Univariate Value at Risk and Expected shortfall

Code

CW7

14 November, 2022

Univariate Value at Risk and Expected shortfall

The VaR of a portfolio measures the value in £ which an investor would lose with some probability (1% or 5%), over a specified horizon. Because VaR represent a loss, it is usually a positive number.

Univariate Value at Risk and Expected shortfall

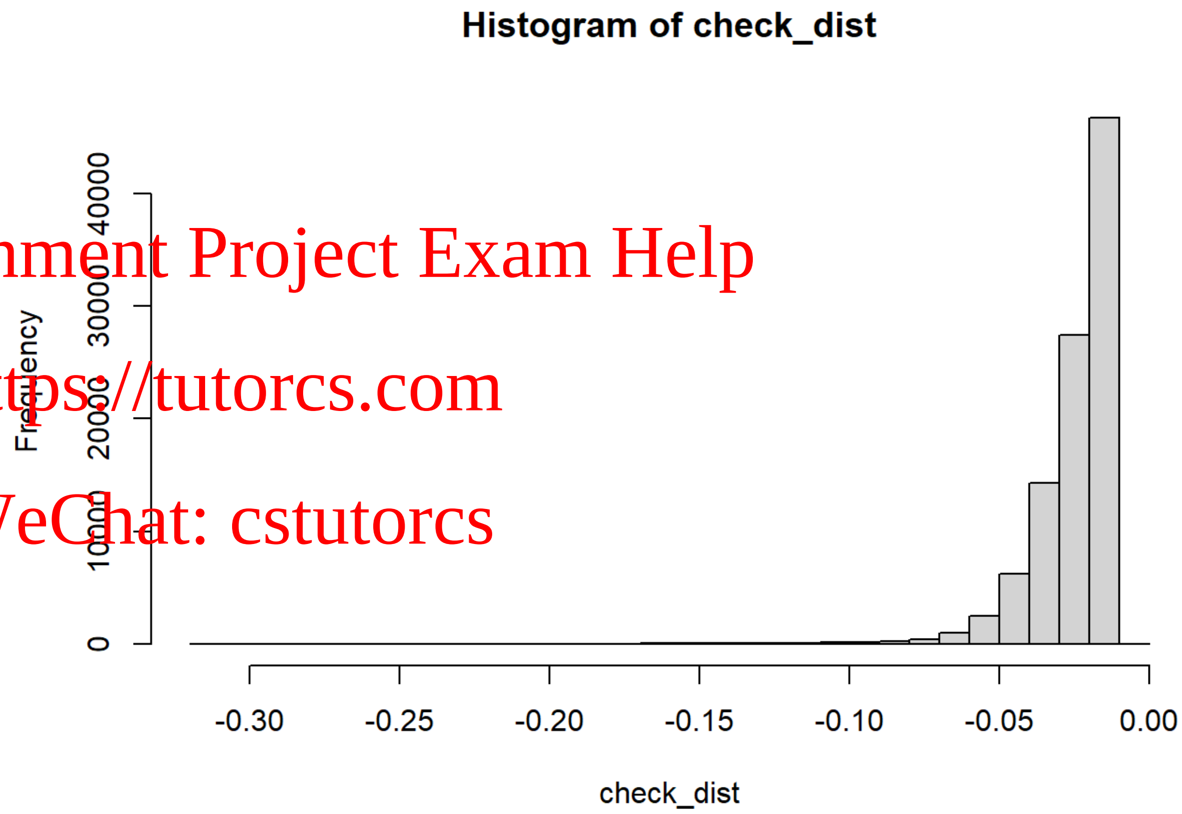
In-sample parametric VaR and ES estimates - Normal

distribution

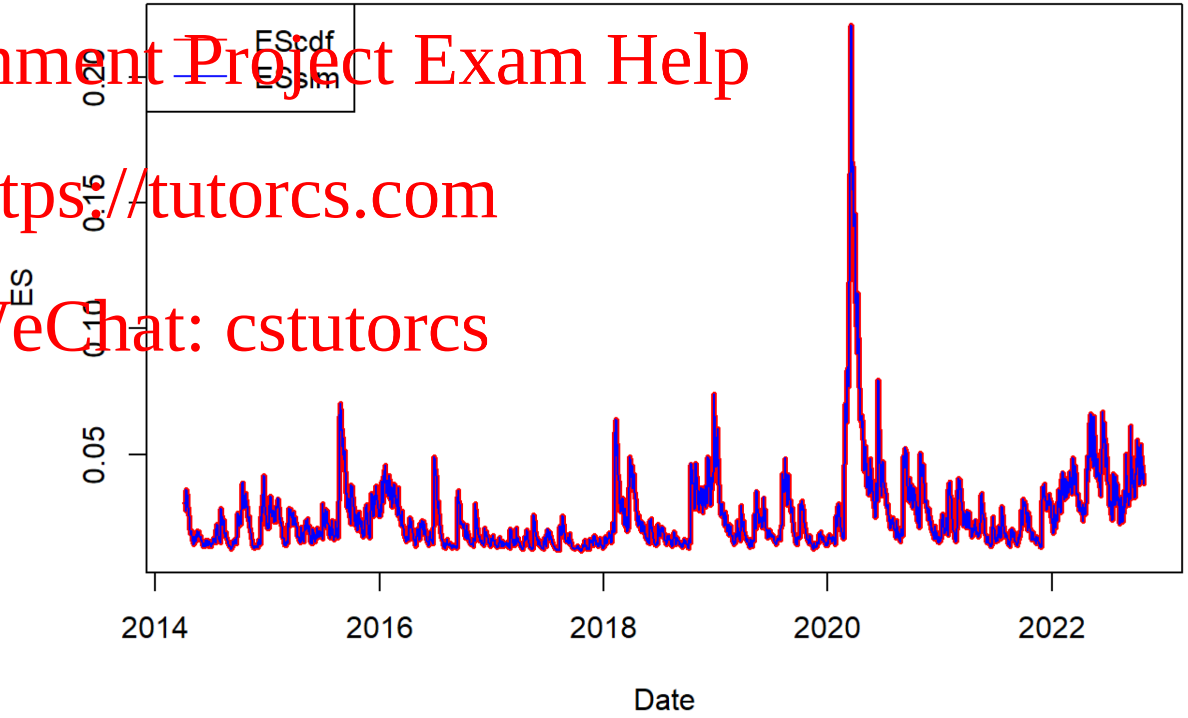
```
# Matrix pre-allocation
sigma <- xts(order.by = index(log_returns_demean))
VaRt <- xts(matrix(nrow = length(index(log_returns_demean)), ncol =
        2), order.by = index(log_returns_demean))
ESt <- xts(matrix(nrow = length(index(log_returns_demean)), ncol =
        2), order.by = index(log_returns_demean))

# Univariate GARCH to estimate time varying conditional volatility
        - Normal distribution
GARCH_1_1 <- ugarchspec(variance.model = list(model = 'sGARCH', gar
        chOrder = c(1,1)),
        mean.model = list(armaOrder = c(0, 0), incl
        ude.mean = FALSE))
GARCH_1_1_fit <- ugarchfit(spec = GARCH_1_1, data = log_returns_dem
        ean, solver = 'hybrid')

sigma <- GARCH_1_1_fit@fit$sigma

# Normal VaR

VaRt[, 1] <- - qnorm(p1) * sigma
VaRt[, 2] <- - qnorm(p5) * sigma

# Normal ES
ESt[, 1] <- dnorm((- VaRt[, 1] / sigma)) * sigma / p1
ESt[, 2] <- dnorm((- VaRt[, 2] / sigma)) * sigma / p5

# Normal simulated ES
sim <- 100000 # Number of simulations
rd <- runif(n = sim, min = 0, max = 1) # generate random numbers un
        iformly distributed in [0,1]
```

Univariate Value at Risk and Expected shortfall

```
EStr <- xts(- mean(qnorm(p1 * rd)) * sigma ,order.by = index(log_re
        turns_demean))

check_dist <- qnorm(p1 * rd)* sigma # check generated distribution
hist(check_dist, breaks = 30)
```



```
# check simulated ES very close to analytical ES
plot(x = index(log_returns_demean), y = ESt[, 1], ylab = "ES", lwd
        = 3,
      xlab = "Date", type = "l", col = "red")
  lines(x = index(log_returns_demean), EStr, col = "blue", lwd = 1)
  legend("topleft", legend = c('EScdf', 'ESsim'), lty = 1, col = c
        ("red", "blue"))
```



In-sample parametric VaR and ES estimates - *t-Student distribution*

```
# Matrix pre-allocation
sigmas <- xts(order.by = index(log_returns_demean))
VaRts <- xts(matrix(nrow = length(index(log_returns_demean)), ncol = 2), order.by = index(log_returns_demean))
ESts <- xts(matrix(nrow = length(index(log_returns_demean)), ncol = 2), order.by = index(log_returns_demean))

# Univariate GARCH to estimate time varying conditional volatility - Normal distribution
GARCH_1_1_t <- ugarchspec(variance.model = list(model = 'sGARCH', garchOrder = c(1, 1)), distribution.model = 'std',
                          mean.model = list(armaOrder = c(0, 0), include.mean = FALSE))
GARCH_1_1_t_fit <- ugarchfit(spec = GARCH_1_1_t, data = log_returns_demean, solver = 'hybrid')

sigmas <- GARCH_1_1_t_fit@fit$sigma
df <- GARCH_1_1_t_fit@fit$coef["shape"]
VaRts[, 1] <- - qt(p = p1, df = df) * sigmas * sqrt((df - 2) / df)
VaRts[, 2] <- - qt(p = p5, df = df) * sigmas * sqrt((df - 2) / df)

# ES Simulated
sim <- 100000
rd <- runif(n = sim, min = 0, max = 1)
ESts[, 1] = - mean(qt(p1 * rd, df)) * sigmas * sqrt((df - 2) / df)
ESts[, 2] = - mean(qt(p5 * rd, df)) * sigmas * sqrt((df - 2) / df)
```


Historical simulation VaR and ES

Unconditional HS VaR

In order to calculate the unconditional VaR we need to find the lowest *p-th* observation in our sample:

1. p1 we lookup the lowest 22nd daily return - 0.01*2157
2. p2 we lookup the lowest 108th daily return - 0.05*2157

```
sortret <- xts(sort(coredata(log_returns_demean)), order.by = index
        (log_returns_demean))
- sortret[22]
```

```
##                   [,1]
## 2014-05-07 0.03459941
```



```
- sortret[108]
```



```
##                   [,1]
## 2014-09-09 0.01807208
```


The unconditional ES is the mean of the lowest observed returns, conditional on VaR violation

```
- mean(sortret[1:22])
```

```
## [1] 0.05037249
```

```
- mean(sortret[1:108])
```

Univariate Value at Risk and Expected shortfall

```
## [1] 0.02945622
```

Time varying VaR and ES

```
sigma_HS <- xts(order.by = index(log_returns_demean))
VaR_HS <- xts(matrix(nrow = length(index(log_returns_demean)), ncol
        = 2), order.by = index(log_returns_demean))
ES_HS <- xts(matrix(nrow = length(index(log_returns_demean)), ncol
        = 2), order.by = index(log_returns_demean))
p = c(p1, p5)
# WE size to get at least 3 violations
for (i in 1:2) {
  WE <- 3 / p[i]
  VaR_HS[, i] <- rollapply(data = log_returns_demean, width = WE, F
        UN = function(x) - sort(coredata(x))[3])
  ES_HS[, i] <- rollapply(data = log_returns_demean, width = WE, FU
        N = function(x) - mean(sort(coredata(x))[1:3]))
}

VaR_HS <- lag(VaR_HS, k = 1, na.pad = TRUE)
ES_HS <- lag(ES_HS, k = 1, na.pad = TRUE)

v <- array(dim = c(length(index(ESt)), 3, 2))
es <- array(dim = c(length(index(ESt)), 3, 2))

for (i in 1:2) {
  v[, ,i] <- cbind(VaRt[, i], VaRts[, i], VaR_HS[, i])
  es[, ,i] <- cbind(ESt[, i], ESts[, i], ES_HS[, i])
}
```

Univariate Value at Risk and Expected shortfall

Comparing 3 methods

```
par(mfrow=c(2,2))

for (i in 1:2) {
  plot(x = index(VaRt), y = v[, 1,i], main = paste("Probability Lev
        el:", p[i]), ylab = "VaR",
       xlab = "Date", type = "l", col = "red", lwd = 2)
  lines(x = index(VaRt), v[, 2,i], col = "blue", lwd = 1)
  lines(x = index(VaRt), v[, 3,i], col = "green", lwd = 2)
  legend("topleft", legend = c('VaRt', 'VaRts', 'VaR_HS'), lty = 1,
        col = c("red", "blue", "green"))

  plot(x = index(ESt), y = es[, 1,i], main = paste("Probability Lev
        el:", p[i]), ylab = "ES",
       xlab = "Date", type = "l", col = "red", lwd = 2)
  lines(x = index(ESt), es[, 2,i], col = "blue", lwd = 1)
  lines(x = index(ESt), es[, 3,i], col = "green", lwd = 2)
  legend("topleft", legend = c('ESt', 'ESts', 'ES_HS'), lty = 1, co
        l = c("red", "blue", "green"))
}
```

Univariate Value at Risk and Expected shortfall