CUHK RMSC4002 Tutorial 5

Benjamin Chan

October 23, 2018

Value at Risk (VaR)

Read in Data

```
# Read in data (a CSV file) under Dataset
d <- read.csv("./../Dataset/stock_2018.csv")
d <- d[, 2:4]                               # Remove Date

class(d)                                    # class: print the names of classes an object
```

```
[1] "data.frame"
```

```
is.matrix(d)
```

```
[1] FALSE
```

```
x <- as.matrix(d)                           # Turn it into matrix
n <- nrow(x)                                # Number of observations
```

Historical Simulation

Suppose that today is day n and v_i is the value of a market variable (stock price or index). Then the value tomorrow estimated based on the i-th scenario is $\hat{v}(i) = v_n \times \frac{v_i}{v_{i-1}}$ for $i = 1, \ldots, n$. The VaR is calculated by the quantile of the n simulated values.

Suppose we spend \$40,000 on buying HSBC, \$30,000 on CLP and \$30,000 on CK on 31 Aug 2018. Mathematically speaking, let

$$w = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = \begin{bmatrix} 40,000 \\ 30,000 \\ 30,000 \end{bmatrix}$$

be amounts of investment in stocks.

We are going to compute the 1-day VaR of this portfolio using historical simulation.

```
xn <- as.vector(x[n, ])                     # Select the last observation
w <- c(40000, 30000, 30000)                 # Amount on each stock
p0 <- sum(w)                                # Initial total amount
ws <- w/xn                                  # Number of shares bought at day n

ns <- n-1                                   # Number of scenarios
hsim <- NULL                                # initialize hsim

for (i in 1:ns) {
    t <- xn*(x[i+1,]/x[i,])                 # Scenario i
    hsim <- rbind(hsim,t)                   # Append t to hsim
}

is.matrix(hsim)                             # is.matrix: test if it is a matrix
```

```
[1] TRUE
```

```r
dim(hsim)                                 # 988x3 matrix: 988 scenarios and 3 stocks
```

```
[1] 988   3
```

```r
ws                                        # Number of shares bought at day n
```

```
[1] 580.9731 327.3680 334.8962
```

```r
is.matrix(ws)
```

```
[1] FALSE
```

```r
(ws <- as.matrix(ws))                     # as.matrix: turn it into a matrix
```

```
         [,1]
[1,] 580.9731
[2,] 327.3680
[3,] 334.8962
```

```r
ps <- as.vector(hsim%*%ws)                # Compute simulated portfolio value

loss <- p0-ps                             # Compute loss
(VaRs <- quantile(loss, 0.99))            # 1-day 99% VaR
```

```
     99%
2181.303
```

Note that the cost of the portfolio is \$100,000 on 31 Aug 2018. The 1-day 99% VaR, obtained from the 99th percentile of loss distribution, is \$2,181.

Normal Model

Let u_1, u_2 and u_3 be daily returns of the stock HSBC, CLP and Cheung Kong respectively. Assume that $u = (u_1, u_2, u_3)'$ follows a trivariate normal distribution with mean zero (please refer to Tutorial 3 for more details). Mathematically speaking, let

$$u = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \sim N_3(0, \Sigma),$$

where Σ is the population covariance matrix of u.

The change in the portfolio value is

$$\Delta P = w'u = w_1 u_1 + w_2 u_2 + w_3 u_3$$

with mean

$$E(\Delta P) = w'E(u) = 0$$

and estimated variance

$$\widehat{\text{Var}}(\Delta P) = w'Sw,$$

where the sample covariance matrix S is used to replace the unknown Σ in $\text{Var}(\Delta P) = w'\Sigma w$.

The 1-day 99% VaR is given by $z_{0.99} \times \sqrt{w'Sw}$, where $z_{0.99} \approx 2.326$ is the 99th percentile of standard normal distribution.

```r
# as.ts: coerce an object to a time-series
t1 <- as.ts(d$HSBC)                       # For stock HSBC (0005)
t2 <- as.ts(d$CLP)                        # For stock CLP (0002)
t3 <- as.ts(d$CK)                         # For stock Cheung Kong (0001)
```

```r
# Compute daily percentage return
u1 <- (lag(t1)-t1)/t1                   # lag: compute a lagged version of a time series
u2 <- (lag(t2)-t2)/t2
u3 <- (lag(t3)-t3)/t3

u <- cbind(u1, u2, u3)                  # Combine into matrix u
head(u)
```

```
               u1           u2           u3
[1,] -0.004784891  0.007451304 -0.001412813
[2,]  0.005408589  0.014339632  0.028289756
[3,] -0.004183978 -0.005952363 -0.013750993
[4,]  0.003601586 -0.005988077 -0.011854537
[5,] -0.007775262 -0.007530245 -0.002824133
[6,] -0.008438693 -0.007587165 -0.019111672
```

```r
# Vectorization: faster approach than for-loop
# Each row of u is returns on a day and w is a fixed vector
# After matrix multiplication, each element of dp is
# a dot product between w and u
dp <- u%*%w                             # Delta P = w'u
dim(dp)                                 # u is 988x3 and w is 3x1
```

```
[1] 988   1
```

```r
(sdp <- apply(dp, 2, sd))               # Standard deviation of portfolio
```

```
[1] 855.5316
```

```r
# Alternatively
S <- var(u)                             # Sample covariance matrix
sqrt(w%*%S%*%w)
```

```
         [,1]
[1,] 855.5316
```

```r
(VaRn <- qnorm(0.99)*sdp)               # 1-day 99% VaR
```

```
[1] 1990.264
```

Note that the 1-day 99% VaR using normal model is `VaRn` (\$1,990) which is less than `VaRs` (\$2,181). The normality assumption is questionable since returns have a fatter tail than that of the normal distribution. Hence the `VaRn` is over-optimistic.

t Model

Alternatively we can model the changes in the portfolio value by Student's t-distribution. Please refer to Tutorial 2 for more details.

```r
ku <- sum((dp/sdp)^4)/nrow(dp) - 3      # Sample excess kurtosis
(v <- round(6/ku + 4))                  # Estimate df, round to the nearest integer
```

```
[1] 6
```

```r
(VaRt <- qt(0.99, v)*sdp)               # 1-day 99% VaR
```

```
[1] 2688.652
```

Note that the 1-day 99% VaR using t model is `VaRt` (\$2,689) which is greater than `VaRn` (\$1,990).

Backtesting

Let 1-day $X\%$ VaR be $\$V$. An exception occurs if the loss is greater than $\$V$ on a given day. If the VaR model is accurate, the probability that the loss is greater than $\$V$ on any given day is $p = 1 - X$. Suppose the total number of days is n. Using Binomial(n, p), the expected number of exceptions is given by $np = n(1 - X)$.

```r
n <- nrow(d)-1                          # Number of observation of u
n1 <- n-250+1                           # Starting index for 250 days before n
x <- as.matrix(d[n1:n,])                # Select the most recent 250 days
ps <- as.vector(x%*%ws)                 # Compute portfolio values
ps <- c(ps, sum(w))                     # Add total amount at the end
loss <- ps[1:250] - ps[2:251]           # Compute the daily losses

(expected_exc <- 250*(1-0.99))          # Expected number of exceptions
```

```
[1] 2.5
```

```r
# Count the number of exceptions in each VaR model
sum(loss>VaRs)                          # Historical Simulation
```

```
[1] 2
```

```r
sum(loss>VaRn)                          # Normal Model
```

```
[1] 2
```

```r
sum(loss>VaRt)                          # t Model
```

```
[1] 1
```

From the output, the number of exceptions in the past 250 days for all models are small.