Forecasting Financial Market: Homework 2

Zixin Huang

February 17, 2020

Exercise 1

Question A

Function **LBtest.m**:

```
function [LBh,LBstat,pvalue]=LBtest(X,L)

% this function performs the Ljung-Box test

% Input: X (T*N), a time series data; L, a scalar representing the lags

% Output: LBh, the logical value representing the result of the test, (0 do
% not reject, 1 reject) ; LBstat, the LB statistic ; pvalue, the pvalue
% corresponding to this test stat.

[LBh,pvalue,LBstat,~]=lbqtest(X,'Lags',L);



end
```

Question B

Function **robustLBtest.m**:

```
function [chi2stat,pvalue]=robustLBtest(X,L)

% this function performs robust Ljung-Box test

% Input: X (T*N) a time series data, L, number of lags

% Ouptut: chi2stat, chi^2 statistic for serial correlation; pvalue, the
% p-value corresponding to this test statistics

lagged_data = lagmatrix(X,L,mean(X));

temp = nwest(X(L+1:end),[ones(T-L,1),lagged_data(L+1:end,:)],L+3);
chi2stat = temp.beta(2:end)'*(temp.vcv(2:end,2:end)\temp.beta(2:end));
pvalue = chi2inv(1-0.05,L);



end
```

Question C

USD/EUR Exchange Rate

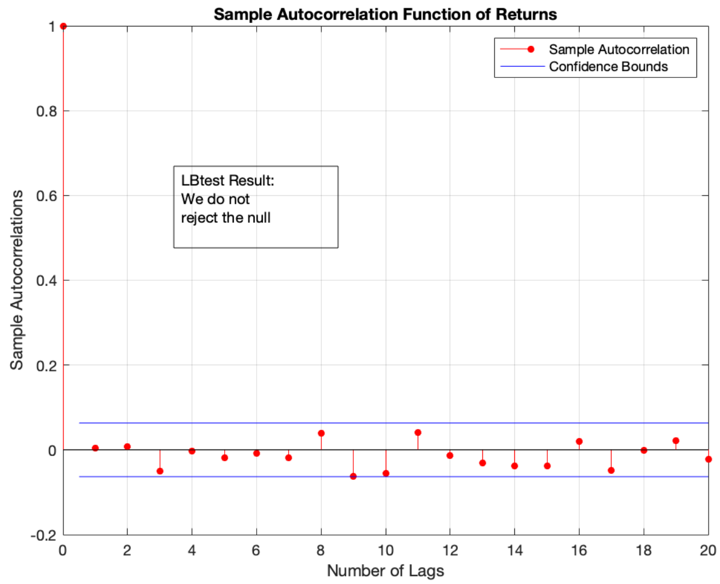

Figure 1: Sample Autocorrelation Function of Returns

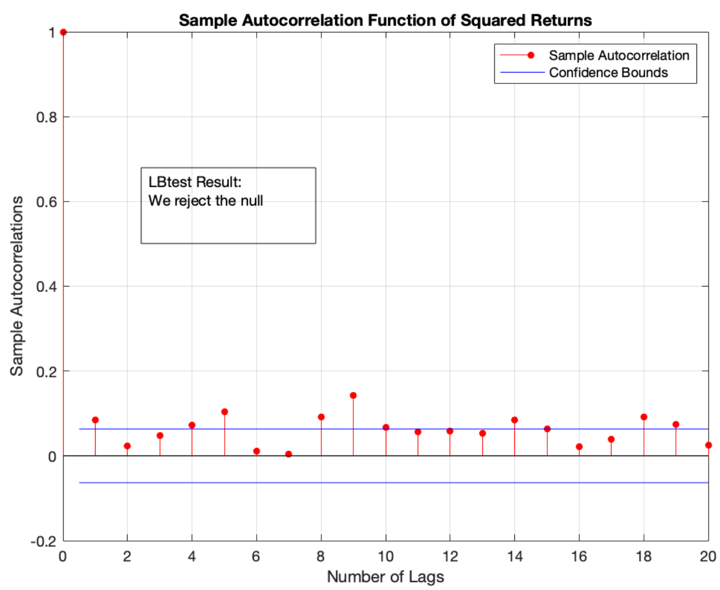

Figure 2: Sample Autocorrelation Function of Squared Returns

SP500

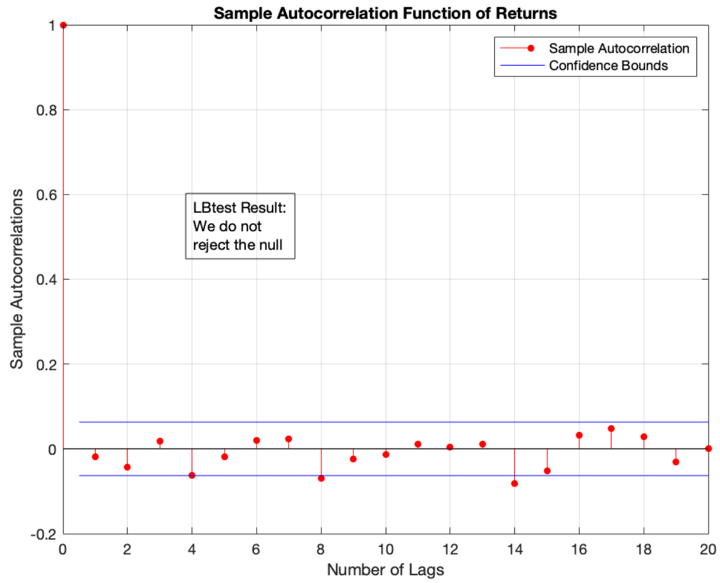

Figure 3: Sample Autocorrelation Function of Returns

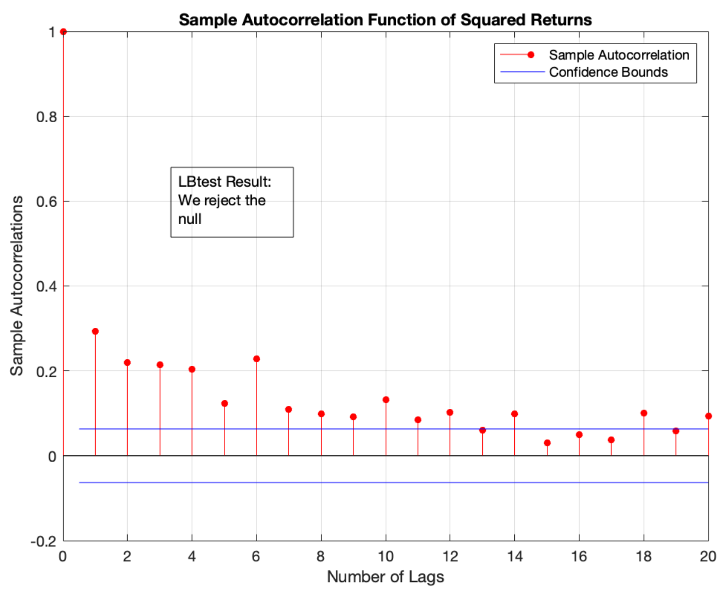

Figure 4: Sample Autocorrelation Function of Squared Returns

NASDAQ100

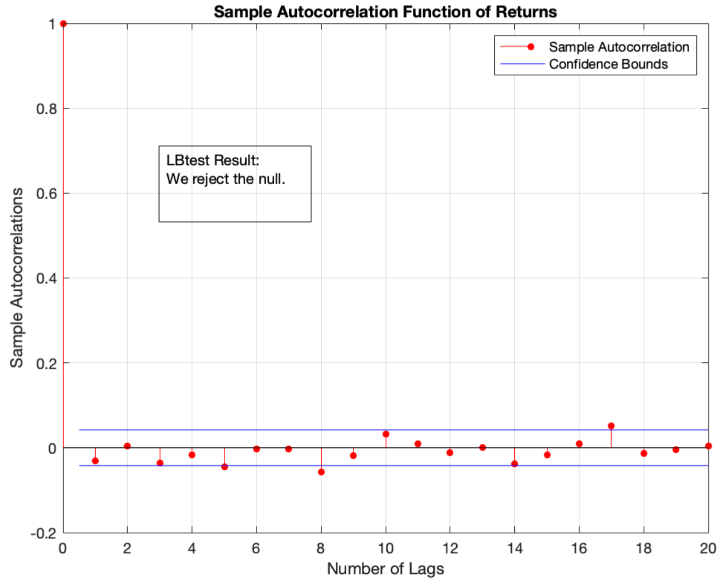

Figure 5: Sample Autocorrelation Function of Returns

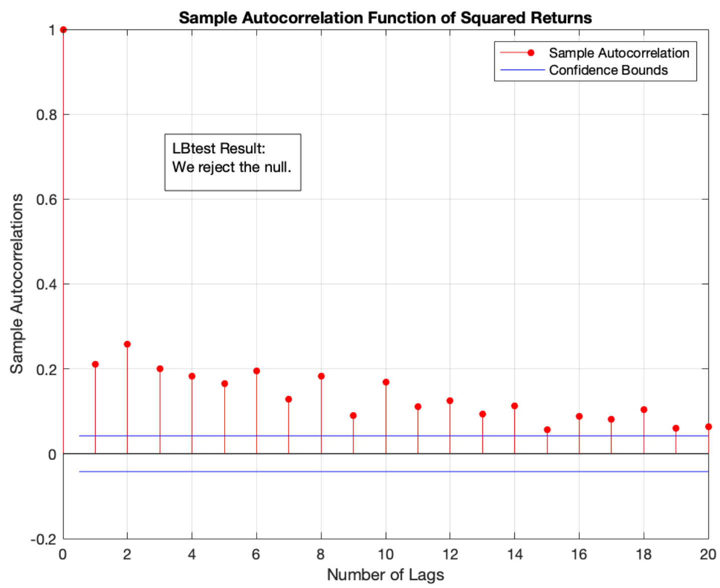

Figure 6: Sample Autocorrelation Function of Squared Returns

Exercise 2

Question A

Function **x3.m**:

```
function y = x3(x)

% this function takes in a scalar x and returns |x|^3

y = abs(x) ^3;



end
```

Question B

Table 1: Result of Minimizing $f(x) = |x|^3$

Function	**Initial Value**	**Solution**	**Function Value**
fminunc	2	0.002	8.3202×10^{-9}

MATLAB process:

```
>> Q2
                                                        First-order
 Iteration  Func-count       f(x)        Step-size       optimality
     0           2                8                          12
     1           4                1      0.0833333            3
     2           6         0.296296              1         1.33
     3           8            0.064              1         0.48
     4          10         0.015625              1        0.188
     5          12       0.00364133              1        0.071
     6          14      0.000863837              1       0.0272
     7          16      0.000203542              1       0.0104
     8          18      4.80841e-05              1      0.00397
     9          20       1.1348e-05              1      0.00151
    10          22      2.67918e-06              1     0.000579
    11          24      6.32443e-07              1     0.000221
    12          26      1.49301e-07              1     8.44e-05
    13          28       3.5245e-08              1     3.22e-05
    14          30       8.3202e-09              1     1.23e-05

Local minimum found.

Optimization completed because the size of the gradient is less than
the value of the optimality tolerance.

<stopping criteria details>

X1 =

    0.0020


FVAL1 =

   8.3202e-09


EXITFLAG1 =

     1


output1 =

  struct with fields:

       iterations: 14
        funcCount: 30
         stepsize: 0.0013
     lssteplength: 1
    firstorderopt: 1.2318e-05
        algorithm: 'quasi-newton'
          message: '↵Local minimum found.↵↵Optimization completed because the size of the↙
gradient is less than↵the value of the optimality tolerance.↵↵<stopping criteria↙
details>↵↵Optimization completed: The first-order optimality measure, 9.475532e-07, is↙
less ↵than options.OptimalityTolerance = 1.000000e-06.↵↵'

>>
```

Question C

Table 2: Result of Minimizing $f(x) = |x|^3$

Function	**Constraints**	**Initial Value**	**Solution**	**Function Value**
fmincon	$x \geq 1$	3	1	1

MATLAB process:

2/13/20 2:38 PM MATLAB Command Window 1 of 1

```
>> Q2
 Iter  Func-count            Fval   Feasibility   Step Length       Norm of   First-order
                                                                       step    optimality
    0           2    2.700000e+01     0.000e+00     1.000e+00     0.000e+00     2.700e+01
    1           4    1.000000e+00     0.000e+00     1.000e+00     2.000e+00     2.200e+01
    2           6    1.000000e+00     0.000e+00     1.000e+00     0.000e+00     0.000e+00

Local minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

<stopping criteria details>

X2 =

     1


FVAL2 =

     1


EXITFLAG2 =

     1


output2 =

  struct with fields:

         iterations: 2
          funcCount: 6
          algorithm: 'sqp'
            message: '↵Local minimum found that satisfies the constraints.↵↵Optimization↙
completed because the objective function is non-decreasing in ↵feasible directions, to↙
within the value of the optimality tolerance,↵and constraints are satisfied to within the↙
value of the constraint tolerance.↵↵<stopping criteria details>↵↵Optimization completed:↙
The relative first-order optimality measure, 0.000000e+00,↵is less than options.↙
OptimalityTolerance = 1.000000e-06, and the relative maximum constraint↵violation,↙
0.000000e+00, is less than options.ConstraintTolerance = 1.000000e-06.↵↵'
    constrviolation: 0
           stepsize: 0
       lssteplength: 1
      firstorderopt: 0

>>
```

Question D

Function **x4.m**:

```
function y=x4(x)

% this function takes in x and returns 2*(2-x^2)^2+x

y = 2*(2-x^2)^2+x;

end
```

Question E

Table 3: Result of Minimizing $f(x) = 2(2-x^2)^2 + x$

Function	Initial Value	Solution	Function Value
fminunc	0	-1.4445	-1.4295

Question F

Table 4: Result of Minimizing $f(x) = 2(2-x^2)^2 + x$

Function	Initial Value	Solution	Function Value
fminunc	1	1.3819	1.3982

Question G

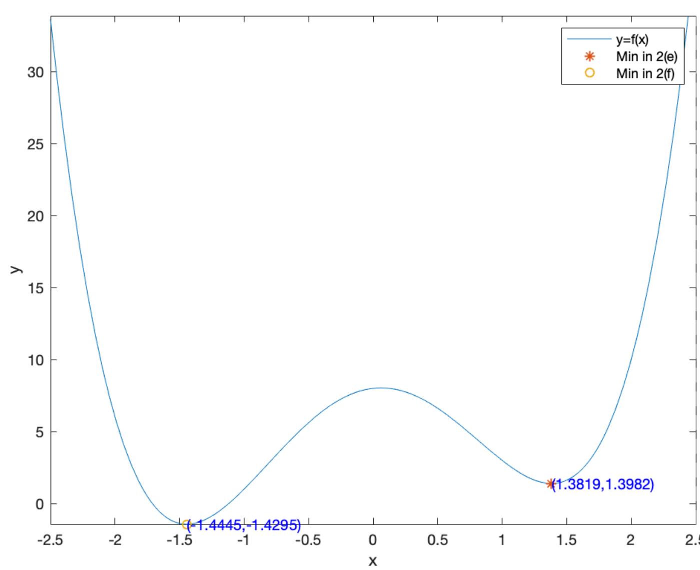

Figure 7: Plot of $f(x) = 2(2-x^2)^2 + x$ on $[-2.5, 2.5]$

Exercise 3

Question A

Function **LL_normal.m**:

```
function n_llN = LL_normal(theta,X)

% this function compute the negative log-likelihood of a univariate normal
% random variable

% Input: theta, (2*1) containing the mean and variance of the normal
    distribution;
% X, the vector of data

% Output: the negative log-likelihood

[T,~]=size(X); % number of obs of our data
ll=0;
for i = 1: T
    ll =ll+log(1/sqrt(2*pi*theta(2))*exp(-(X(i)-theta(1))^2/(2*theta(2))))
        ;

end

n_llN = -ll;

end
```

Question B

In this question, I use the SP500 daily returns from 2015 to 2019 as my time series data.

Table 5: Sample Statistics

Data	Sample Mean	Sample Variance
SP500 Daily Return	2.7163×10^{-4}	7.5238×10^{-5}

Table 6: Result of Finding the optimal value of $\hat{\theta} = (\mu, \sigma^2)$

Function	Constraints	Initial Value	Solution
fmincon	$\sigma^2 > 0$	$(0, 1)$	$(2.7166 \times 10^{-4}, 7.5156 \times 10^{-5})$

From Table 5 and Table 6 we can see that the optimal $\hat{\theta} = (\mu, \sigma^2)$ is nearly identical to the sample mean and the sample variance.

MATLAB process:

```
>> Q3
 Iter  Func-count            Fval   Feasibility   Step Length       Norm of   First-order
                                                                       step    optimality
    0           3    9.244900e+02     0.000e+00     1.000e+00     0.000e+00     5.030e+02
Objective function returned Inf; trying a new point...
    1           7    3.800923e+02     0.000e+00     7.000e-01     7.256e-01     1.121e+03
Objective function returned Inf; trying a new point...
    2          27    3.320016e+02     0.000e+00     2.326e-03     2.828e-01     1.279e+03
Objective function returned Inf; trying a new point...
    3          31   -2.653607e+02     0.000e+00     7.000e-01     2.115e-01     4.138e+03
Objective function returned Inf; trying a new point...
    4          35   -8.791418e+02     0.000e+00     7.000e-01     7.269e-02     1.399e+04
Objective function returned Inf; trying a new point...
    5          39   -1.490422e+03     0.000e+00     7.000e-01     2.629e-02     4.690e+04
Objective function returned Inf; trying a new point...
    6          43   -2.088830e+03     0.000e+00     7.000e-01     8.887e-03     1.532e+05
Objective function returned Inf; trying a new point...
    7          47   -2.657698e+03     0.000e+00     7.000e-01     2.722e-03     4.639e+05
Objective function returned Inf; trying a new point...
    8          51   -3.137587e+03     0.000e+00     7.000e-01     7.320e-04     1.009e+06
Objective function returned Inf; trying a new point...
    9          55   -3.325741e+03     0.000e+00     7.000e-01     1.637e-04     1.895e+06
Objective function returned Inf; trying a new point...
   10          67   -3.338965e+03     0.000e+00     4.035e-02     1.955e-03     2.074e+06
Objective function returned Inf; trying a new point...
   11         114   -3.340557e+03     0.000e+00     1.529e-07     1.550e-04     3.679e+05
Objective function returned Inf; trying a new point...
   12         142   -3.342003e+03     0.000e+00     1.341e-04     8.364e-04     3.835e+05
   13         145   -3.346378e+03     0.000e+00     1.000e+00     2.325e-04     7.725e+05
   14         148   -3.348663e+03     0.000e+00     1.000e+00     4.613e-05     2.207e+05
   15         151   -3.348952e+03     0.000e+00     1.000e+00     8.825e-06     4.274e+04
   16         154   -3.348962e+03     0.000e+00     1.000e+00     1.372e-06     3.137e+03
   17         157   -3.348962e+03     0.000e+00     1.000e+00     1.062e-07     4.173e+01

Local minimum possible. Constraints satisfied.

fmincon stopped because the size of the current step is less than
the value of the step size tolerance and constraints are
satisfied to within the value of the constraint tolerance.

<stopping criteria details>

thetahat =

   1.0e-03 *

    0.2717
    0.0752


fval =

  -3.3490e+03


exitflag =

     2
```

```
output =

  struct with fields:

         iterations: 17
          funcCount: 157
          algorithm: 'sqp'
            message: '↵Local minimum possible. Constraints satisfied.↵↵fmincon stopped↙
because the size of the current step is less than↵the value of the step size tolerance↙
and constraints are ↵satisfied to within the value of the constraint tolerance.↙
↵↵<stopping criteria details>↵↵Optimization stopped because the relative changes in all↙
elements of x are↵less than options.StepTolerance = 1.000000e-06, and the relative↙
maximum constraint↵violation, 0.000000e+00, is less than options.ConstraintTolerance =↙
1.000000e-06.↵↵'
    constrviolation: 0
           stepsize: 1.0624e-07
       lssteplength: 1
      firstorderopt: 41.7298

>>
```

Exercise 4

In this question, I use the de-meaned daily return of the USD/EUR exchange rates from 2015 to 2019 as my time series data. I choose parameters $\hat{\theta}$ by maximizing the log-likelihood of my data.

In GARCH model, we estimate series of variance by:

$$\sigma_t^2 = \omega + \beta\sigma_{t-1}^2 + \alpha\epsilon_{t-1}^2 \tag{1}$$

By applying maximum likelihood, we can find the optimal $\hat{\theta} = (\omega, \beta, \alpha)$:

Table 7: Result of Optimal $\hat{\theta}$

ω	β	α
1.04×10^{-14}	2.4793×10^{-8}	0.9997

Then I use the optimal $\hat{\theta}$ I found to generate the GARCH variance series σ_t^2.

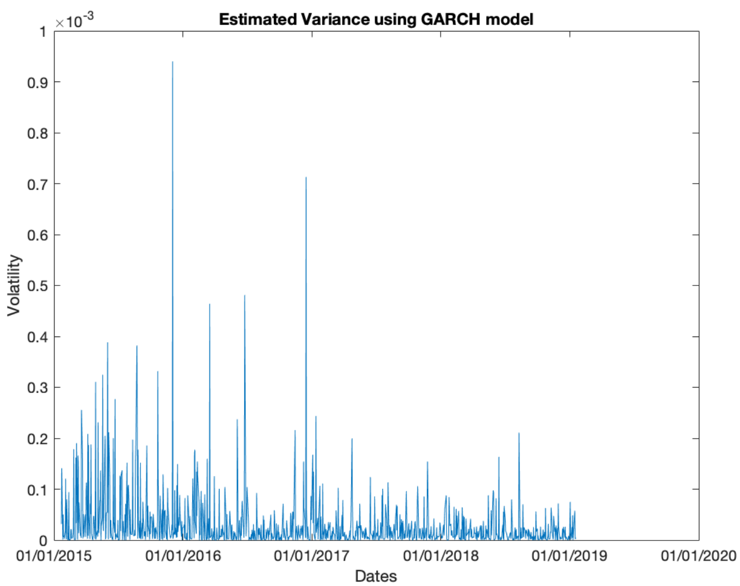

Figure 8: GARCH Variance Series of the USD/EUR Exchange Returns

Function garch_variance.m:

```
function GARCHvol = garch_variance(theta, data)

% this function generate variance (volatility) sig^2 series usign the
    GARCH model

% Input: theta (omega, beta, alpha); data, a de-meaned time series data
% which can used as residuals

% Output: variance series generated by the GARCH model


```

```
[T,~] = size(data); % this is the number of observations

sig0 = var(data); % set the sample variance as the initial value

GARCHvol = [sig0; zeros(T-1,1)]; % create the vector for volatility series

for t = 2:T

    GARCHvol(t) = theta(1)+theta(2) * GARCHvol(t-1) + theta(3) * data(t)
        ^2;

end

end
```

I compute the log-likelihood by using **GARCH_LL.m**:

```
function garch_llh = GARCH_LL(theta,data)

% this function computes the log-likelihood of variance series based on
    the
% GARCH (1,1) model of the variance

% Input: theta(omega, beta, alpha); data, demeaned returns which is the
% series of resiudals

% Output: the negative log-likelihood of data given theta

% compute the number of obs of our data
[T,~]=size(data);
% generate the sigma^2 (T*1)of the distribution of residuals
sig2 = garch_variance(theta, data);


% compute the log-likelihood of residuals, which assumed Gaussian(0,sig^2)
ll=0;
for t = 1: T
    ll =ll+log(1/sqrt(2*pi*sig2(t))*exp(-(data(t))^2/(2*sig2(t))));

end

% compute the negative log-likelihood
garch_llh = -ll;


end
```

Exercise 5

Question A

Since the return of our asset y_t can be written as

$$y_t = \mu_t + \epsilon_t \tag{2}$$

And we assume constant mean $\mu_t = \mu$. Then we can obtain the residual:

$$\epsilon_t = y_t - \mu \tag{3}$$

where μ can be approximated by the sample mean.

Question B

The GARCH(1,1) model is given by

$$\sigma_t^2 = \omega + \beta\sigma_{t-1}^2 + \alpha\epsilon_{t-1}^2 \tag{4}$$

Table 8: Estimated GARCH(1,1) Model Parameters

ω	β	α
1.7005×10^{-6}	0.86828	0.12731

Question C

The annualized standard deviaiton: $\sqrt{252 \times \sigma_t^2}$

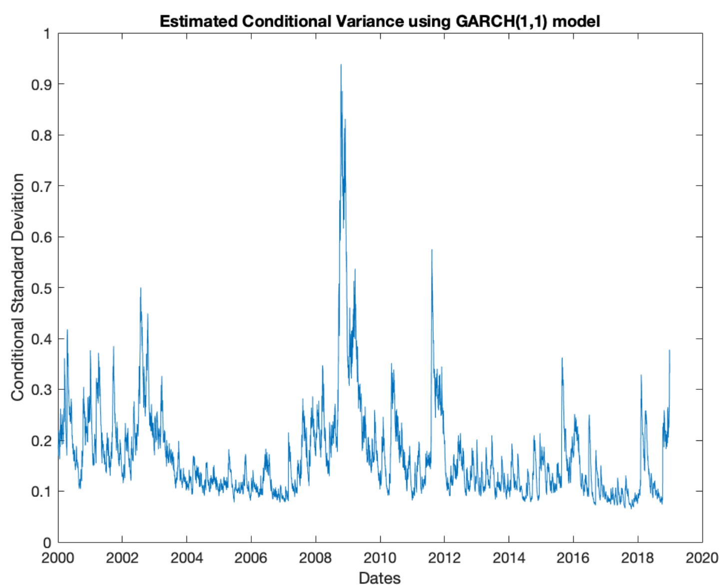

Figure 9: Estimated Conditional Volatility in Annualized Standard Deviation