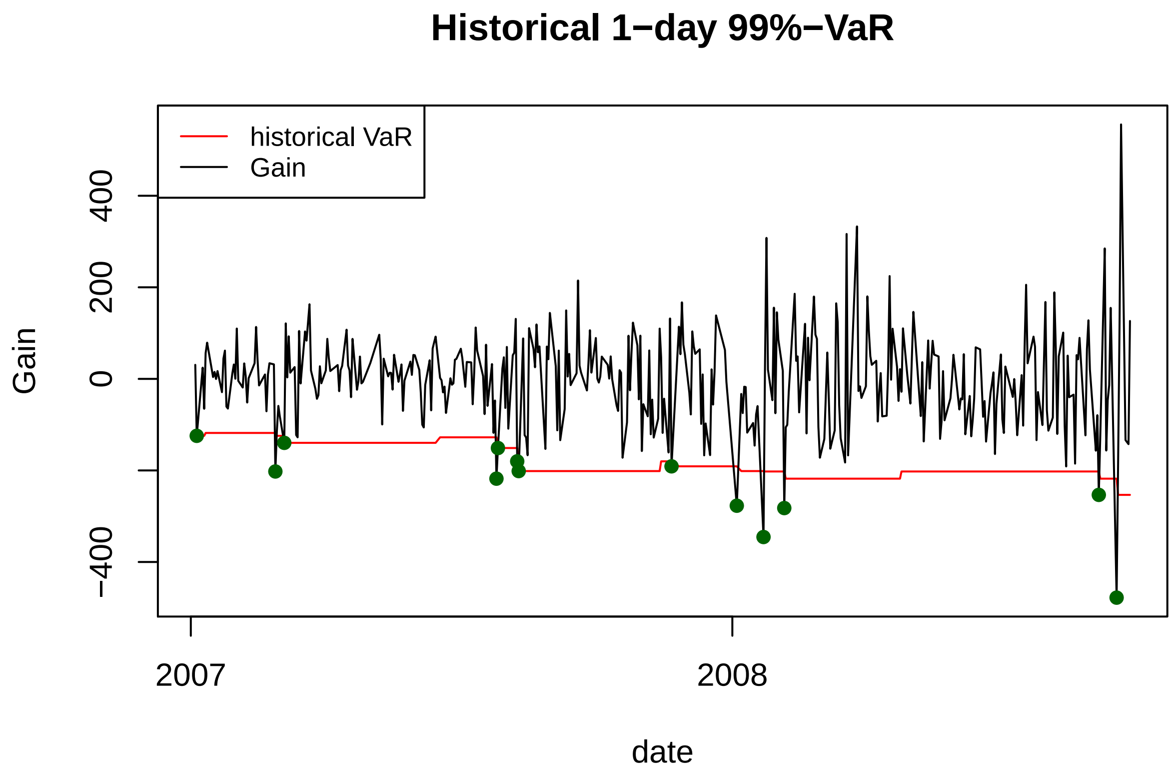



```
ylim=c(min(VaR, VaRexp, hw3$gain), max(VaR,VaRexp, hw3$gain))
plot(x=hw3$date[which(hw3$date >= as.Date("2007-01-01", "%Y-%m-%d"))],
     y=-VaR, xlab="date", ylab="Gain", main="1-day 99%-VaR", type="l", col="red", ylim=ylim)
points(x=hw3$date[which(hw3$date >= as.Date("2007-01-01", "%Y-%m-%d"))],
```

```
       y=-VaRexp, col="blue", type="l")
points(x=hw3$date[which(hw3$date >= as.Date("2007-01-01", "%Y-%m-%d"))],
       y=hw3$gain[which(hw3$date >= as.Date("2007-01-01", "%Y-%m-%d"))],
       type="l", col="black")
legend("topleft", c("historical VaR", "exponential weighted VaR", "Gain"),
       col=c("red", "blue", "black"), cex=0.8, lwd=1)
```



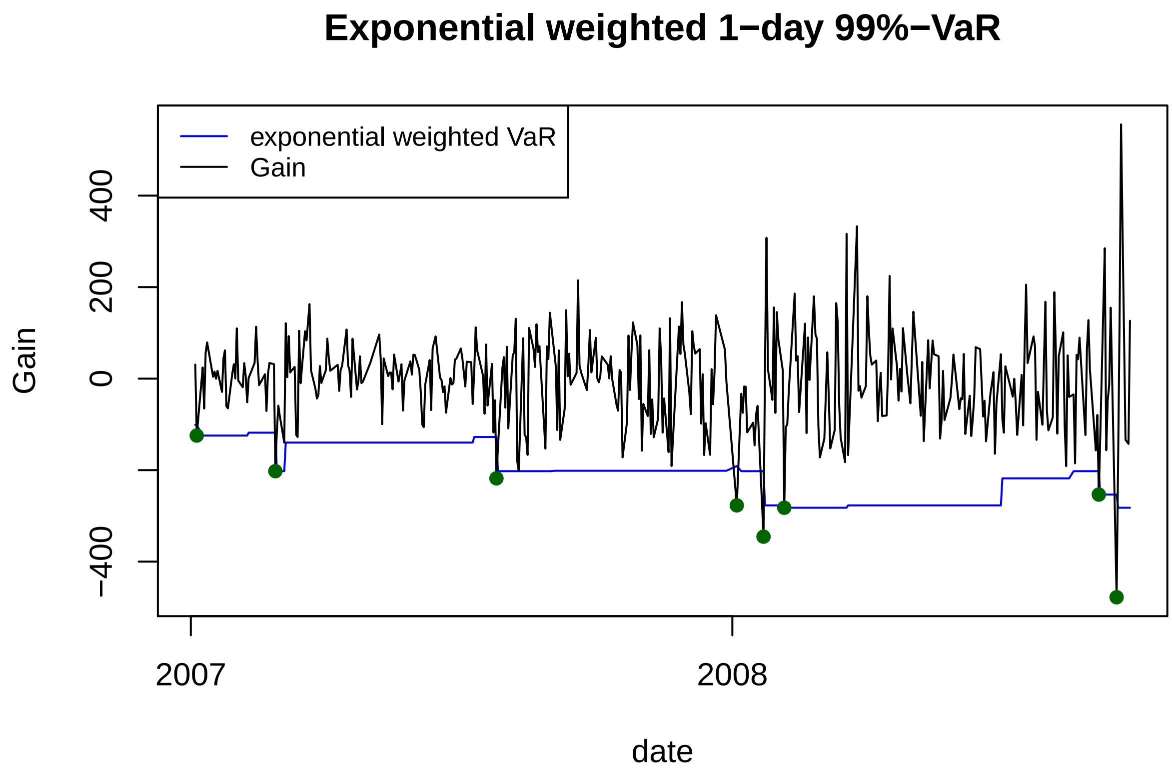

```
#calculate exception of VaR/VaRexp
exception1 = which(hw3$gain[which(hw3$date >= as.Date("2007-01-01", "%Y-%m-%d"))] < -VaR)+
  (length(hw3$gain)-length(VaR))
exception2 = which(hw3$gain[which(hw3$date >= as.Date("2007-01-01", "%Y-%m-%d"))] < -VaRexp)+
  (length(hw3$gain)-length(VaR))

#Historical VaR
plot(x=hw3$date[which(hw3$date >= as.Date("2007-01-01", "%Y-%m-%d"))],
     y=-VaR, xlab="date", ylab="Gain", main="Historical 1-day 99%-VaR",
     type="l", col="red", ylim=ylim)
points(x=hw3$date[which(hw3$date >= as.Date("2007-01-01", "%Y-%m-%d"))],
       y=hw3$gain[which(hw3$date >= as.Date("2007-01-01", "%Y-%m-%d"))],
       type="l", col="black")
points(x=hw3$date[exception1], y=hw3$gain[exception1],
       col="dark green", type="p", pch=16)
legend("topleft", c("historical VaR", "Gain"),
       col=c("red", "black"), cex=0.8, lwd=1)
```




```
print(c("exception:", length(exception1)))
```

```
## [1] "exception:" "13"
```

```
#exponential VaR
plot(x=hw3$date[which(hw3$date >= as.Date("2007-01-01", "%Y-%m-%d"))],
     y=-VaRexp, xlab="date", ylab="Gain", main="Exponential weighted 1-day 99%-VaR",
     type="l", col="blue", ylim=ylim)
points(x=hw3$date[which(hw3$date >= as.Date("2007-01-01", "%Y-%m-%d"))],
       y=hw3$gain[which(hw3$date >= as.Date("2007-01-01", "%Y-%m-%d"))],
       type="l", col="black")
points(x=hw3$date[exception2], y=hw3$gain[exception2], col="dark green", type="p", pch=16)
legend("topleft", c("exponential weighted VaR", "Gain"),
       col=c("blue", "black"), cex=0.8, lwd=1)
```




```
print(c("exception:", length(exception2)))
```

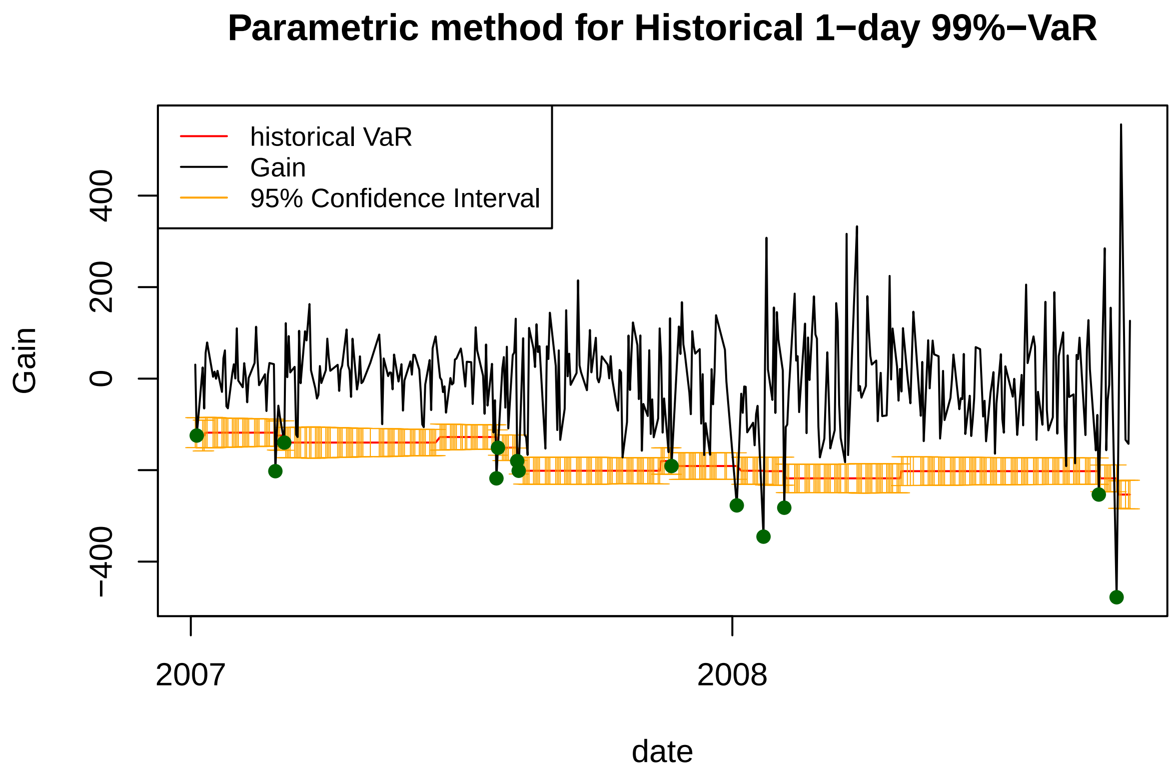
```
## [1] "exception:" "8"
```

We can observe from the figure 1 (1-day 99%-VaR) that the exponential weighted VaR is more sensetive to jump in the latest data because not only the scale of the jump but also the weight assigned to the nearest data. Only a tail loss happens in the next period, it will greatly contribute to the exponential weighted VaR while the equal weighted historical will not.

2. For each day in the sample, compute the 95% confidence intervals of the historical VaR and the exponential weighted VaR you obtained in Question 1, using both parametric (for the historical VaR) and bootstrap methods (for the two measures). For the parametric method, assume the gains are normally distributed.

```
#Q2
j = 1

for(i in start:length(hw3$date)) {
    data = data.table(hw3[1:(i-1), ])
    lambda = 0.995
    n = length(data$date)
    ii = seq(1,n)
    weight = lambda^(n-ii)*(1-lambda)/(1-lambda^n)
    data[, Weight := weight]
    VaR[j] = abs(sort(data$gain)[ceiling(0.01*i)])

    data = data[order(data$gain), ]
    v1 = v2 = vector()
```

```r
    #bootstrap historical VaR
    sample1 = matrix(sample(data$gain, size=i*1000, replace=T), nrow=i, ncol=1000)
    v1 = apply(sample1, MARGIN=2, FUN=function(x){return(abs(sort(x)[ceiling(0.01*i)]))})
    VaRb[j] = mean(v1)
    upperb[j] = quantile(v1, 0.975)
    lowerb[j] = quantile(v1, 0.025)

    #bootstrap exponential weighted VaR
    sample2 = matrix(sample(data$gain, size=i*1000, replace=T,
                            prob=data$Weight), nrow=i, ncol=1000)
    v2 = apply(sample2, MARGIN=2, FUN=function(x){
        return(abs(sort(x)[ceiling(0.01*i)]))
    })
    VaRexpb[j] = mean(v2)
    upperexpb[j] = quantile(v2, 0.975)
    lowerexpb[j] = quantile(v2, 0.025)
    j = j+1
}

#Historical paramteric method
Q2p = matrix(nrow=length(VaR), ncol=3, c(lower, VaR, upper))
colnames(Q2p) = c("lwr (2.5%)", "VaR", "upr (97.5%)")

plot(x=hw3$date[which(hw3$date >= as.Date("2007-01-01", "%Y-%m-%d"))],
     y=-VaR, xlab="date", ylab="Gain", main="Parametric method for Historical 1-day 99%-VaR",
     type="l", col="red", ylim=ylim)
arrows(x0=hw3$date[which(hw3$date >= as.Date("2007-01-01", "%Y-%m-%d"))],
       y0=-Q2p[, 1], y1=-Q2p[, 3], lwd=0.5, angle=90, code=3, length=0.05, col="orange")
points(x=hw3$date[which(hw3$date >= as.Date("2007-01-01", "%Y-%m-%d"))],
       y=hw3$gain[which(hw3$date >= as.Date("2007-01-01", "%Y-%m-%d"))],
       type="l", col="black")
points(x=hw3$date[exception1], y=hw3$gain[exception1], col="dark green", type="p", pch=16)
legend("topleft", c("historical VaR", "Gain", "95% Confidence Interval"),
       col=c("red", "black", "orange"), cex=0.8, lwd=1)
```

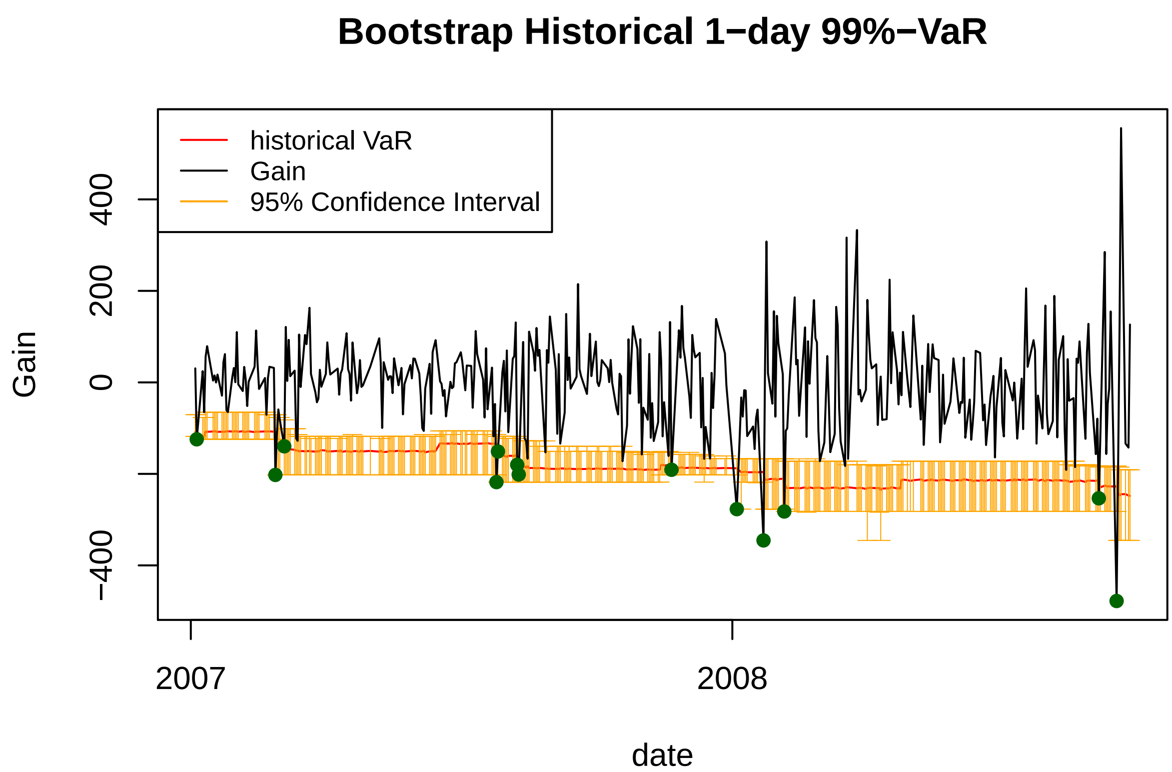



```
head(Q2p, 20)
```

```
##       lwr (2.5%)      VaR upr (97.5%)
##  [1,]   85.30454 118.0893    150.8741
##  [2,]   85.60678 118.0893    150.5719
##  [3,]   90.68167 124.4312    158.1808
##  [4,]   91.00401 124.4312    157.8584
##  [5,]   84.53049 118.0893    151.6482
##  [6,]   84.67917 118.0893    151.4995
##  [7,]   84.61997 118.0893    151.5587
##  [8,]   84.94191 118.0893    151.2368
##  [9,]   85.25864 118.0893    150.9201
## [10,]   85.56307 118.0893    150.6156
## [11,]   85.86615 118.0893    150.3125
## [12,]   86.05176 118.0893    150.1269
## [13,]   86.25843 118.0893    149.9203
## [14,]   86.35329 118.0893    149.8254
## [15,]   86.27702 118.0893    149.9017
## [16,]   86.16841 118.0893    150.0103
## [17,]   86.45288 118.0893    149.7258
## [18,]   86.69874 118.0893    149.4800
## [19,]   86.96846 118.0893    149.2102
## [20,]   86.56081 118.0893    149.6179
```

```
#Bootstraping
Q2 = matrix(nrow=length(VaRb), ncol=3, c(lowerb, VaRb, upperb))
colnames(Q2) = c("lwr (2.5%)", "VaR", "upr (97.5%)")
```

```r
plot(x=hw3$date[which(hw3$date >= as.Date("2007-01-01", "%Y-%m-%d"))],
     y=-VaRb, xlab="date", ylab="Gain", main="Bootstrap Historical 1-day 99%-VaR",
     type="l", col="red", ylim=ylim)
arrows(x0=hw3$date[which(hw3$date >= as.Date("2007-01-01", "%Y-%m-%d"))],
       y0=-Q2[, 1], y1=-Q2[, 3], lwd=0.5, angle=90, code=3, length=0.05, col="orange")
points(x=hw3$date[which(hw3$date >= as.Date("2007-01-01", "%Y-%m-%d"))],
       y=hw3$gain[which(hw3$date >= as.Date("2007-01-01", "%Y-%m-%d"))],
       type="l", col="black")
points(x=hw3$date[exception1], y=hw3$gain[exception1], col="dark green", type="p", pch=16)
legend("topleft", c("historical VaR", "Gain", "95% Confidence Interval"),
       col=c("red", "black", "orange"), cex=0.8, lwd=1)
```

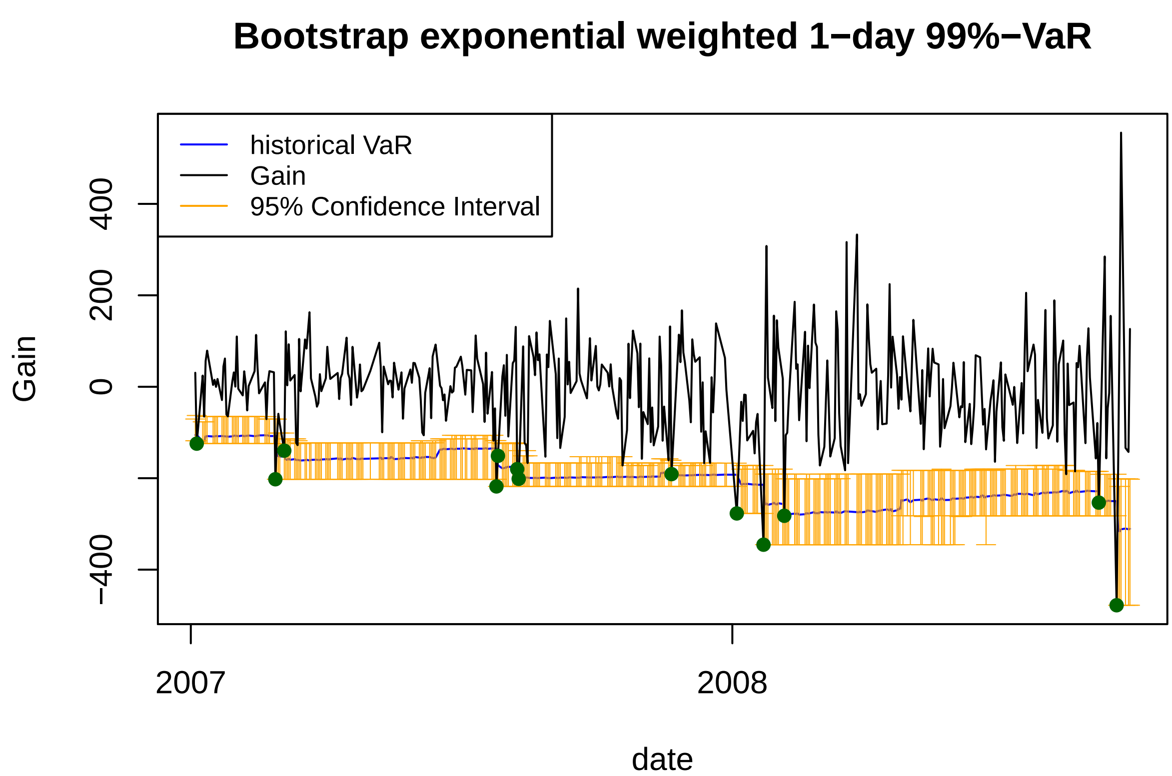



```r
head(Q2, 20)
```

```
##       lwr (2.5%)      VaR upr (97.5%)
##  [1,]   70.54522 108.1663    118.0893
##  [2,]   70.54522 108.8868    118.0893
##  [3,]   76.72657 118.0266    124.4312
##  [4,]   76.72657 118.4525    124.4312
##  [5,]   65.27459 107.4863    124.4312
##  [6,]   65.27459 108.0439    124.4312
##  [7,]   65.27459 107.3061    124.4312
##  [8,]   65.27459 106.5890    124.4312
##  [9,]   65.27459 107.6162    124.4312
## [10,]   65.27459 108.0541    124.4312
## [11,]   65.27459 107.8456    124.4312
```

```
## [12,]    65.27459 108.0532    124.4312
## [13,]    65.27459 107.7303    124.4312
## [14,]    65.27459 107.0075    124.4312
## [15,]    65.27459 106.9732    124.4312
## [16,]    65.27459 107.3942    124.4312
## [17,]    65.27459 107.1842    124.4312
## [18,]    65.27459 106.7242    124.4312
## [19,]    65.27459 106.9092    124.4312
## [20,]    65.27459 107.8996    124.4312
```

```r
Q2exp = matrix(nrow=length(VaRexpb), ncol=3, c(lowerexpb, VaRexpb, upperexpb))
colnames(Q2exp) = c("lwr (2.5%)", "VaR", "upr (97.5%)")

plot(x=hw3$date[which(hw3$date >= as.Date("2007-01-01", "%Y-%m-%d"))],
     y=-VaRexpb, xlab="date", ylab="Gain", main="Bootstrap exponential weighted 1-day 99%-VaR",
     type="l", col="blue", ylim=ylim)
arrows(x0=hw3$date[which(hw3$date >= as.Date("2007-01-01", "%Y-%m-%d"))],
       y0=-Q2exp[, 1], y1=-Q2exp[, 3], lwd=0.5, angle=90, code=3, length=0.05, col="orange")
points(x=hw3$date[which(hw3$date >= as.Date("2007-01-01", "%Y-%m-%d"))],
       y=hw3$gain[which(hw3$date >= as.Date("2007-01-01", "%Y-%m-%d"))],
       type="l", col="black")
points(x=hw3$date[exception1], y=hw3$gain[exception1], col="dark green", type="p", pch=16)
legend("topleft", c("historical VaR", "Gain", "95% Confidence Interval"),
       col=c("blue", "black", "orange"), cex=0.8, lwd=1)
```



```r
head(Q2exp, 20)
```

```
##       lwr (2.5%)      VaR upr (97.5%)
##  [1,]   70.54522 105.4630    118.0893
##  [2,]   62.82088 105.2308    118.0893
##  [3,]   76.72657 119.3381    124.4312
##  [4,]   76.72657 119.1490    124.4312
##  [5,]   65.27459 108.4304    124.4312
##  [6,]   65.27459 107.9088    124.4312
##  [7,]   65.27459 108.7081    124.4312
##  [8,]   65.27459 108.6291    124.4312
##  [9,]   65.27459 107.3769    124.4312
## [10,]   65.27459 108.4034    124.4312
## [11,]   65.27459 107.6572    124.4312
## [12,]   65.27459 107.9978    124.4312
## [13,]   65.27459 108.6365    124.4312
## [14,]   65.27459 108.4009    124.4312
## [15,]   65.27459 109.2028    124.4312
## [16,]   65.27459 108.4576    124.4312
## [17,]   65.27459 106.8148    124.4312
## [18,]   65.27459 107.2878    124.4312
## [19,]   65.27459 107.5268    124.4312
## [20,]   65.27459 108.0529    124.4312
```

3. Assess the normality of the gain distribution.

```r
#Q3
jarque.bera.test(hw3$gain)
```

```
##
##  Jarque Bera Test
##
## data:  hw3$gain
## X-squared = 362.07, df = 2, p-value < 2.2e-16
```

p-value is much smaller than 0.05 which implies that we should reject the Null-hypothesis; therefore, the gain is NOT normal distribution.

4. For each day in the sample, compute the volatility of the portfolio in the past month. Normalize gains with estimated volatility. Compare the distribution of the normalized gain with the original one and discuss which is closer to normal distribution.

```r
#Q4
j = 1
gainNormal = vector()

for(i in start:length(hw3$date)) {
    volatility = sd(hw3$gain[(i-30):(i-1)])
    gainNormal[j] = (hw3$gain[i]-mean(hw3$gain[(i-30):(i-1)]))/volatility
    j = j+1
}

#Normalized gain
jarque.bera.test(gainNormal)
```

```
##
##  Jarque Bera Test
```

```
##
## data:  gainNormal
## X-squared = 24.055, df = 2, p-value = 5.976e-06
```

```r
#Original gain
jarque.bera.test(hw3$gain[start:(length(hw3$date))])
```

```
##
##  Jarque Bera Test
##
## data:  hw3$gain[start:(length(hw3$date))]
## X-squared = 187.64, df = 2, p-value < 2.2e-16
```

Normalized gain has smaller p-value (5.976e-06) than the orighinal one (2.2e-16) so that the normailzed gain is closer to the normal distribution than the original one.

5. Repeat Question 4 but assuming that somebody tells you in advance what the volatility will be in the next month.

```r
#Q5
j = 1
gainAdvancedNormal = vector()

for(i in start:(length(hw3$date)-30)) {
    volatility = sd(hw3$gain[(i+1):(i+30)])
    gainAdvancedNormal[j] = (hw3$gain[i]-mean(hw3$gain[(i-30):(i-1)]))/volatility
    j = j+1
}
jarque.bera.test(gainAdvancedNormal)
```

```
##
##  Jarque Bera Test
##
## data:  gainAdvancedNormal
## X-squared = 22.167, df = 2, p-value = 1.536e-05
```

The p-value (1.536e-05) of gain normalized by the future volatility is smaller than the above two to the gain normalized by the previous one month volatility, which is also much close to normal distribution compared to the original gain.

6. Write a proposal to the head of trading for measuring the risk of this trade in real time, justifying your choices.

VaR measurement: Normalized historical gain by one-month-ahead portfolio volatility to generate normal-distribution-liked gains. Calculated each day's normalized VaR with the historical data. Scale back by volatility calculated at that day.

Since in our VaR model, we assume gains are normal distributions, we should use normal-distribution-liked gain for the calculation of VaR. We can see from the JB test of the original gain that the original gain is far from the normal distribution. Thus we should use the normalized gain for our calculation of VaR. Besides, unless we have perfect forecast of the future volatility, we should use the historical volatility as the normalization factor. Q4 provides a result of the normalized gain is closer to a normal distribution.

2 Interview Questions

1. How many independent random variables uniformly distributed on [0; 1] should you generate to ensure that there is at least one between 0.70 and 0.72 with probability 95%?

```r
p = function(x) {return((0.98)^x-0.05)}
uniroot(p, c(-1000, 100000))$root
```

```
## [1] 148.2837
```

Each time we draw from the random variable, we will only have the number between 0.70 and 0.72 with $\frac{0.2}{1.0}$ probability. In order words, we will have 0.98 probability to fail to have the number between 0.70 and 0.72. Having a 95% probability to have at least one draw between 0.70 and 0.72 represents that we have 5% probability that every time of the draw fail to be between 0.70 to 0.72. So that, assume we draw n times, 0.98^n = 0.05, and solve the n. We need at least n=148.2837 independetn random variables uniformaly distributed on [0:1].

2. What is the ten-day 99% VaR of a portfolio with a five-day 98% VaR of $10 million?

Since we don't know the real distribution of the return, we cannot calculate the VaR. We can only give approximated answer: Because of the 98% 5-day VaR, we have 0.02 probability to loss more than 10million. We want to know the 99% 10-day VaR, we can see this situation as two draws. Then there are three situations: (1) 0.02×0.02: loss more than 20 million (2) $0.02 \times 0.98 \times 2$: loss more than 10 million + loss less than 10 million (3) 0.98×0.98: loss less than 10 millin + loss less than 10 million We can see that 99% $(1-0.99 = 0.01)$ falls in the second situation. However, without the real distribution, we still cannot derive the 99% 10-day VaR.

3. How would you compute π using Monte Carlo simulations? What is the standard deviation of this method?

1. Construct 2 independent uniformal distribution random variables X, Y range from 0 to 1, representing the coordinate.
2. Generate 10000 (simulation times) points with paired X and Y on the X-Y coordinate.
3. Compute those points' distance to origin point (0,0), then count the number of points with the distance that is smaller than 1.
4. Counted number divided by the total simulation (e.g. 10000) will be the probability that a point falling in the quarter circle, also is the ratio of the area of a quarter circle to a 1x1 square, which is $\frac{\pi r^2}{4} : r^2, where r = 1$. So that, we have probability, which is the ratio, and then we can solve the π.

$$\pi = 4 \times Probability$$

4. What is the Gamma of an option? Why is it preferable to have small Gamma? Why is the Gamma of plain vanilla options positive?

(1) Gamma is the $\frac{\partial \Delta}{\partial S}$, a mearsurement of a option's delta sensitivity to the change of the underlying asset.

(2) If Gamma is samll, we don't have to worry about gamma hedging too much, which reduce the cost of hedging.

(3) We can see from the payoff of call and put vanilla options that the second derivative of both payoff are positive. For call option, the payoff starts with slope 0, and the the slope gets higher, take the derivative of S on the slope $\frac{\partial Slope}{\partial S}$, it is a positive change; For a put option, the slope starts from negative, and then goes to 0, it is also a positive change.