# Empirical Finance: Assignment 4

**By Group 15**

| | |
|---|---|
| Lanlan Hou | 2801069 |
| Rick van der Ploeg | 2782774 |
| Sebastiaan van Dijen | 2769505 |
| Wa Chak Sou | 2796840 |

School of Business and Economics

**24th November 2025**

# QUESTION 1: Parametric Risk Measures

## a. AR(1)-GARCH(1,1) model estimation

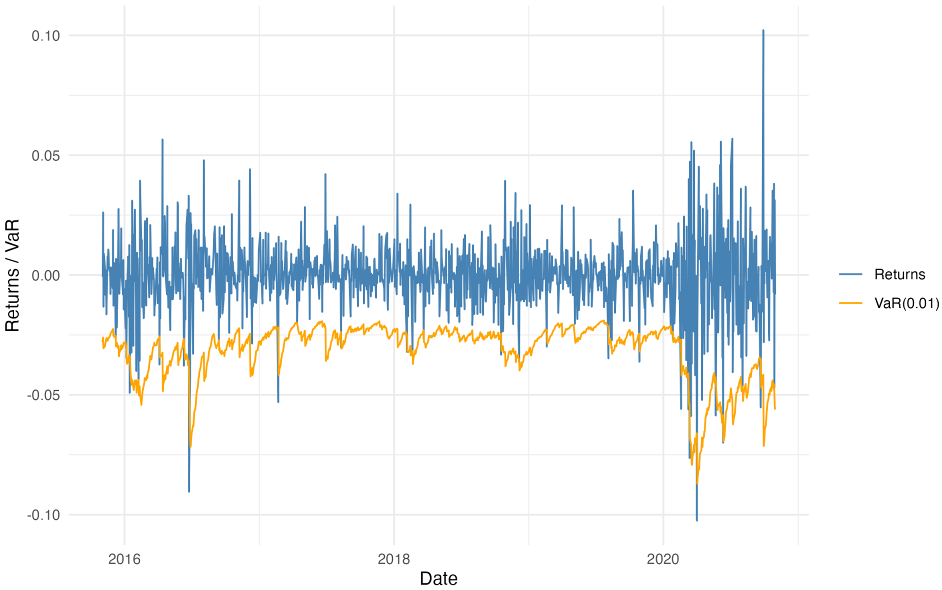


Figure 1.1.1 The forecast and realized simple returns for HSBC using the AR(1)-GARCH(1,1) model

The 1% parametric VaR obtained from the rolling AR(1)- GARCH(1, 1) model clearly responds to changes in market changes. With a 250-day refit window, the VaR series is relatively smooth in corresponded periods but also shows sharp downward movements with large absolute values, especially large losses. This pattern reflects volatility clustering. Overall, the model produces time-varying risk estimates that increase in periods with larger volatility and seems moderate when market is less changeable.

## b. Expected Shortfall (ES) and Value-at-Risk (VaR)

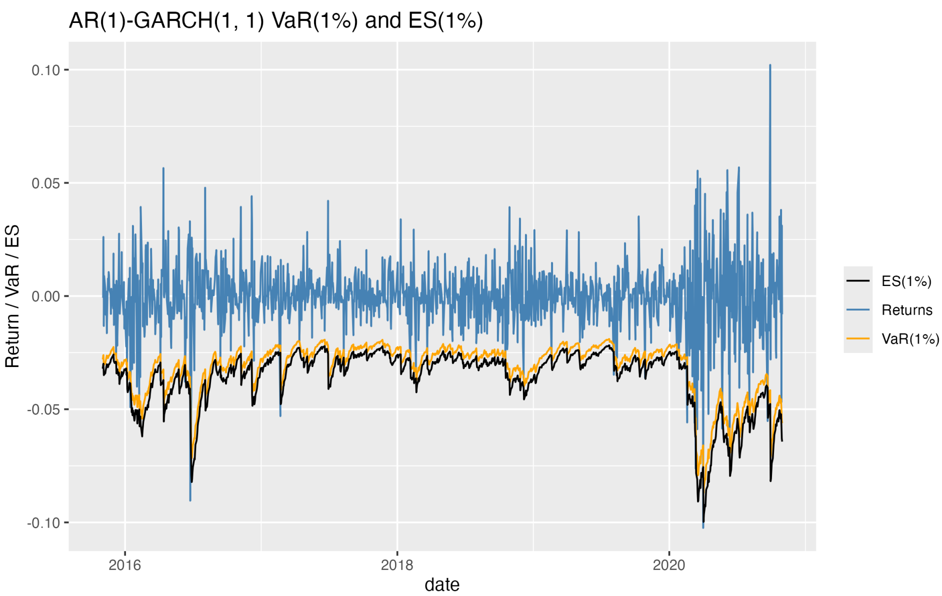


Figure 1.2.1 Realized simple returns, VaR and ES from the AR(1)-GARCH(1,1) model

The ES(1%) series lies right below the VaR(1%) series throught all periods, shows the fact that the excepted shortfall measures the average loss in the worst 1% of the outcomes, while VaR only shows the 1% quantile threshold. During the more calm periods, ES and VaR seems to be closer. While when the market has larger volatility, ES becomes more negative and the difference is larger. This shows that ES captures the worst situation of extreme losses, and VaR tends to captures the frequency. In general, ES provides a more conservative and informative extreme cases measure, especially when the mrket has larger volatility.

## Question 2: Nonparametric Risk Measures

### a. Nonparametric Value-at-Risk (VaR)

| **Start Date** | **VaR(0.01)** |
|---|---|
| 2020-04-01 | 0.05524580 |
| 2020-09-28 | 0.02731398 |
| 2017-06-05 | 0.03018712 |
| 2019-05-31 | 0.05610651 |

Table 2.1.1: The VaR at 1% level (Estimation Window = 300)

We use a 300-day rolling window, corresponding to the general rule of minimum $W = 3/p$. It is long enough to provide a stable empirical distribution and short enough to reflect the changes in the markets. We choose four dates from the second half of the sample: the day with the largest negative return, the day with the largest positive return, and two random dates from calm period. The historical simulation VaR is computed as the empirical 1% quantile of the previous daily returns of the window. All historical VaR(1%) estimates are negative because the 1% empirical quantile corresponds to the worst daily losses in the previous 300 days. During the volatile periods such as early 2020, VaR is around -5.5%, while in the calm period like 2019 is close to 2.7%. This reflects how historical simulation captures the movements of the market and show more conservative when there is extreme losses in the selected window.

### b. Nonparametric Expected Shortfall (ES)

| **Start Date** | **VaR(0.01)** | **ES(0.01)** |
|---|---|---|
| 2020-04-01 | 0.05524580 | 0.06128020 |
| 2020-09-28 | 0.02731398 | 0.03410098 |
| 2017-06-05 | 0.03018712 | 0.03196654 |
| 2019-05-31 | 0.05610651 | 0.06377331 |

Table 2.2.1: The comparison of VaR and ES at 1% level (Estimation Window = 300)

Using the same 300-day window, the historical ES(1%) is computed as the average return below 1% quantile. For all four dates, ES is more negative than VaR, reflecting the additional severity of losses once reaches the VaR threshold. The difference is larger on those dates in volatile dates and smaller during the calm periods. Overall, ES captures the extreme losses better than VaR.

# Appendix

```
# Load required packages
library(tidyquant)       # Financial data extraction + tidy returns
library(dplyr)           # Data manipulation
library(rugarch)         # GARCH estimation and forecasting
library(ggplot2)         # Visualization
library(PerformanceAnalytics)

# Fetch Data
symbol <- "HSBC"
start_date <- "2010-11-01"
end_date <- "2020-11-01"

# Download daily adjusted close prices
HSBC_data <- tq_get(symbol, get="stock.prices", from=start_date, to=end
    _date)

# Compute daily simple returns
HSBC_returns <- HSBC_data %>%
  tq_transmute(select = adjusted, mutate_fun = periodReturn, period="
      daily", type="arithmetic")

# Spilt sample into first 5 years and last 5 years
returns_first5y <- HSBC_returns %>% filter(date < "2015-11-01")
returns_last5y <- HSBC_returns %>% filter(date >= "2015-11-01")

# Specify AR(1)-GARCH(1, 1) with normal errors
spec <- ugarchspec(
  variance.model = list(model="sGARCH", garchOrder=c(1,1)),
  mean.model = list(armaOrder = c(1,0), include.mean = TRUE),
  distribution.model = "norm"
)

# Use full sample for rolling forecasts
ret_vec <- HSBC_returns$daily.returns

# Total sample
T <- length(ret_vec)

# First 5 years length
T1 <- nrow(returns_first5y)

# Rolling estimation for VaR firecasting
roll <- ugarchroll(
  spec = spec,
  data = ret_vec,
  forecast.length = nrow(returns_last5y),
  refit.every = 250,   # refit window as one trading year
  refit.window = "recursive",
  calculate.VaR = TRUE,
  VaR.alpha = 0.01,
  solver = "hybrid"
)

# Extract computed VaR
VaR_df <- as.data.frame(roll@forecast$VaR)
VaR_001 <- VaR_df[,"alpha(1%)"]
```

````

# Attach to last-5-year data
returns_last5y$VaR_001 <- VaR_001

# Plot VaR and returns
ggplot(returns_last5y, aes(x = date)) +
  geom_line(aes(y = daily.returns, color = "Returns")) +
  geom_line(aes(y = VaR_001, color = "VaR(0.01)")) +
  labs(y = "Returns / VaR", x = "Date") +
  scale_color_manual("", values = c("Returns" = "steelblue", "VaR(0.01)" = "orange")) +
  theme_minimal()

ggsave("~/Downloads/Returns vs. VaR.png", width = 8, height = 5, dpi = 300)
```
````

Figure 1: Question 1(a)

````

```{r Exercise1b, echo=TRUE}

Extract conditional mean and volatility from GARCH forecasts
density_df <- as.data.frame(roll@forecast$density)

#colnames(density_df) should be capital letters
mu <- density_df$Mu
sigma <- density_df$Sigma

# Parametric ES formula with normaility
ES_001 <- mu - sigma * (dnorm(qnorm(0.01)) / 0.01)

returns_last5y$ES_001 <- ES_001

# Plot returns, VaR and ES
ggplot(returns_last5y, aes(x=date)) +
          geom_line(aes(y=daily.returns, color = "Returns")) +
        geom_line(aes(y=VaR_001, color = "VaR(1%)")) +
          geom_line(aes(y=ES_001, color = "ES(1%)")) +
          scale_color_manual("", values=c("Returns"="steelblue",
                                          "VaR(1%)" = "orange",
                                          "ES(1%)" = "black")) +
          labs(y="Return / VaR / ES", x="date", title = "AR(1)-GARCH(1, 1) VaR(1%) and ES(1%)")
theme_minimal()

ggsave("~/Downloads/1b_AR(1)-GARCH(1, 1) VaR(1%) and ES(1%).png", width = 8, height = 5, dpi = 300)
```
````

Figure 2: Question 1(b)

```
raw <- read.xlsx('student_groups_stocks.xlsx', sheet = 1)

groupNumber <- 15
nameOfStock <- raw$Stock.Name[groupNumber]
startDate <- raw$Start.Date[groupNumber]
endDate <- raw$`End.Date.(+10y)`[groupNumber]

```
```

```
# compute the start date of the final five years
originalDate <- as.Date(startDate)
endDate_fiveYears <- originalDate %m+% years(5)

# extract the daily stock price from source
getSymbols(nameOfStock, src = 'yahoo', from = startDate, to = endDate)

# extract the data of first five years
firstFiveYear <- window(HSBC, start = startDate, end = endDate_
    fiveYears)

# extract the Adjusted Close for the first five years
adjustedPrice_firstFiveYears<- as.numeric(firstFiveYear[,'HSBC.Adjusted
    '])

compute_simpleReturns <- function(input_priceSeries){
  result <- numeric(length(input_priceSeries) - 1)

  for(i in (2:length(input_priceSeries))){
    result[i-1] <- (input_priceSeries[i] - input_priceSeries[i-1]) /
                      input_priceSeries[i-1]
  }

  return(result)
}

# compute the Daily Simple Returns for the first five years
simpleReturns_firstFiveYears <- compute_simpleReturns(adjustedPrice_
    firstFiveYears)

# extract the data of last five years
lastFiveYear <- window(HSBC, start = endDate_fiveYears, end = endDate)
date_lastFiveYear <- index(lastFiveYear)

# extract the Adjusted Close for the next five years
adjustedPrice_lastFiveYears<- as.numeric(lastFiveYear[,'HSBC.Adjusted'
    ])

# compute the Daily Simple Returns for the next five years
simpleReturns_lastFiveYears <- compute_simpleReturns(adjustedPrice_
    lastFiveYears)

probability <- 0.01
window <- 3/probability

#choose the min and max date and two random ones
firstDate <- returns_last5y$date[which.min(returns_last5y$daily.returns
    )]
secondDate <- returns_last5y$date[which.max(returns_last5y$daily.
    returns)]
thirdDate <- returns_last5y$date[400]
fourthDate <- returns_last5y$date[900]

valueAtRisk_NonParametric <- function(input_startDate, input_window,
    input_probability){
  this_endDate <- as.Date(input_startDate) %m+% days(input_window)
  duration <- window(HSBC, start = input_startDate, end = this_endDate)

```

```
  # extract the Adjusted Close for the next five years
  this_adjustedPrice <- as.numeric(duration[,'HSBC.Adjusted'])

  this_simpleReturns <- compute_simpleReturns(this_adjustedPrice)
  sorted_this_simpleReturns <- sort(this_simpleReturns)

  this_VaR <- -sorted_this_simpleReturns[input_window*input_probability]

  return(this_VaR)
}

# compute the VaR using Non-Parametric Approach
first_VaR <- valueAtRisk_NonParametric(firstDate, window, probability)
second_VaR <- valueAtRisk_NonParametric(secondDate, window, probability)
third_VaR <- valueAtRisk_NonParametric(thirdDate, window, probability)
fourth_VaR <- valueAtRisk_NonParametric(fourthDate, window, probability)

nonParametric_VaR_df = data.frame(Date = c(firstDate,
                                           secondDate,
                                           thirdDate,
                                           fourthDate),
                                  VaR_0.01 = c(first_VaR,
                                               second_VaR,
                                               third_VaR,
                                               fourth_VaR))
nonParametric_VaR_df
```

Figure 3: Question 2(a)

```
expectedShortfall_NonParametric <- function(input_startDate, input_window, input_probability){
  this_endDate <- as.Date(input_startDate) %m+% days(input_window)
  duration <- window(HSBC, start = input_startDate, end = this_endDate)

  # extract the Adjusted Close for the duration
  this_adjustedPrice <- as.numeric(duration[,'HSBC.Adjusted'])

  this_simpleReturns <- compute_simpleReturns(this_adjustedPrice)
  sorted_this_simpleReturns <- sort(this_simpleReturns)

  this_VaR <- -sorted_this_simpleReturns[input_window*input_probability]
  this_ES <- -mean(sorted_this_simpleReturns[1:floor(input_window*input_probability)])

  return(this_ES)
}

# compute the ES using Non-Parametric Approach
first_ES <- expectedShortfall_NonParametric(firstDate, window, probability)
second_ES <- expectedShortfall_NonParametric(secondDate, window, probability)
third_ES <- expectedShortfall_NonParametric(thirdDate, window, probability)
```

```
fourth_ES <- expectedShortfall_NonParametric(fourthDate, window, probability)

nonParametric_risk_df = data.frame(Date = c(firstDate,
                                            secondDate,
                                            thirdDate,
                                            fourthDate),
                                   VaR_0.01 = c(first_VaR,
                                                second_VaR,
                                                third_VaR,
                                                fourth_VaR),
                                   ES_0.01 = c(first_ES,
                                               second_ES,
                                               third_ES,
                                               fourth_ES))
nonParametric_risk_df
```

Figure 4: Question 2(b)