Financial Econometrics
ASSIGNMENT 4

0323642 - Passeri Francesca
0332884 - D'Alessandro Giorgia
0333001 - Saieva Lorenzo
0323474 - Sancamillo Tiziano

May 2023

1 Introduction

The following report focuses on the logarithmic return series of L'Oréal S.A. from January 1, 2005 to April 14, 2023 with the aim to understand if leverage plays an important role in forecasting volatility and Value at Risk.

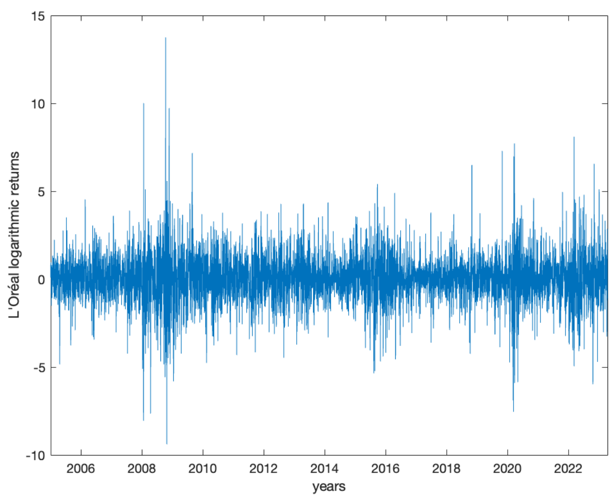

Figure 1: *log-returns of the five companies selected*

By means of a rolling forecast experiment, this work will provide a comparison between the one-day-ahead forecasts of 1 conditional $VaR(VaR_{t+1})$ and volatility $(\tilde{h}_{t+1|t})$ using 4 models:

- Gaussian GARCH(1,1)
- Gaussian EGARCH(1,1)
- Student's t-GARCH(1,1)
- Student's t-GJR-GARCH(1,1)

1.1 Evaluation of conditional VaR predictions

Conditional VaR can be estimated using two different approaches: the econometric approach and the RiskMetrics approach. The first one assumes that a conditional heteroscedasticity model (like those listed above) is specified and estimated for log returns (y_t). Then, the one-step-ahead predictive distribution $y_{t+1}|Y_t$ is estimated. This procedure can be applied both to the Gaussian and the Student-t case:

$$Gaussian - case : VaR = -(\mu_{t+1|t} + z_p\sqrt{h_{t+1}}) \tag{1}$$

$$t - Student - case : VaR = -(\mu_{t+1|t} + \frac{t_{v,p}}{\sqrt{v/(v-2)}}\sqrt{h_{t+1}}) \tag{2}$$

The second approach, the predictions RiskMetrics' exponential smoothing method with parameter $\lambda = 0.06$, will be used as a benchmark in the following analysis. In this case, the VaR is defined as:

$$VaR = -\sqrt{h_{t+1}}\phi^{-1}(p) \tag{3}$$

The evaluation of the VaR predictions will be based on the three following assessments: Backtesting, the total loss incurred and the Diebold-Mariano test.

Backtesting consists of the computation of the number of violations $y_{t+1} < -VaR_{t+1}$ over the m test observations and in the comparison between them and their expected number, 0.01m. Specifically, using a rolling window of size T = 3500 and starting from the 3501^{st} observation, the four models are estimated every 20 days and the corresponding one-step-ahead Value at Risk (VaR) and volatility predictions are computed by adding one future observation and deleting the initial one. The process is repeated until the end of the sample is reached so that each forecast is based on a training sample of T observations.

Otherwise, the VaR estimates can be assessed by the total loss incurred, where the loss function is the check function:

$$L(y_{t+1}, VaR_{t+1}) = (\alpha - \delta_{t+1}) * (y_{t+1} + VaR_{t+1}) \tag{4}$$

and

$$\delta_{t+1} = I(y_{t+1} \leq -VaR_{t+1}) \tag{5}$$

Of course, the lower the loss function, the higher the accuracy of the model. Finally, using the Diabold-Mariano test statistic it is possible to determine whether the forecasts obtained from a model are as accurate as those obtained from the benchmark (RiskMetrics). In particular, the DM test statistic is based on the sample mean loss differential of models i and j:

$$\bar{d} = \frac{1}{m} \sum_{t=T}^{T+m-1} [L(y_{t+1} + VaR_{i,t+1} - L(y_{t+1} + VaR_{j,t+1})]$$

(6)

1.2 Evaluation of volatility predictions

As regards the evaluation of the volatility predictions, the assessment criterion is the mean square forecast error:

$$MSFE = \frac{1}{m} \sum_{t=T}^{T+m-1} [y_{t+1}^2 - \tilde{h}_{t+1|t}]^2 \tag{7}$$

Where y_{t+1}^2 is the volatility proxy representing the forecast target.

2 One-day-ahead forecasts of 1% conditional VaR using the five models

Figure 2 superimposes the forecasts of the 1% conditional VaR in the five models over the yield series. Moreover, it shows the number of violations $y_{t+1} < -VaR_{t+1}$ for each case and it seems that VaR estimation performances do not differ significantly among the models.

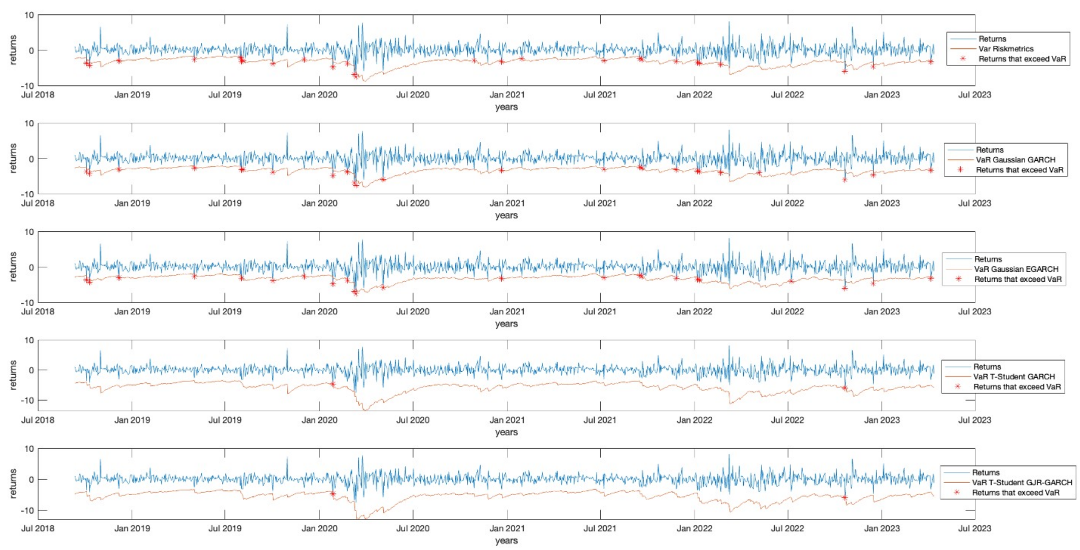

Figure 2: *Violations $y_{t+1} < -VaR_{t+1}$ for the 5 models defined above*

The following table summarises the results arising from the Backtesting approach:

MODEL	α	**EXP. VAR EXCEED**	**ACTUAL VAR EXCEED**	**VARIATION (%)**
GARCH(1,1)	1%	11.77	24	2.04%
EGARCH(1,1)	1%	11.77	23	1.95%
t-GARCH(1,1)	1%	11.77	15	1.27%
t-GJR-GARCH(1,1)	1%	11.77	15	1.27%
RISKMETRICS	1%	11.77	26	2.21%

Table 1: *results arising from the Backtesting approach*

In all cases the number of violations is greater than the expected one, especially for the Gaussian GARCH(1,1), the Gaussian EGARCH(1,1) and the RiskMetrics models. Therefore, in the referring period, in all cases, the losses faced were higher than the losses expected by the VaR. Since the Student-t models are those that are nearer to the referring nominal value (1%), according to the Backtesting approach they are the most precise and, hence, the most preferable ones.

As regards the total loss that occurred for the 5 models, these are shown in Table 2:

MODEL	**TOTAL LOSS OCCURRED**
GARCH(1,1)	61.1071
EGARCH(1,1)	60.9034
t-GARCH(1,1)	59.2339
t-GJR-GARCH(1,1)	57.3318
RISKMETRICS	63.7335

Table 2: *total loss that occurred for the 5 models*

Since the lower the loss function, the higher the accuracy of the model, according to the obtained results the t-GJR-GARCH(1,1) seems to be the best estimating method for the forecast of the VaR given that it minimises the loss function and the distance between the expected VaR exceed and the actual VaR exceed.

Finally, in Table 3 are reported the p-values resulting from the Diebold-Mariano test:

TEST	**P-VALUE**
GARCH(1,1) VS BENCHMARK	0.0140
EGARCH(1,1) VS BENCHMARK	0.0490
t-GARCH(1,1) VS BENCHMARK	0.0187
t-GJR-GARCH(1,1) VS BENCHMARK	0.0091

Table 3: *p-values resulting from the Diebold-Mariano test*

Since the p-values are all smaller than 0.05, it is possible to reject the null hypothesis that the considered models present the same forecast accuracy as the benchmark. This result is coherent with those of the previous two assessments since the four models always turn out to be better performing than the benchmark when forecasting the VaR.

3 One-day-ahead forecasts of volatility using the five models

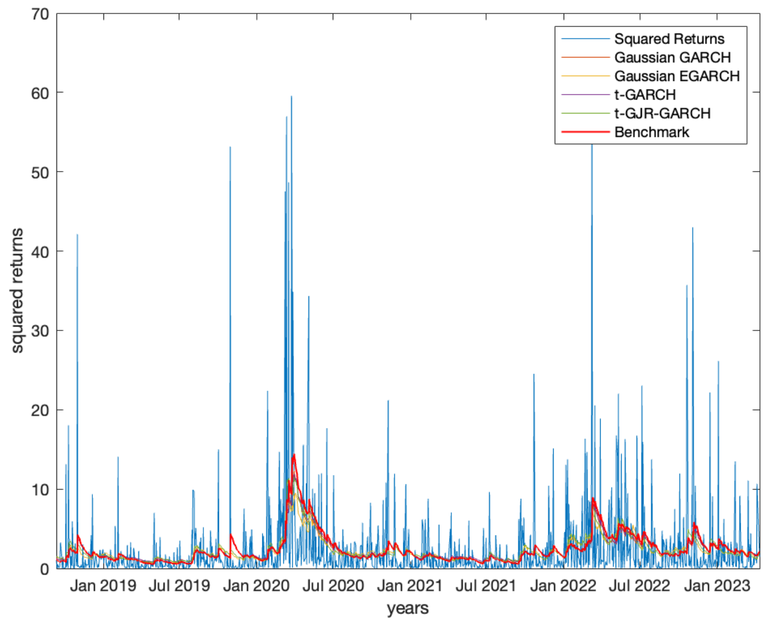

Figure 3: *comparison between the volatility forecasts estimated through the five models.*

In order to evaluate which model better performs in terms of the estimate of the volatility predictions, we compare the values of the Mean Square Forecast Error obtained for the five models:

MODEL	**MSFE**
GARCH(1,1)	31.0260
EGARCH(1,1)	30.2620
t-GARCH(1,1)	31.1105
t-GJR-GARCH(1,1)	29.9926
RISKMETRICS	31.2119

Table 4: *values of the MSFE obtained for the five models*

The MSFE criterion confirms that the best model is the Student-t-GJR(1,1) since it presents the minimum value of MSFE.

4 Conclusion

Based on the previous analysis, the t-GJR-GARCH(1,1) model consistently demonstrates favourable performance across multiple measures. It shows the lowest variation in variance exceedance, the lowest total loss value, the most significant difference in forecasting accuracy according to the Diebold-Mariano test and the lowest MSFE value, indicating better overall forecasting accuracy. Therefore, it looks like heavy tails and leverage are useful when predicting VaR and volatility. While the Gaussian EGARCH(1,1) and Gaussian GARCH(1,1) models generally perform slightly worse than the t-GJR-GARCH(1,1) model, they still exhibit reasonable accuracy and perform better than the benchmark (RiskMetrics model), which consistently shows higher variation and total losses, weaker differences in accuracy, and higher forecasting errors.

During periods of market stress such as the beginning of the Covid-19 pandemic and the invasion of Ukraine, the four models outperformed the benchmark. This is coherent with the results obtained so far from the assessment of conditional VaR and volatility forecasts. Graphically, the differences in the performance of the RiskMetrics with respect to the other models are underlined by higher peaks in volatility forecasts and lower peaks in VaR predictions. These conclusions are confirmed also by looking at the forecast error plot, where both in the Covid-19 pandemic period and in the early 2022 (Russia-Ukraine War) the forecast error of the benchmark is particularly higher than all the other models.

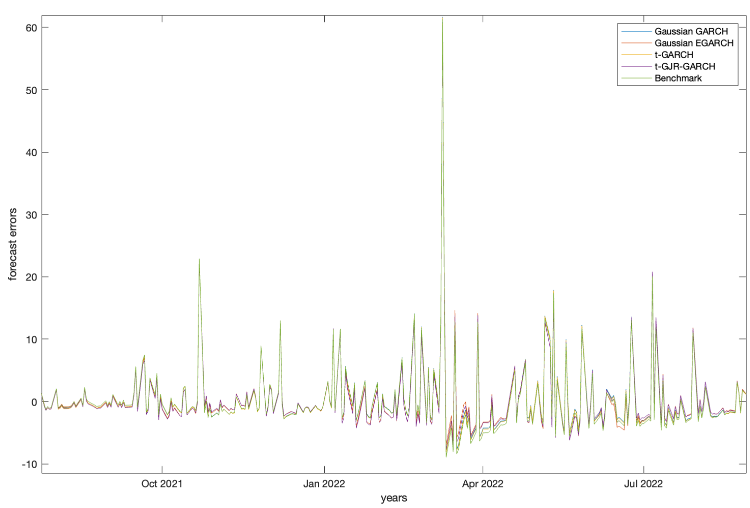

Figure 4: *comparison of the forecast errors of the five models.*

5 Appendix

```
clear all
close all
clc
%% EXERCISE 1
tab_NaN = readtable("OR.PA.csv");
tab = rmmissing(tab_NaN);

P = tab.Close; % close price
log_p = log(P);      % log prices
y = 100*diff(log_p); % logarithmic returns series
n = length(y);
dates = tab.Date(2:end);
vt = 1:n; % time index, t

% plot of returns
figure(1);
plot(dates,y);
ylabel("L'Or al logarithmic returns");
xlabel("years");

T = 3500; % size of the rolling window
m = n - T - 1; % observations

% Gaussian GARCH
Mdl1 = garch(1,1);
mPar1 = NaN(3,m);
vh_frcst1 = NaN(m,1);

% Gaussian EGARCH
Mdl2 = egarch(1,1);
mPar2 = NaN(4,m);
vh_frcst2 = NaN(m,1);

% Student-t GARCH
Mdl3 = garch(1,1);  Mdl3.Distribution = "t";
mPar3 = NaN(4,m);
vh_frcst3 = NaN(m,1);

% Student-t GJR
Mdl4  = gjr(1,1);    Mdl4.Distribution = "t";
mPar4 = NaN(5,m);
vh_frcst4 = NaN(m,1);
```

```
% Riskmetrics
MdlES =  arima('Constant', 0, 'D', 1,'MA',-.94, '
   Variance', 1);
vh_frcst_es = NaN(m,1);

vVar1 = []
vVar2 = []
vVar3 = []
vVar4 = []
vVarES = []
dp = 0.01
dt = norminv(0.01)
for i = 0:m-1;
    yr = y(i+1:T+i);         % rolling sample
    if any(i==0:20:m-1);
      [EstMdl1,~,~,~] = estimate(Mdl1, yr); % garch
         (1,1) gaussian
      [EstMdl2,~,~,~] = estimate(Mdl2, yr); % egarch
         gaussian
      [EstMdl3,~,~,~] = estimate(Mdl3, yr); % garch
         (1,1) t
      [EstMdl4,~,~,~] = estimate(Mdl4, yr); % gjr
         garch t
    end

    % Gaussian GARCH
    mPar1(:,i+1) = [EstMdl1.Constant; EstMdl1.ARCH
       {[1]}; EstMdl1.GARCH{[1]}];
    dh_frcst1    = forecast(EstMdl1, 1,'Y0',yr);
    vh_frcst1(i+1) = dh_frcst1; % forecast about
       volatility
    dmu1 = EstMdl1.Offset;
    vVar1 = - (dmu1 + dt*sqrt(vh_frcst1));

    % Gaussian EGARCH
    mPar2(:,i+1) = [EstMdl2.Constant; EstMdl2.ARCH
       {[1]}; EstMdl2.GARCH{[1]}; EstMdl2.Leverage
       {[1]}];
    dh_frcst2    = forecast(EstMdl2, 1,'Y0',yr);
    vh_frcst2(i+1) = dh_frcst2;
    dmu2 = EstMdl2.Offset;
    vVar2 = - (dmu2 + dt*sqrt(vh_frcst2));

    % Student-t GARCH
    mPar3(:,i+1) = [EstMdl3.Constant; EstMdl3.ARCH
       {[1]}; EstMdl3.GARCH{[1]}; EstMdl3.Distribution
```

```
        .DoF];
    dh_frcst3    = forecast(EstMdl3, 1,'Y0',yr);
    vh_frcst3(i+1) = dh_frcst3;
    dmu3 = EstMdl3.Offset;
    dnu3 = EstMdl3.Distribution.DoF;
    dt_3 = tinv(dp,dnu3)/sqrt(dnu3/(dnu3 - 2));
    vVar3 = - (dmu3 + dt_3*sqrt(vh_frcst3));

    % Student-t GJR
    mPar4(:,i+1) = [EstMdl4.Constant; EstMdl4.ARCH
        {[1]}; EstMdl4.GARCH{[1]}; EstMdl4.Leverage
        {[1]}; EstMdl4.Distribution.DoF];
    dh_frcst4    = forecast(EstMdl4, 1,'Y0',yr);
    vh_frcst4(i+1) = dh_frcst4;
    dmu4 = EstMdl4.Offset;
    dnu4 = EstMdl4.Distribution.DoF;
    dt_4 = tinv(dp,dnu4)/sqrt(dnu4/(dnu4 - 2));
    vVar4 = - (dmu4 + dt_4*sqrt(vh_frcst4));

    %Riskmetrics
    dh_frcstES     = forecast(MdlES, 1,'Y0',yr.^2);
    vh_frcst_es(i+1) = dh_frcstES; %forecast about
        volatility
    vVarES = -(sqrt(vh_frcst_es)*dt);
end
ve1   =  y(T+1:end-1).^2 - vh_frcst1;    % forecast
   error Gaussian GARCH
ve2   =  y(T+1:end-1).^2 - vh_frcst2;    % forecast
   error Gaussian EGARCH
ve3   =  y(T+1:end-1).^2 - vh_frcst3;    % forecast
   error Student-t GARCH
ve4   =  y(T+1:end-1).^2 - vh_frcst4;    % forecast
   error Student-t GJR
veES   =  y(T+1:end-1).^2 - vh_frcst_es; % forecast
   error Riskmetrics

dates_forecast = dates(T+1:end-1)
figure('Name', 'Comparison Volatility Predictions with
    Benchmark')
plot(dates_forecast,[y(T+1:end-1).^2, vh_frcst1,
   vh_frcst2, vh_frcst3, vh_frcst4]);
hold on
plot(dates_forecast,vh_frcst_es,'LineWidth',1,'color',
   'r')
xlabel('years')
ylabel('squared returns')
```

```
xlim([dates_forecast(1), dates_forecast(end)])
legend('Squared Returns','Gaussian GARCH', 'Gaussian
   EGARCH', 't-GARCH', 't-GJR-GARCH', 'Benchmark')
hold off

figure('Name','Comparison Forecast Errors')
plot(dates_forecast,[ve1,ve2, ve3, ve4, veES])
xlabel('years')
ylabel('forecast errors')
xlim([dates_forecast(1), dates_forecast(end)])
legend('Gaussian GARCH', 'Gaussian EGARCH', 't-GARCH',
    't-GJR-GARCH', 'Benchmark')

mean([ve1.^2,ve2.^2,ve3.^2,ve4.^2])

figure('Name','Comparison VaR')
plot(dates_forecast,[y(T+1:end-1),- vVar1, - vVar2, -
   vVar3, - vVar4])
hold on
plot(dates_forecast,- vVarES,'LineWidth',2,'color','r'
   )
xlabel('years')
ylabel('returns')
xlim([dates_forecast(1), dates_forecast(end)])
legend('Returns','Gaussian GARCH', 'Gaussian EGARCH',
   't-GARCH', 't-GJR-GARCH', 'Benchmark')
hold off

expected_exceed = m*0.01;
actual_exceed_1 = sum(y(T+1:end-1) < - vVar1);
actual_exceed_2 = sum(y(T+1:end-1) < - vVar2);
actual_exceed_3 = sum(y(T+1:end-1) < - vVar3); % less
   exceeds since t-student
actual_exceed_4 = sum(y(T+1:end-1) < - vVar4); % less
   exceeds since t-student
actual_exceed_ES = sum(y(T+1:end-1) < - vVarES);

idx_1     = y(T+1:end-1) < - vVar1;
returns_1 = y(T+1:end-1).*idx_1;
returns_1(returns_1 == 0) = nan;            % We find
   the returns that exceed the VAR

idx_2     = y(T+1:end-1) < - vVar2;
returns_2 = y(T+1:end-1).*idx_2;
returns_2(returns_2 == 0) = nan;            % We find
   the returns that exceed the VAR
```

```
idx_3      = y(T+1:end-1) < - vVar3;
returns_3 = y(T+1:end-1).*idx_3;
returns_3(returns_3 == 0) = nan;           % We find
   the returns that exceed the VAR

idx_4      = y(T+1:end-1) < - vVar4;
returns_4 = y(T+1:end-1).*idx_4;
returns_4(returns_4 == 0) = nan;           % We find
   the returns that exceed the VAR

idx_ES     = y(T+1:end-1) < - vVarES;
returns_es = y(T+1:end-1).*idx_ES;
returns_es(returns_es == 0) = nan;         % We find
   the returns that exceed the VAR

subplot(5,1,1)
plot(dates_forecast, y(T+1:end-1))
hold on
plot(dates_forecast, - vVarES),
hold on
plot(dates_forecast, returns_es,'r*')
legend('Returns' , 'Var Riskmetrics','Returns that
   exceed VaR')
xlabel('years')
ylabel('returns')
hold off

subplot(5,1,2)
plot(dates_forecast, y(T+1:end-1))
hold on
plot(dates_forecast, - vVar1)
hold on
plot(dates_forecast, returns_1,'r*')
legend('Returns' , 'VaR Gaussian GARCH','Returns that
   exceed VaR')
xlabel('years')
ylabel('returns')
hold off

subplot(5,1,3)
plot(dates_forecast, y(T+1:end-1))
hold on
plot(dates_forecast, - vVar2)
hold on
plot(dates_forecast, returns_2,'r*')
```

```
legend('Returns' , 'VaR Gaussian EGARCH','Returns that
    exceed VaR')
xlabel('years')
ylabel('returns')
hold off

subplot(5,1,4)
plot(dates_forecast, y(T+1:end-1))
hold on
plot(dates_forecast, - vVar3)
hold on
plot(dates_forecast, returns_3,'r*')
legend('Returns' , 'VaR T-Student GARCH','Returns that
    exceed VaR')
xlabel('years')
ylabel('returns')
hold off

subplot(5,1,5)
plot(dates_forecast, y(T+1:end-1))
hold on
plot(dates_forecast, - vVar4)
hold on
plot(dates_forecast, returns_4,'r*')
legend('Returns' , 'VaR T-Student GJR-GARCH','Returns
   that exceed VaR')
xlabel('years')
ylabel('returns')
hold off

%% LOSS
y1 = y(T+1:end-1);

% Gaussian GARCH
loss_1 = fCheckFunction(y1 + vVar1, dp);
total_loss_1= sum(loss_1);

% Gaussian EGARCH
loss_2 = fCheckFunction(y1 + vVar2, dp);
total_loss_2 = sum(loss_2);

% T-Student GARCH
loss_3 = fCheckFunction(y1 + vVar3, dp);
total_loss_3 = sum(loss_3);

% T-Student GJR-GARCH
```

```
loss_4 = fCheckFunction(y1 + vVar4, dp);
total_loss_4 = sum(loss_4);

% Riskmetrics
loss_ES = fCheckFunction(y1 + vVarES, dp);
total_loss_ES = sum(loss_ES);

%% Diebold-Mariano test
m_DM     = m^(1/3);
omega = 0;

% Gaussian GARCH
d_1 = loss_1 - loss_ES;
D_1 = mean(d_1);
spec_dens_1 = 2*pi*fBartlettSpectralDensityEst(d_1,
   m_DM,omega);
DM_1 = D_1./(sqrt(spec_dens_1./m))    % Result of
   Diebold-Mariano test
pvalue_1 = normcdf(DM_1,dp)           % pvalue of
   Diebold-Mariano test

% Gaussian EGARCH
d_2 = loss_2 - loss_ES;
D_2 = mean(d_2);
spec_dens_2 = 2*pi*fBartlettSpectralDensityEst(d_2,
   m_DM,omega);
DM_2 = D_2./(sqrt(spec_dens_2./m))    % Result of
   Diebold-Mariano test
pvalue_2 = normcdf(DM_2,dp)           % pvalue of
   Diebold-Mariano test

% T-Student GARCH
d_3 = loss_3 - loss_ES;
D_3 = mean(d_3);
spec_dens_3 = 2*pi*fBartlettSpectralDensityEst(d_3,
   m_DM,omega);
DM_3 = D_3./(sqrt(spec_dens_3./m))    % Result of
   Diebold-Mariano test
pvalue_3 = normcdf(DM_3,dp)           % pvalue of
   Diebold-Mariano test

% T-Student GJR-GARCH
d_4 = loss_4 - loss_ES;
D_4 = mean(d_4);
spec_dens_4 = 2*pi*fBartlettSpectralDensityEst(d_4,
   m_DM,omega);
```

```
DM_4 = D_4./(sqrt(spec_dens_4./m))     % Result of
   Diebold-Mariano test
pvalue_4 = normcdf(DM_4,dp)            % pvalue of
   Diebold-Mariano test

%% MSFE
sqrd_returns = y1.^2;
MSFE_ES = 1/m*sum((sqrd_returns - vh_frcst_es).^2); %
   Mean squared forecast error of Riskmetrics
MSFE_1 = 1/m*sum((sqrd_returns - vh_frcst1).^2);    %
   Mean squared forecast error of Gaussian Garch
MSFE_2 = 1/m*sum((sqrd_returns -vh_frcst2).^2); % Mean
    squared forecast error of Gaussian EGARCH
MSFE_3 = 1/m*sum((sqrd_returns -vh_frcst3).^2); % Mean
    squared forecast error of T-Student Garch
MSFE_4 = 1/m*sum((sqrd_returns -vh_frcst4).^2); % Mean
    squared forecast error of T-Student GJR-GARCH
```