Financial Risk Forecasting
Seminar Week 8: Implementing Risk Forecasting

Jon Danielsson
London School of Economics

Version 4.0

8 Implementing Risk Forecasting

8.1 Why risk forecasting matters

Risk forecasting is the practical application of the volatility models we have studied in previous weeks. While GARCH and DCC models help us understand how volatility and correlations evolve over time, risk forecasting translates these insights into concrete risk measures that financial institutions use daily.

The two main risk measures we focus on are VaR and ES.

These measures are used for: - Regulatory capital calculations (Basel requirements) - Internal risk limits and position sizing - Portfolio optimisation and risk budgeting - Stress testing and scenario analysis

The forecasting methods we implement (HS, EWMA, GARCH) represent different approaches to estimating the distribution of future returns, each with distinct advantages depending on market conditions and data availability.

For more detail, see VaR in the R notebook.

Our focus in this session is forecasting risk one day into the future, building on the volatility models from previous weeks.

Make sure that you are in the same folder as the data you created in Seminar 2, as we will use the `Returns.RData` file that we created then.

8.2 The plan for this week

2. Univariate estimate of ES and VaR using HS
3. Use a function to do HS
4. Multivariate estimate ES and VaR using HS
5. HS VaR with different estimation windows
6. EWMA VaR
7. GARCH VaR

8.3 Loading and setup

Load the returns data and select assets for analysis. We use JPMorgan Chase (JPM) as our primary example, representing the financial sector. Make sure that you are in the same folder as the data you created in Seminar 2, as we will use the `Returns.RData` file that we created then.

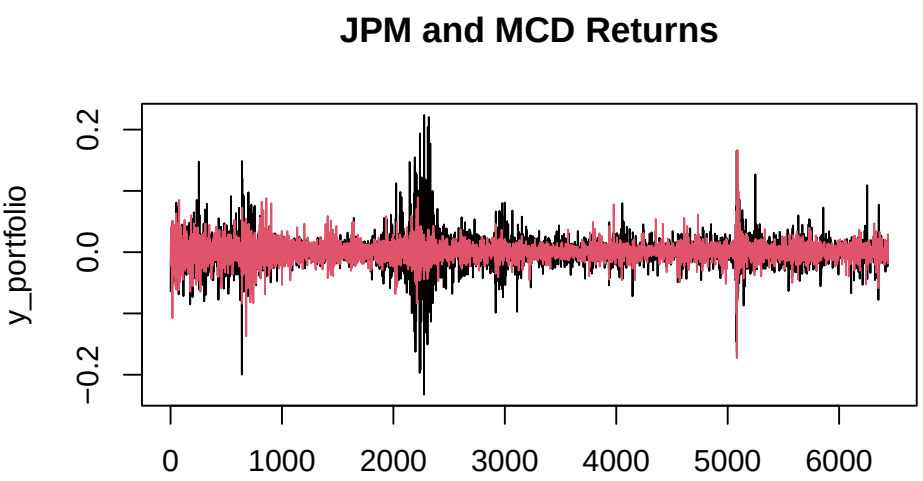
8.4 Loading data and libraries

```
library(rugarch)
```

```
load('Returns.RData')

# Focus on JPM for univariate analysis
y_single = Returns$JPM

# For portfolio analysis, we'll combine JPM with McDonald's (different sector)
y_portfolio = Returns[,c("JPM","MCD")]
matplot(y_portfolio, type='l', lty=1, main="JPM and MCD Returns")
```



Set the probability and portfolio value with explanations:

```
# Probability level: 0.02 = 2% = 98th percentile (1-in-50 day extreme event)
p=0.02
# Portfolio value: $1000 for illustration (would be actual portfolio value in practice)
value=1000
```

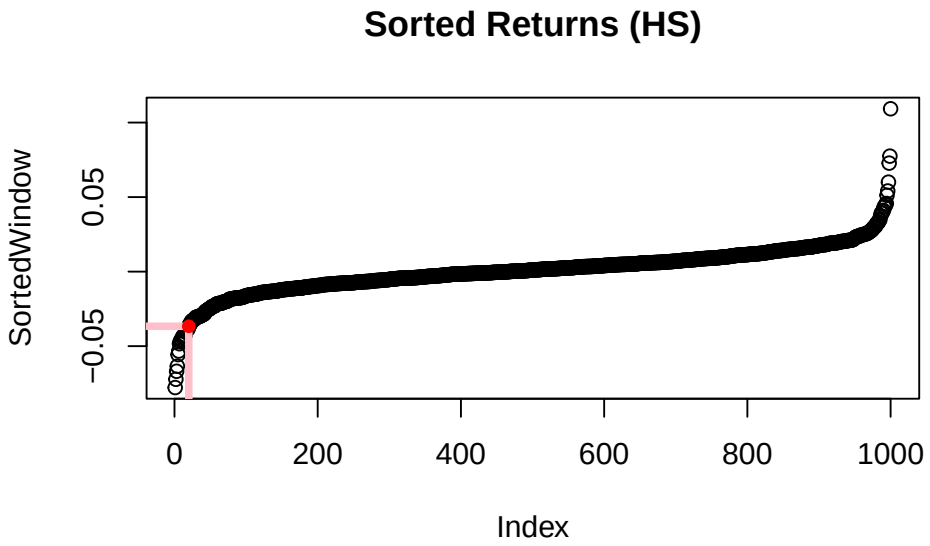
8.5 HS

Decide on the estimation window:

```
# Estimation window: 1000 days   4 years of trading days
# Common choice balancing sufficient data with model stability
We=1000
```

Plot the tails and show the VaR using our estimation window:

```
# Use the most recent We observations for estimation
window = tail(y_single, We)
SortedWindow = sort(window)
plot(SortedWindow, main="Sorted Returns (HS)")
VaR_point = SortedWindow[p*We]
segments(p*We, -100, p*We, VaR_point, col='pink', lwd=4)
segments(p*We, VaR_point, -100, VaR_point, col='pink', lwd=4)
points(p*We, VaR_point, col='red', pch=16)
```



Calculate the VaR and ES

```
VaR=-value*SortedWindow[p*We]
ES=-value*mean(SortedWindow[1:(p*We)])
VaR
```

```
[1] 36.68284
```

```
ES
```

```
[1] 49.26674
```

```
VaR=round(VaR,1)
ES=round(ES,1)
VaR
```

```
[1] 36.7
```

```
ES
```

```
[1] 49.3
```

Print out the combined results and put them into a string.

```
cat("The probability is ",100*p,"%",
  " and the portfolio value is $",
  value,". ", sep="")
```

```
The probability is 2% and the portfolio value is $1000.
```

```
cat("In this case, the VaR=$",VaR,
  ", while the ES is $",
  ES,"\n",sep="")
```

```
In this case, the VaR=$36.7, while the ES is $49.3
```

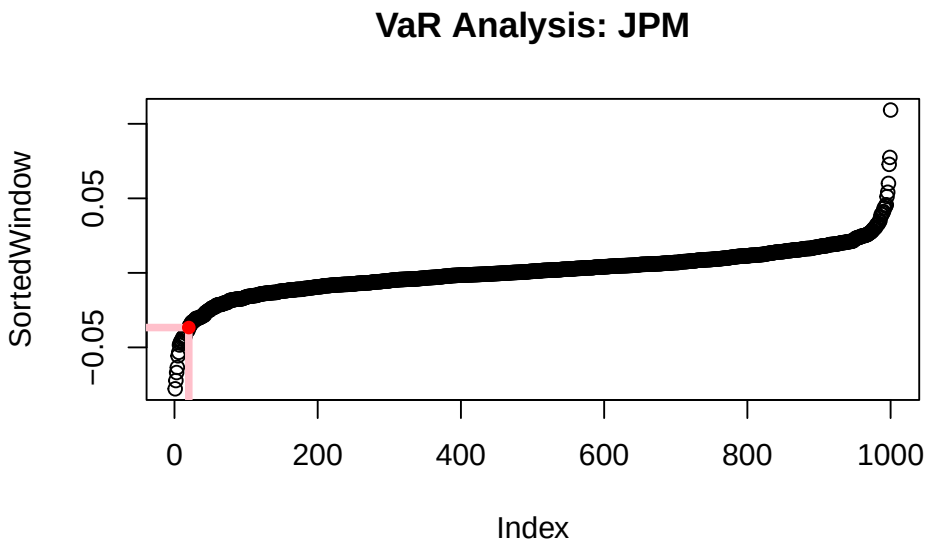
```
string=paste0("The probability is ",
  100*p,"%",
  " and the portfolio value is $",value,". ")
string=paste0(string,
  "In this case, the VaR=$",
  VaR,", while the ES is $",ES)
cat(string)
```

```
The probability is 2% and the portfolio value is $1000. In this case, the VaR=$36.7, while
```

8.6 Use a function to do HS

Having implemented the basic HS approach, we can make it reusable for different assets and parameters. This is important in practice where risk managers need to calculate VaR for many assets with consistent methodology:

```
RunVaR = function(Returns, Asset, p, We, value){
  window = tail(Returns[[Asset]], We)
  SortedWindow = sort(window)
  plot(SortedWindow, main=paste("VaR Analysis:", Asset))
  VaR_point = SortedWindow[p*We]
  segments(p*We, -100, p*We, VaR_point, col='pink', lwd=4)
  segments(p*We, VaR_point, -100, VaR_point, col='pink', lwd=4)
  points(p*We, VaR_point, col='red', pch=16)
  VaR = -value * SortedWindow[p*We]
  ES = -value * mean(SortedWindow[1:(p*We)])
  VaR = round(VaR, 1)
  ES = round(ES, 1)
  string = paste0("The probability is ", 100*p, "% ",
    "and the portfolio value is $", value, ". ")
  string = paste0(string, "In this case, the ", Asset,
     " VaR=$", VaR, ", while the ES is $", ES)
  return(list(VaR=VaR, ES=ES, summary=string))
}
# Test the function
RunVaR(Returns=Returns, Asset="JPM", p=p, We=We, value=value)
```




```
$VaR
[1] 36.7

$ES
[1] 49.3

$summary
[1] "The probability is 2% and the portfolio value is $1000. In this case, the JPM VaR=$36
```

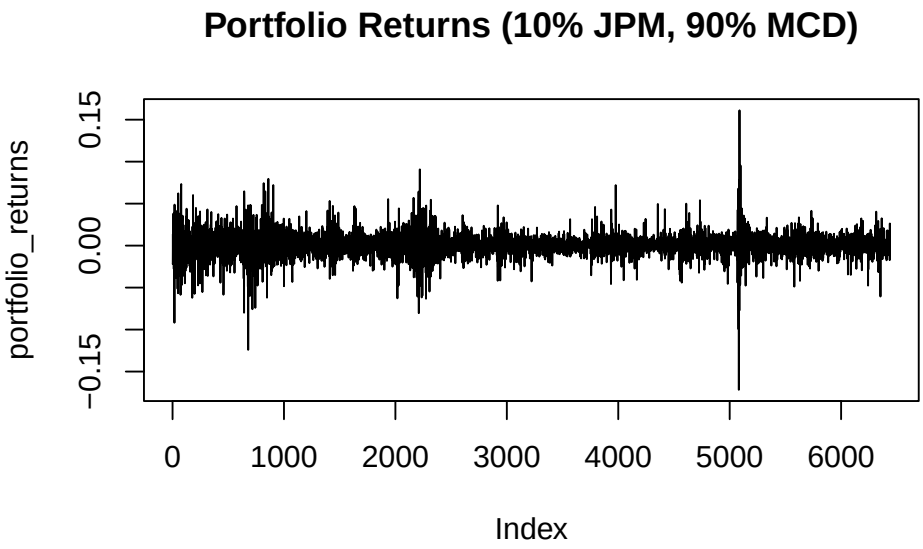
You can modify the function to optionally make the plot or save it to disk. In practice, you might save plots in multiple formats, such as png, svg, pdf and eps.

8.7 Multivariate HS VaR and ES

Moving from single assets to portfolios is fundamental in risk management, as most financial institutions hold diversified positions. Portfolio VaR cannot simply be added from individual asset VaRs due to correlation effects. We demonstrate this by creating a simple two-asset portfolio:

```r
# Portfolio weights: 10% financials, 90% consumer discretionary
portfolio_weights = c(0.1, 0.9)
portfolio_assets = c("JPM", "MCD")

# Calculate portfolio returns using matrix multiplication
portfolio_returns = as.matrix(Returns[,portfolio_assets]) %*% portfolio_weights
plot(portfolio_returns, type='l', main="Portfolio Returns (10% JPM, 90% MCD)")
```



We can then proceed as we did with single asset risk, but now the portfolio returns incorporate both individual asset volatilities and their correlation structure.

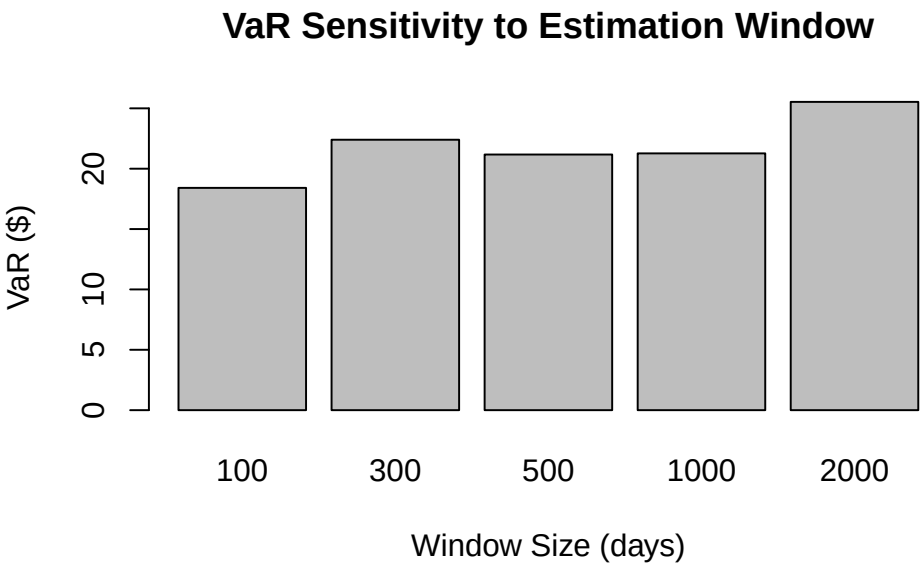
8.8 HS VaR with different estimation windows

A key decision in risk management is choosing the estimation window. Longer windows provide more data but may include outdated market regimes, while shorter windows are more responsive but potentially less stable. We examine this trade-off:

```
# Different estimation windows to test sensitivity
windows = c(100, 300, 500, 1000, 2000)
VaR_results = rep(NA, length(windows))

for(i in 1:length(windows)){
  window_data = tail(portfolio_returns, windows[i])
  VaR_results[i] = -value * sort(window_data)[p * windows[i]]
}

barplot(VaR_results, names.arg=windows,
        main="VaR Sensitivity to Estimation Window",
        xlab="Window Size (days)", ylab="VaR ($)")
```



8.9 EWMA VaR

While HS uses equal weights for all observations, EWMA recognizes that recent observations may be more informative about future volatility. This connects to our Week 7 multivariate EWMA but focuses on univariate forecasting:

- Lambda ($\lambda = 0.94$): Industry standard giving 94% weight to previous variance and 6% to new observation. Higher λ means slower adaptation to volatility changes.
- Burn-in period (30 observations): Number of iterations needed for the EWMA estimate to converge from the initial sample variance to the true EWMA value.

The EWMA variance update formula is: $\sigma_t^2 = \lambda\sigma_{t-1}^2 + (1-\lambda)r_{t-1}^2$

```
# Lambda: 0.94 is industry standard for daily data
lambda = 0.94
# Burn-in: iterations needed for convergence from initial variance
Burn = 30
y_ewma = Returns$JPM
s2 = var(y_ewma)  # Start with sample variance
N = length(y_ewma)

# Burn-in loop: iteratively update variance estimate
# Working backwards from most recent observation
for(i in 1:Burn){
  s2 = lambda*s2 + (1-lambda)*y_ewma[N-i]^2
}
VaR_ewma = -value*qnorm(p, sd=sqrt(s2))
VaR_ewma
```

```
[1] 30.19922
```

8.10 GARCH VaR

GARCH models, covered in Week 6, provide the most sophisticated approach by modelling volatility clustering patterns. Unlike HS (which assumes past returns represent future distribution) or EWMA (which weights recent observations more), GARCH explicitly models how volatility evolves over time:

```
# Use the same estimation window as HS for comparison
y_garch = tail(Returns$JPM, We)
spec = ugarchspec(
  mean.model = list(
    armaOrder=c(0,0),
    include.mean=FALSE)
)
fit = ugarchfit(spec=spec, data=y_garch, solver = "hybrid")
coef(fit)
```

```
       omega       alpha1        beta1
5.625329e-05 1.084434e-01 6.631790e-01
```

```
tail(y_garch, 1)[1]
```

```
[1] 0.006322759
```

```
tail(fit@fit$var, 1)
```

```
[1] 0.000220537
```

```
# Calculate one-step-ahead conditional variance
s2_forecast = coef(fit)[1] +
  coef(fit)[2] * tail(y_garch, 1)^2 +
  coef(fit)[3] * tail(fit@fit$var, 1)

VaR_garch = -value * qnorm(p, sd=sqrt(s2_forecast))
VaR_garch
```

```
[1] 29.53717
```

8.11 Comparing the three methods

Now we can compare the VaR estimates from all three approaches:

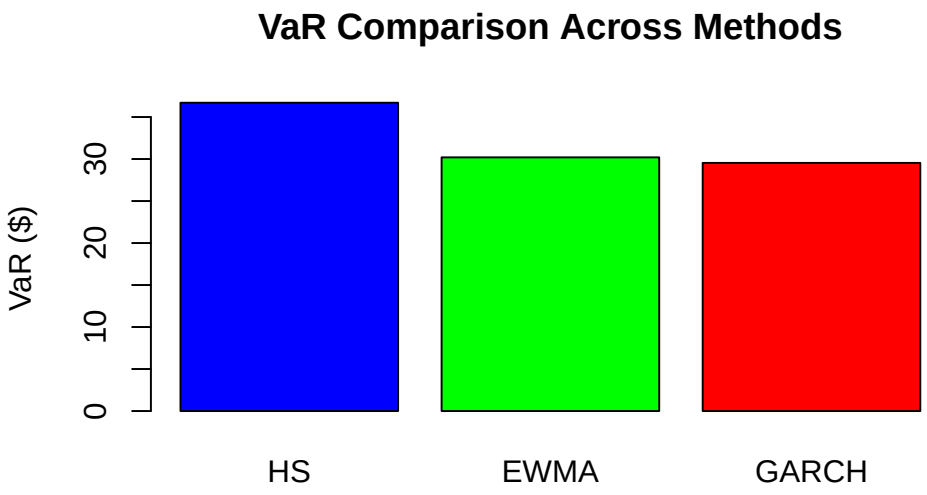
```
# Collect results from all methods
methods = c("HS", "EWMA", "GARCH")
VaR_estimates = c(VaR, VaR_ewma, VaR_garch)

# Create comparison table
comparison_df = data.frame(
  Method = methods,
  VaR_Estimate = round(VaR_estimates, 2)
)
print(comparison_df)
```

```
  Method VaR_Estimate
```

```
1     HS        36.70
2   EWMA        30.20
3  GARCH        29.54
```

```r
# Visual comparison
barplot(VaR_estimates, names.arg = methods,
        main = "VaR Comparison Across Methods",
        ylab = "VaR ($)", col = c("blue", "green", "red"))
```



8.12 Recap

8.12.1 In this seminar, we have covered:

- Understanding risk forecasting as the practical application of volatility modelling
- Implementing VaR (VaR) and ES (ES) using different methods
- HS (HS) for risk forecasting:
  - Using sorted returns to find quantiles
  - Calculating VaR at specified probability levels
  - Computing ES as the average of tail losses
  - Creating reusable functions for consistent risk calculations
- Portfolio risk analysis:
  - Combining multiple assets with weights
  - Matrix multiplication for portfolio returns
  - Understanding how correlations affect portfolio risk
- Comparing different estimation windows and their impact on VaR estimates
- EWMA for volatility forecasting:
  - Parameter selection ($\lambda = 0.94$) and burn-in periods
  - Connection to multivariate EWMA from Week 7
- GARCH-based VaR calculations:
  - Using conditional volatility forecasts from Week 6 models
  - One-step-ahead risk predictions
- Systematic comparison of all three methods:
  - Understanding when to use each approach

  - Interpreting differences in VaR estimates
- Visualizing risk measures with plots and annotations

8.12.2 Some new functions used:

- `segments()` — add line segments to plots
- `points()` — add points to existing plots

8.13 Optional exercises

1. Make a function that takes a vector of returns, estimation method and the necessary parameters as inputs and then writes image files with plots of the estimation results to the disk, which then can be used in other applications, such as PowerPoint or Word.
2. Make a function that takes as arguments a vector of returns, probability, λ and a vector of estimation window sizes and makes a data frame with three columns, (method, estimation window, VaR, ES), where the rows are all the possible combinations of the estimation method and window size, like HS(500), HS1000, etc.
3. Add an option to the function in the last exercise that optionally makes a barplot with the method on the x-axis and VaR on the y-axis.
4. Add an option to the function in the last exercise that optionally makes an informative printout of the results.
5. Add an option to the function in the last exercise that allows you to pick any number of the seven stocks in `Returns` along with a portfolio weight and uses that for analysis of portfolio risk.
6. Add an option to the function in the last exercise that allows you to use DCC for the risk forecasts.
7. Make an R Quarto file that does this analysis and then makes a complete report as a pdf or word file.