

```

%% IMPORT AND TRANSFORM DATA - IN-SAMPLE
clear

% Import data and rename variables
DataMW = readtable("DataMW.xlsx");
DataMM = readtable("DataMM.xlsx");
DataQM = readtable("DataQM.xlsx");
DataQQ = readtable("DataQQ.xlsx");

DataMW.Properties.VariableNames = {'DateMW' 'NEX1MW' 'SYSMW'};
DataMM.Properties.VariableNames = {'DateMM' 'NEX1MM' 'SYSMM'};
DataQM.Properties.VariableNames = {'DateQM' 'NEX1QM' 'SYSQM'};
DataQQ.Properties.VariableNames = {'DateQQ' 'NEX1QQ' 'SYSQQ'};

% Transform the data into log prices
DataMW.LogNEX1MW = log(DataMW.NEX1MW);
DataMW.LogSYSMW = log(DataMW.SYSMW);

DataMM.LogNEX1MM = log(DataMM.NEX1MM);
DataMM.LogSYSMM = log(DataMM.SYSMM);

DataQM.LogNEX1QM = log(DataQM.NEX1QM);
DataQM.LogSYSQM = log(DataQM.SYSQM);

DataQQ.LogNEX1QQ = log(DataQQ.NEX1QQ);
DataQQ.LogSYSQQ = log(DataQQ.SYSQQ);

% Calculate log returns
DataRetMW = table();
DataRetMW.DateMW = DataMW.DateMW(2:end);
DataRetMW.RetLogSYSMW = DataMW.LogSYSMW(2:end) - DataMW.LogSYSMW(1:end-1);
DataRetMW.RetLogNEX1MW = DataMW.LogNEX1MW(2:end) - DataMW.LogNEX1MW(1:end-1);

DataRetMM = table();
DataRetMM.DateMM = DataMM.DateMM(2:end);
DataRetMM.RetLogSYSMM = DataMM.LogSYSMM(2:end) - DataMM.LogSYSMM(1:end-1);
DataRetMM.RetLogNEX1MM = DataMM.LogNEX1MM(2:end) - DataMM.LogNEX1MM(1:end-1);

DataRetQM = table();
DataRetQM.DateQM = DataQM.DateQM(2:end);
DataRetQM.RetLogSYSQM = DataQM.LogSYSQM(2:end) - DataQM.LogSYSQM(1:end-1);
DataRetQM.RetLogNEX1QM = DataQM.LogNEX1QM(2:end) - DataQM.LogNEX1QM(1:end-1);

DataRetQQ = table();
DataRetQQ.DateQQ = DataQQ.DateQQ(2:end);
DataRetQQ.RetLogSYSQQ = DataQQ.LogSYSQQ(2:end) - DataQQ.LogSYSQQ(1:end-1);
DataRetQQ.RetLogNEX1QQ = DataQQ.LogNEX1QQ(2:end) - DataQQ.LogNEX1QQ(1:end-1);

%% IMPORT AND TRANSFORM DATA - OUT-OF-SAMPLE
% Import data and rename variables
DataMWout = readtable("DataMWout.xlsx");
DataMMout = readtable("DataMMout.xlsx");
DataQMout = readtable("DataQMout.xlsx");
DataQQout = readtable("DataQQout.xlsx");

DataMWout.Properties.VariableNames = {'DateMWout' 'NEX1MWout' 'SYSMWout'};

```

```
DataMMout.Properties.VariableNames = {'DateMMout' 'NEX1MMout' 'SYSMMout'};
DataQMout.Properties.VariableNames = {'DateQMout' 'NEX1QMout' 'SYSQMout'};
DataQQout.Properties.VariableNames = {'DateQQout' 'NEX1QQout' 'SYSQQout'};
```

```
% Transform the data into log prices
```

```
DataMWout.LogNEX1MWout = log(DataMWout.NEX1MWout);
DataMWout.LogSYSMWout = log(DataMWout.SYSMWout);
```

```
DataMMout.LogNEX1MMout = log(DataMMout.NEX1MMout);
DataMMout.LogSYSMMout = log(DataMMout.SYSMMout);
```

```
DataQMout.LogNEX1QMout = log(DataQMout.NEX1QMout);
DataQMout.LogSYSQMout = log(DataQMout.SYSQMout);
```

```
DataQQout.LogNEX1QQout = log(DataQQout.NEX1QQout);
DataQQout.LogSYSQQout = log(DataQQout.SYSQQout);
```

```
% Calculate log returns
```

```
DataRetMWout = table();
DataRetMWout.DateMWout = DataMWout.DateMWout(2:end);
DataRetMWout.RetLogNEX1MWout = DataMWout.LogNEX1MWout(2:end) -
DataMWout.LogNEX1MWout(1:end-1);
DataRetMWout.RetLogSYSMWout = DataMWout.LogSYSMWout(2:end) -
DataMWout.LogSYSMWout(1:end-1);
StdSYSMWout = std(DataRetMWout.RetLogSYSMWout)
StdNEX1MWout = std(DataRetMWout.RetLogNEX1MWout)
```

```
DataRetMMout = table();
DataRetMMout.DateMMout = DataMMout.DateMMout(2:end);
DataRetMMout.RetLogNEX1MMout = DataMMout.LogNEX1MMout(2:end) -
DataMMout.LogNEX1MMout(1:end-1);
DataRetMMout.RetLogSYSMMout = DataMMout.LogSYSMMout(2:end) -
DataMMout.LogSYSMMout(1:end-1);
StdSYSMMout = std(DataRetMMout.RetLogSYSMMout)
StdNEX1MMout = std(DataRetMMout.RetLogNEX1MMout)
```

```
DataRetQMout = table();
DataRetQMout.DateQMout = DataQMout.DateQMout(2:end);
DataRetQMout.RetLogNEX1QMout = DataQMout.LogNEX1QMout(2:end) -
DataQMout.LogNEX1QMout(1:end-1);
DataRetQMout.RetLogSYSQMout = DataQMout.LogSYSQMout(2:end) -
DataQMout.LogSYSQMout(1:end-1);
StdSYSQMout = std(DataRetQMout.RetLogSYSQMout)
StdNEX1QMout = std(DataRetQMout.RetLogNEX1QMout)
```

```
DataRetQQout = table();
DataRetQQout.DateQQout = DataQQout.DateQQout(2:end);
DataRetQQout.RetLogNEX1QQout = DataQQout.LogNEX1QQout(2:end) -
DataQQout.LogNEX1QQout(1:end-1);
DataRetQQout.RetLogSYSQQout = DataQQout.LogSYSQQout(2:end) -
DataQQout.LogSYSQQout(1:end-1);
StdSYSQQout = std(DataRetQQout.RetLogSYSQQout)
StdNEX1QQout = std(DataRetQQout.RetLogNEX1QQout)
```

```
%% SECTION II - DESCRIPTIVE STATISTICS AND STATISTICAL TESTS
```

```
% Descriptive spot and futures return statistics
```

```
MeanRetLogSYSMW = mean(DataRetMW.RetLogSYSMW)
```

```

SDRetLogSYSMW      = std(DataRetMW.RetLogSYSMW)
SkewRetLogSYSMW    = skewness(DataRetMW.RetLogSYSMW)
KurtRetLogSYSMW    = kurtosis(DataRetMW.RetLogSYSMW)

MeanRetLogNEX1MW   = mean(DataRetMW.RetLogNEX1MW)
SDRetLogNEX1MW     = std(DataRetMW.RetLogNEX1MW)
SkewRetLogNEX1MW   = skewness(DataRetMW.RetLogNEX1MW)
KurtRetLogNEX1MW   = kurtosis(DataRetMW.RetLogNEX1MW)

MeanRetLogSYSMM    = mean(DataRetMM.RetLogSYSMM)
SDRetLogSYSMM      = std(DataRetMM.RetLogSYSMM)
SkewRetLogSYSMM    = skewness(DataRetMM.RetLogSYSMM)
KurtRetLogSYSMM    = kurtosis(DataRetMM.RetLogSYSMM)

MeanRetLogNEX1MM   = mean(DataRetMM.RetLogNEX1MM)
SDRetLogNEX1MM     = std(DataRetMM.RetLogNEX1MM)
SkewRetLogNEX1MM   = skewness(DataRetMM.RetLogNEX1MM)
KurtRetLogNEX1MM   = kurtosis(DataRetMM.RetLogNEX1MM)

MeanRetLogSYSQM    = mean(DataRetQM.RetLogSYSQM)
SDRetLogSYSQM      = std(DataRetQM.RetLogSYSQM)
SkewRetLogSYSQM    = skewness(DataRetQM.RetLogSYSQM)
KurtRetLogSYSQM    = kurtosis(DataRetQM.RetLogSYSQM)

MeanRetLogNEX1QM   = mean(DataRetQM.RetLogNEX1QM)
SDRetLogNEX1QM     = std(DataRetQM.RetLogNEX1QM)
SkewRetLogNEX1QM   = skewness(DataRetQM.RetLogNEX1QM)
KurtRetLogNEX1QM   = kurtosis(DataRetQM.RetLogNEX1QM)

MeanRetLogSYSQQ    = mean(DataRetQQ.RetLogSYSQQ)
SDRetLogSYSQQ      = std(DataRetQQ.RetLogSYSQQ)
SkewRetLogSYSQQ    = skewness(DataRetQQ.RetLogSYSQQ)
KurtRetLogSYSQQ    = kurtosis(DataRetQQ.RetLogSYSQQ)

MeanRetLogNEX1QQ   = mean(DataRetQQ.RetLogNEX1QQ)
SDRetLogNEX1QQ     = std(DataRetQQ.RetLogNEX1QQ)
SkewRetLogNEX1QQ   = skewness(DataRetQQ.RetLogNEX1QQ)
KurtRetLogNEX1QQ   = kurtosis(DataRetQQ.RetLogNEX1QQ)

% Jarque-Bera - testing non-normality
[h1,pval1,jbstat1,critval1] = jbtest(DataRetMW.RetLogSYSMW, 0.01);
[h2,pval2,jbstat2,critval2] = jbtest(DataRetMW.RetLogNEX1MW, 0.01);
[h3,pval3,jbstat3,critval3] = jbtest(DataRetMM.RetLogSYSMM, 0.01);
[h4,pval4,jbstat4,critval4] = jbtest(DataRetMM.RetLogNEX1MM, 0.01);
[h5,pval5,jbstat5,critval5] = jbtest(DataRetQM.RetLogSYSQM, 0.01);
[h6,pval6,jbstat6,critval6] = jbtest(DataRetQM.RetLogNEX1QM, 0.01);
[h7,pval7,jbstat7,critval7] = jbtest(DataRetQQ.RetLogSYSQQ, 0.01);
[h8,pval8,jbstat8,critval8] = jbtest(DataRetQQ.RetLogNEX1QQ, 0.01);

% ADF - unit roots - levels
[r1,p1,adfstat1,crit1] = adftest(DataMW.LogSYSMW);
[r2,p2,adfstat2,crit2] = adftest(DataMW.LogNEX1MW);
[r3,p3,adfstat3,crit3] = adftest(DataMM.LogSYSMM);
[r4,p4,adfstat4,crit4] = adftest(DataMM.LogNEX1MM);
[r5,p5,adfstat5,crit5] = adftest(DataQM.LogSYSQM);
[r6,p6,adfstat6,crit6] = adftest(DataQM.LogNEX1QM);
[r7,p7,adfstat7,crit7] = adftest(DataQQ.LogSYSQQ);

```

```
[r8,p8,adfstat8,crit8] = adfstest(DataQQ.LogNEX1QQ);
```

```
% ADF - unit roots - first differences
```

```
[r9,p9,adfstat9,crit9] = adfstest(DataRetMW.RetLogSYSMW);  
[r10,p10,adfstat10,crit10] = adfstest(DataRetMW.RetLogNEX1MW);  
[r11,p11,adfstat11,crit11] = adfstest(DataRetMM.RetLogSYSMM);  
[r12,p12,adfstat12,crit12] = adfstest(DataRetMM.RetLogNEX1MM);  
[r13,p13,adfstat13,crit13] = adfstest(DataRetQM.RetLogSYSQM);  
[r14,p14,adfstat14,crit14] = adfstest(DataRetQM.RetLogNEX1QM);  
[r15,p15,adfstat15,crit15] = adfstest(DataRetQQ.RetLogSYSQQ);  
[r16,p16,adfstat16,crit16] = adfstest(DataRetQQ.RetLogNEX1QQ);
```

```
% KPSS - stationarity - levels
```

```
[s1,pv1,kpssstat1,cv1] = kpsstest(DataMW.LogSYSMW);  
[s2,pv2,kpssstat2,cv2] = kpsstest(DataMW.LogNEX1MW);  
[s3,pv3,kpssstat3,cv3] = kpsstest(DataMM.LogSYSMM);  
[s4,pv4,kpssstat4,cv4] = kpsstest(DataMM.LogNEX1MM);  
[s5,pv5,kpssstat5,cv5] = kpsstest(DataQM.LogSYSQM);  
[s6,pv6,kpssstat6,cv6] = kpsstest(DataQM.LogNEX1QM);  
[s7,pv7,kpssstat7,cv7] = kpsstest(DataQQ.LogSYSQQ);  
[s8,pv8,kpssstat8,cv8] = kpsstest(DataQQ.LogNEX1QQ);
```

```
% KPSS - stationarity - first differences
```

```
[s9,pv9,kpssstat9,cv9] = kpsstest(DataRetMW.RetLogSYSMW);  
[s10,pv10,kpssstat10,cv10] = kpsstest(DataRetMW.RetLogNEX1MW);  
[s11,pv11,kpssstat11,cv11] = kpsstest(DataRetMM.RetLogSYSMM);  
[s12,pv12,kpssstat12,cv12] = kpsstest(DataRetMM.RetLogNEX1MM);  
[s13,pv13,kpssstat13,cv13] = kpsstest(DataRetQM.RetLogSYSQM);  
[s14,pv14,kpssstat14,cv14] = kpsstest(DataRetQM.RetLogNEX1QM);  
[s15,pv15,kpssstat15,cv15] = kpsstest(DataRetQQ.RetLogSYSQQ);  
[s16,pv16,kpssstat16,cv16] = kpsstest(DataRetQQ.RetLogNEX1QQ);
```

```
% Find residuals - deviations from mean - to test autocorrelation and  
% heteroscedasticity
```

```
ResSYSMW = DataRetMW.RetLogSYSMW - MeanRetLogSYSMW;  
ResNEX1MW = DataRetMW.RetLogNEX1MW - MeanRetLogNEX1MW;  
ResSYSMM = DataRetMM.RetLogSYSMM - MeanRetLogSYSMM;  
ResNEX1MM = DataRetMM.RetLogNEX1MM - MeanRetLogNEX1MM;  
ResSYSQM = DataRetQM.RetLogSYSQM - MeanRetLogSYSQM;  
ResNEX1QM = DataRetQM.RetLogNEX1QM - MeanRetLogNEX1QM;  
ResSYSQQ = DataRetQQ.RetLogSYSQQ - MeanRetLogSYSQQ;  
ResNEX1QQ = DataRetQQ.RetLogNEX1QQ - MeanRetLogNEX1QQ;
```

```
% Ljung-Box - autocorrelation - Q(.)
```

```
[LB1,pvalue1,LBstat1,cval1] = lbqtest(ResSYSMW, 'lags', [6,12])  
[LB2,pvalue2,LBstat2,cval2] = lbqtest(ResNEX1MW, 'lags', [6,12])  
[LB3,pvalue3,LBstat3,cval3] = lbqtest(ResSYSMM, 'lags', [6,12])  
[LB4,pvalue4,LBstat4,cval4] = lbqtest(ResNEX1MM, 'lags', [6,12])  
[LB5,pvalue5,LBstat5,cval5] = lbqtest(ResSYSQM, 'lags', [6,12])  
[LB6,pvalue6,LBstat6,cval6] = lbqtest(ResNEX1QM, 'lags', [6,12])  
[LB7,pvalue7,LBstat7,cval7] = lbqtest(ResSYSQQ, 'lags', [6])  
[LB8,pvalue8,LBstat8,cval8] = lbqtest(ResNEX1QQ, 'lags', [6])
```

```
% Ljung-Box - arch-effects - Q^2(.)
```

```
[LB9,pvalue9,LBstat9,cval9] = lbqtest(ResSYSMW.^2, 'lags', [6,12])  
[LB10,pvalue10,LBstat10,cval10] = lbqtest(ResNEX1MW.^2, 'lags', [6,12])  
[LB11,pvalue11,LBstat11,cval11] = lbqtest(ResSYSMM.^2, 'lags', [6,12])
```

```

[LB12,pvalue12,LBstat12,cval12] = lbqtest(ResNEX1MM.^2,'lags',[6,12])
[LB13,pvalue13,LBstat13,cval13] = lbqtest(ResSYSQM.^2,'lags',[6,12])
[LB14,pvalue14,LBstat14,cval14] = lbqtest(ResNEX1QM.^2,'lags',[6,12])
[LB15,pvalue15,LBstat15,cval15] = lbqtest(ResSYSQQ.^2,'lags',[6])
[LB16,pvalue16,LBstat16,cval16] = lbqtest(ResNEX1QQ.^2,'lags',[6])

% ARCH-effects with archtest
[k1,pValue1,archstat1,cValue1] = archtest(ResSYSMW, 'Lags', [6,12])
[k2,pValue2,archstat2,cValue2] = archtest(ResNEX1MW, 'Lags', [6,12])
[k3,pValue3,archstat3,cValue3] = archtest(ResSYSMM, 'Lags', [6,12])
[k4,pValue4,archstat4,cValue4] = archtest(ResNEX1MM, 'Lags', [6,12])
[k5,pValue5,archstat5,cValue5] = archtest(ResSYSQM, 'Lags', [6,12])
[k6,pValue6,archstat6,cValue6] = archtest(ResNEX1QM, 'Lags', [6,12])
[k7,pValue7,archstat7,cValue7] = archtest(ResSYSQQ, 'Lags', [6])
[k8,pValue8,archstat8,cValue8] = archtest(ResNEX1QQ, 'Lags', [6])

% Correlation between spot and futures - both in and out of sample
CorrMW = corr(DataRetMW.RetLogNEX1MW,DataRetMW.RetLogSYSMW)
CorrMM = corr(DataRetMM.RetLogNEX1MM,DataRetMM.RetLogSYSMM)
CorrQM = corr(DataRetQM.RetLogNEX1QM,DataRetQM.RetLogSYSQM)
CorrQQ = corr(DataRetQQ.RetLogNEX1QQ,DataRetQQ.RetLogSYSQQ)

CorrMWout = corr(DataRetMWout.RetLogNEX1MWout,DataRetMWout.RetLogSYSMWout)
CorrMMout = corr(DataRetMMout.RetLogNEX1MMout,DataRetMMout.RetLogSYSMMout)
CorrQMout = corr(DataRetQMout.RetLogNEX1QMout,DataRetQMout.RetLogSYSQMout)
CorrQQout = corr(DataRetQQout.RetLogNEX1QQout,DataRetQQout.RetLogSYSQQout)

%% SECTION III - HEDGE RATIOS
% IN-SAMPLE
% OLS hedge
OLSMW = fitlm(DataRetMW, 'RetLogSYSMW ~ RetLogNEX1MW')
BMW = OLSMW.Coefficients.Estimate(2)
[EstCov1,se1,coeff1] = hac(OLSMW)

OLSMM = fitlm(DataRetMM, 'RetLogSYSMM ~ RetLogNEX1MM')
BMM = OLSMM.Coefficients.Estimate(2)
[EstCov2,se2,coeff2] = hac(OLSMM)

OLSQM = fitlm(DataRetQM, 'RetLogSYSQM ~ RetLogNEX1QM')
BQM = OLSQM.Coefficients.Estimate(2)
[EstCov3,se3,coeff3] = hac(OLSQM)

OLSQQ = fitlm(DataRetQQ, 'RetLogSYSQQ ~ RetLogNEX1QQ')
BQQ = OLSQQ.Coefficients.Estimate(2)
[EstCov4,se4,coeff4] = hac(OLSQQ)

% CCC-GARCH
% For MW
GarchSYSMW = garch('GARCHlags', 1, 'ARCHlags', 1)
EstGarchSYSMW = estimate(GarchSYSMW, DataRetMW.RetLogSYSMW);
DataRetMW.ConVarSYSMW = infer(EstGarchSYSMW, DataRetMW.RetLogSYSMW);
DataRetMW.ConStdSYSMW = sqrt(DataRetMW.ConVarSYSMW);

GarchNEX1MW = garch('GARCHlags', 1, 'ARCHlags', 1)
EstGarchNEX1MW = estimate(GarchNEX1MW, DataRetMW.RetLogNEX1MW);
DataRetMW.ConVarNEX1MW = infer(EstGarchNEX1MW, DataRetMW.RetLogNEX1MW);
DataRetMW.ConStdNEX1MW = sqrt(DataRetMW.ConVarNEX1MW);

```

```

CovMW = CorrMW .* DataRetMW.ConStdSYSMW .* DataRetMW.ConStdNEX1MW;
BetaMW = CovMW(2:end) ./ DataRetMW.ConVarNEX1MW(2:end)

% OUT-OF-SAMPLE - for MW
DataRetMWout.ForConVarSYSMW = forecast(EstGarchSYSMW, 51, 'Y0',
DataRetMW.ConVarSYSMW)
ForConStdSYSMWout = sqrt(DataRetMWout.ForConVarSYSMW)

DataRetMWout.ForConVarNEX1MW = forecast(EstGarchNEX1MW, 51, 'Y0',
DataRetMW.ConVarNEX1MW)
ForConStdNEX1MWout = sqrt(DataRetMWout.ForConVarNEX1MW)

ForCovMWout = CorrMW .* ForConStdSYSMWout .* ForConStdNEX1MWout
ForBetaMWout = ForCovMWout(2:end) ./ DataRetMWout.ForConVarNEX1MW(2:end)

GarchSYSMWout = garch('GARCHlags', 1, 'ARCHlags', 1)
EstGarchSYSMWout = estimate(GarchSYSMWout,
DataRetMWout.RetLogSYSMWout);
DataRetMWout.ConVarSYSMWout = infer(EstGarchSYSMWout,
DataRetMWout.RetLogSYSMWout);
DataRetMWout.ConStdSYSMWout = sqrt(DataRetMWout.ConVarSYSMWout)

options = optimoptions(@fmincon, 'Algorithm', 'interior-
point');
GarchNEX1MWout = garch('GARCHlags', 1, 'ARCHlags', 1)
EstGarchNEX1MWout = estimate(GarchNEX1MWout,
DataRetMWout.RetLogNEX1MWout, 'options', options);
DataRetMWout.ConVarNEX1MWout = infer(EstGarchNEX1MWout,
DataRetMWout.RetLogNEX1MWout);
DataRetMWout.ConStdNEX1MWout = sqrt(DataRetMWout.ConVarNEX1MWout)

CovMWout = CorrMWout * DataRetMWout.ConStdSYSMWout .*
DataRetMWout.ConStdNEX1MWout

%% SECTION IV - HEDGE VARIANCES
% IN-SAMPLE
% Unhedged variances
VarRetLogSYSMW = var(DataRetMW.RetLogSYSMW)
VarRetLogSYSMM = var(DataRetMM.RetLogSYSMM)
VarRetLogSYSQM = var(DataRetQM.RetLogSYSQM)
VarRetLogSYSQQ = var(DataRetQQ.RetLogSYSQQ)

% Naïve hedge
VarNaiveMW = var(DataRetMW.RetLogSYSMW) + (1)^2*var(DataRetMW.RetLogNEX1MW) -
(2*(1)*cov(DataRetMW.RetLogSYSMW, DataRetMW.RetLogNEX1MW))
VarNaiveMM = var(DataRetMM.RetLogSYSMM) + (1)^2*var(DataRetMM.RetLogNEX1MM) -
(2*(1)*cov(DataRetMM.RetLogSYSMM, DataRetMM.RetLogNEX1MM))
VarNaiveQM = var(DataRetQM.RetLogSYSQM) + (1)^2*var(DataRetQM.RetLogNEX1QM) -
(2*(1)*cov(DataRetQM.RetLogSYSQM, DataRetQM.RetLogNEX1QM))
VarNaiveQQ = var(DataRetQQ.RetLogSYSQQ) + (1)^2*var(DataRetQQ.RetLogNEX1QQ) -
(2*(1)*cov(DataRetQQ.RetLogSYSQQ, DataRetQQ.RetLogNEX1QQ))

% OLS hedge
VarOLSMW = var(DataRetMW.RetLogSYSMW) + (BMW)^2*var(DataRetMW.RetLogNEX1MW) -
(2*(BMW)*cov(DataRetMW.RetLogSYSMW, DataRetMW.RetLogNEX1MW))

```

```

VarOLSMM = var(DataRetMM.RetLogSYSMM) + (BMM)^2*var(DataRetMM.RetLogNEX1MM) -
(2*(BMM)*cov(DataRetMM.RetLogSYSMM,DataRetMM.RetLogNEX1MM))
VarOLSQM = var(DataRetQM.RetLogSYSQM) + (BQM)^2*var(DataRetQM.RetLogNEX1QM) -
(2*(BQM)*cov(DataRetQM.RetLogSYSQM,DataRetQM.RetLogNEX1QM))
VarOLSQQ = var(DataRetQQ.RetLogSYSQQ) + (BQQ)^2*var(DataRetQQ.RetLogNEX1QQ) -
(2*(BQQ)*cov(DataRetQQ.RetLogSYSQQ,DataRetQQ.RetLogNEX1QQ))

```

% CCC-GARCH

```

VarGarchMW = DataRetMW.ConVarSYSMW(2:end) +
BetaMW.^2.*DataRetMW.ConVarNEX1MW(2:end) - 2.*BetaMW.*CovMW(2:end)
meanVarGarchMW = mean(VarGarchMW)

```

% OUT-OF-SAMPLE

% Unhedged variances

```

VarRetLogSYSMWout = var(DataRetMWout.RetLogSYSMWout)
VarRetLogSYSMMout = var(DataRetMMout.RetLogSYSMMout)
VarRetLogSYSQMout = var(DataRetQMout.RetLogSYSQMout)
VarRetLogSYSQQout = var(DataRetQQout.RetLogSYSQQout)

```

% Naïve hedge

```

VarNaiveMWout = var(DataRetMWout.RetLogSYSMWout) +
(1)^2*var(DataRetMWout.RetLogNEX1MWout) -
(2*(1)*cov(DataRetMWout.RetLogSYSMWout,DataRetMWout.RetLogNEX1MWout))
VarNaiveMMout = var(DataRetMMout.RetLogSYSMMout) +
(1)^2*var(DataRetMMout.RetLogNEX1MMout) -
(2*(1)*cov(DataRetMMout.RetLogSYSMMout,DataRetMMout.RetLogNEX1MMout))
VarNaiveQMout = var(DataRetQMout.RetLogSYSQMout) +
(1)^2*var(DataRetQMout.RetLogNEX1QMout) -
(2*(1)*cov(DataRetQMout.RetLogSYSQMout,DataRetQMout.RetLogNEX1QMout))
VarNaiveQQout = var(DataRetQQout.RetLogSYSQQout) +
(1)^2*var(DataRetQQout.RetLogNEX1QQout) -
(2*(1)*cov(DataRetQQout.RetLogSYSQQout,DataRetQQout.RetLogNEX1QQout))

```

% OLS hedge

```

VarOLSMWout = var(DataRetMWout.RetLogSYSMWout) +
(BMW)^2*var(DataRetMWout.RetLogNEX1MWout) -
(2*(BMW)*cov(DataRetMWout.RetLogSYSMWout,DataRetMWout.RetLogNEX1MWout))
VarOLSMMout = var(DataRetMMout.RetLogSYSMMout) +
(BMM)^2*var(DataRetMMout.RetLogNEX1MMout) -
(2*(BMM)*cov(DataRetMMout.RetLogSYSMMout,DataRetMMout.RetLogNEX1MMout))
VarOLSQMout = var(DataRetQMout.RetLogSYSQMout) +
(BQM)^2*var(DataRetQMout.RetLogNEX1QMout) -
(2*(BQM)*cov(DataRetQMout.RetLogSYSQMout,DataRetQMout.RetLogNEX1QMout))
VarOLSQQout = var(DataRetQQout.RetLogSYSQQout) +
(BQQ)^2*var(DataRetQQout.RetLogNEX1QQout) -
(2*(BQQ)*cov(DataRetQQout.RetLogSYSQQout,DataRetQQout.RetLogNEX1QQout))

```

% CCC-GARCH

```

VarGarchMWout = DataRetMWout.ConVarSYSMWout(2:end) +
ForBetaMWout.^2.*DataRetMWout.ConVarNEX1MWout(2:end) -
2.*ForBetaMWout.*CovMWout(2:end)
meanVarGarchMWout = mean(VarGarchMWout)

```

%% SECTION V - HEDGE PERFORMANCE

% IN-SAMPLE

% Naïve hedge

```

HENaiveMW = 1 - (VarNaiveMW/VarRetLogSYSMW)

```

```
HENaiveMM = 1 - (VarNaiveMM/VarRetLogSYSMM)
HENaiveQM = 1 - (VarNaiveQM/VarRetLogSYSQM)
HENaiveQQ = 1 - (VarNaiveQQ/VarRetLogSYSQQ)
```

```
% OLS hedge
```

```
HEOLSMW = 1 - (VarOLSMW/VarRetLogSYSMW)
HEOLSMM = 1 - (VarOLSMM/VarRetLogSYSMM)
HEOLSQM = 1 - (VarOLSQM/VarRetLogSYSQM)
HEOLSQQ = 1 - (VarOLSQQ/VarRetLogSYSQQ)
```

```
% GARCH
```

```
HEGarchMW = 1 - (meanVarGarchMW/VarRetLogSYSMW)
```

```
% OUT-OF-SAMPLE
```

```
% Naive hedge
```

```
HENaiveMWout = 1 - (VarNaiveMWout/VarRetLogSYSMWout)
HENaiveMMout = 1 - (VarNaiveMMout/VarRetLogSYSMMout)
HENaiveQMout = 1 - (VarNaiveQMout/VarRetLogSYSQMout)
HENaiveQQout = 1 - (VarNaiveQQout/VarRetLogSYSQQout)
```

```
% OLS hedge
```

```
HEOLSMWout = 1 - (VarOLSMWout/VarRetLogSYSMWout)
HEOLSMMout = 1 - (VarOLSMMout/VarRetLogSYSMMout)
HEOLSQMout = 1 - (VarOLSQMout/VarRetLogSYSQMout)
HEOLSQQout = 1 - (VarOLSQQout/VarRetLogSYSQQout)
```

```
% GARCH
```

```
HEGarchMWout = 1 - (meanVarGarchMWout/VarRetLogSYSMWout)
```