金融与经济大数据挖掘作业——基于混合分布的EGARCH模型

16300200020 张言健

数据说明

低频日数据：沪深300-恒生指数-标准普尔

	时间	开盘价	最高价	最低价	收盘价
沪深-300	time	op_sh	high_sh	low_sh	cl_sh
恒生指数	time	op_hs	high_hs	low_hs	cl_hs
标准普尔	time	op_sp	high_sp	low_sp	cl_sp

高频5分钟数据：台湾指数期货与现货

日期时间	期货	现货
dateid	index_futures	index_spot

建立单变量的基于混合正态分布的AR-EGARCH模型

EGARCH模型是考虑杠杆效应GARCH模型，其基本形式为：

$r_t = \mu_t + \alpha_t$

$\alpha_t = \delta_t \varepsilon_t$, $\varepsilon_t \in N(0, \delta_t)$

$\ln\left(\sigma_t^2\right) = \alpha_0 + \sum_{i=1}^{p} \alpha_i \ln\left(\sigma_{t-i}^2\right) + \sum_{j=1}^{q} \beta_j g\left(\varepsilon_{t-j}\right)$

其中 $g\left(\varepsilon_t\right) = \begin{cases} (\theta+\gamma)\varepsilon_t - \gamma E\left|\varepsilon_t\right|, \varepsilon_t \geq 0 \\ (\theta-\gamma)\varepsilon_t - \gamma E\left|\varepsilon_t\right|, \varepsilon_t < 0 \end{cases}$

由于随机时间序列变量常具有厚尾特征，上述条件均值ε_t的处理可以服从正态分布外， 还可服从能够刻画厚尾性的混合正态分布。

混合正态分布的概率密度函数为

$\varepsilon_t \sim i.i.d. \quad MN(\xi, p) = \begin{cases} N\left(0, \sigma^2\right) & 1-p \\ N\left(0, \xi\sigma^2\right) & p \end{cases}$

其中, $0 < \xi < 1$, $\sigma^2 = (1-p+\xi p)^{-1}$, $Var(\varepsilon_t) = 1$

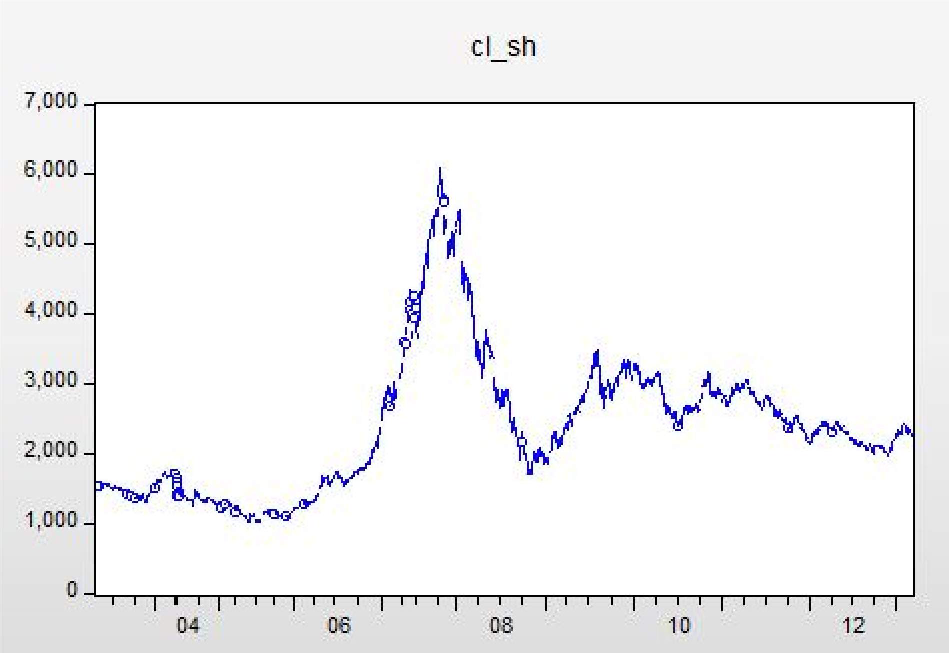
故 ε_t 的混合正态分布概率密度函数为

$f\left(\varepsilon_t\right) = \frac{p}{\sqrt{2\pi\xi\sigma^2}} \exp\left(-\frac{\varepsilon_t^2}{2\xi\sigma^2}\right) + \frac{1-p}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\varepsilon_t^2}{2\sigma^2}\right)$

低频数据以沪深100指数为例

1. 序列描述性分析

在视图中点击View-graph-line，得到如下图



2. 考察序列的平稳性

可以使用根检验来考察平稳性，点击View-Unit Root Test，Test Type选择Augmented Dickey-Fuller

Augmented Dickey-Fuller Unit Root Test on CL_SH

Null Hypothesis: CL_SH has a unit root
Exogenous: Constant
Lag Length: 0 (Automatic - based on SIC, maxlag=25)

		t-Statistic	Prob.*
Augmented Dickey-Fuller test statistic		-1.393755	0.5869
Test critical values:	1% level	-3.433218	
	5% level	-2.862693	
	10% level	-2.567430	

*MacKinnon (1996) one-sided p-values.

Augmented Dickey-Fuller Test Equation
Dependent Variable: D(CL_SH)
Method: Least Squares
Date: 05/01/19 Time: 22:28
Sample (adjusted): 4/29/2003 3/18/2013
Included observations: 2133 after adjustments

Variable	Coefficient	Std. Error	t-Statistic	Prob.
CL_SH(-1)	-0.001634	0.001172	-1.393755	0.1635
C	4.237215	3.035339	1.395961	0.1629

可以看出原始时间序列不平稳，需要对原始时间序列进行一阶差分后再检验

Augmented Dickey-Fuller Unit Root Test on D(CL_SH)

Null Hypothesis: D(CL_SH) has a unit root
Exogenous: Constant
Lag Length: 0 (Automatic - based on SIC, maxlag=25)

		t-Statistic	Prob.*
Augmented Dickey-Fuller test statistic		-47.85173	0.0001
Test critical values:	1% level	-3.433220	
	5% level	-2.862694	
	10% level	-2.567430	

*MacKinnon (1996) one-sided p-values.

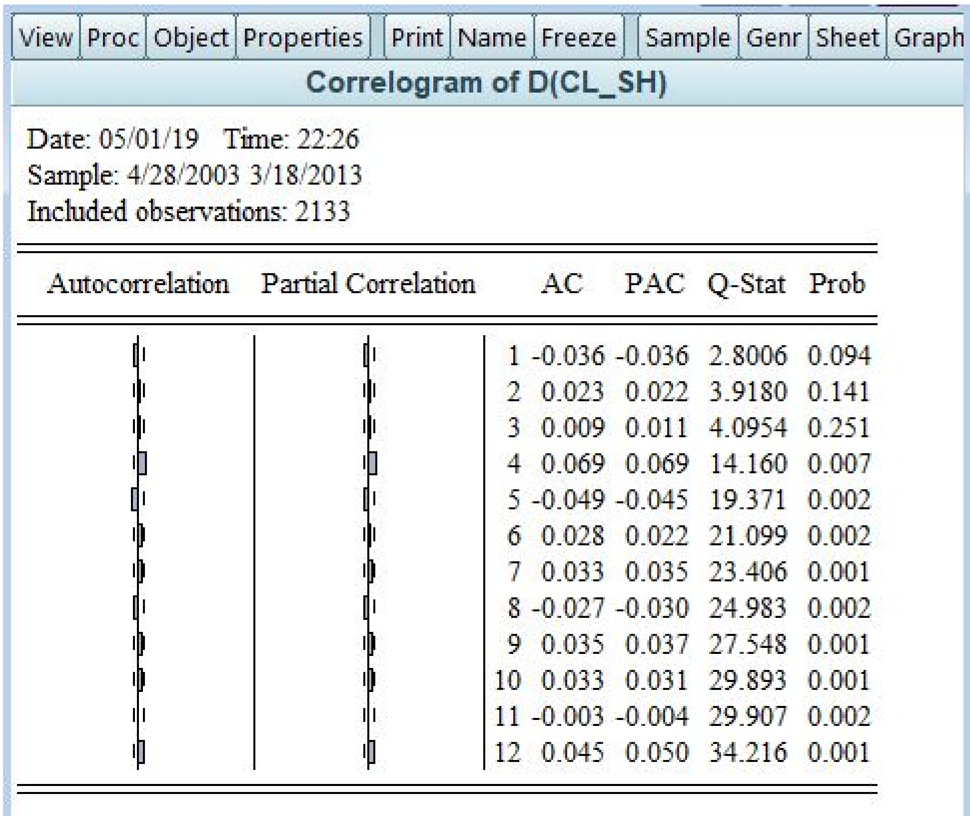
Augmented Dickey-Fuller Test Equation
Dependent Variable: D(CL_SH,2)
Method: Least Squares
Date: 05/01/19 Time: 22:29
Sample (adjusted): 4/30/2003 3/18/2013
Included observations: 2132 after adjustments

Variable	Coefficient	Std. Error	t-Statistic	Prob.
D(CL_SH(-1))	-1.036219	0.021655	-47.85173	0.0000
C	0.355346	1.162771	0.305603	0.7599

t统计量-47,85，对应P值接近0，可以看出此时时间序列已经平稳

3. 序列的自相关性和偏自相关性检验

在视图中点击View-correlogram，选择1st difference(我们刚刚检验过)，将滞后阶数Lags to include设置为12



View | Proc | Object | Properties | Print | Name | Freeze | Sample | Genr | Sheet | Graph

Correlogram of D(CL_SH)

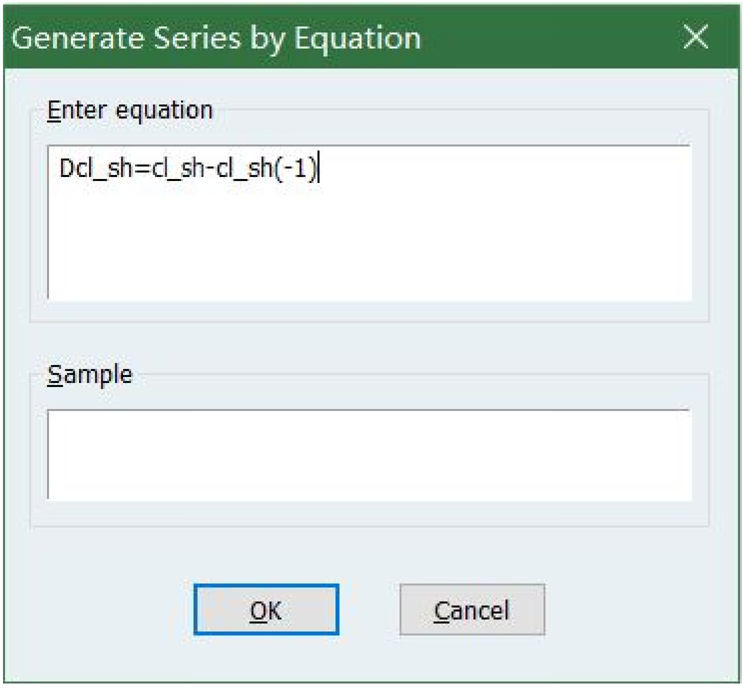
Date: 05/01/19 Time: 22:26
Sample: 4/28/2003 3/18/2013
Included observations: 2133

Autocorrelation	Partial Correlation		AC	PAC	Q-Stat	Prob
		1	-0.036	-0.036	2.8006	0.094
		2	0.023	0.022	3.9180	0.141
		3	0.009	0.011	4.0954	0.251
		4	0.069	0.069	14.160	0.007
		5	-0.049	-0.045	19.371	0.002
		6	0.028	0.022	21.099	0.002
		7	0.033	0.035	23.406	0.001
		8	-0.027	-0.030	24.983	0.002
		9	0.035	0.037	27.548	0.001
		10	0.033	0.031	29.893	0.001
		11	-0.003	-0.004	29.907	0.002
		12	0.045	0.050	34.216	0.001

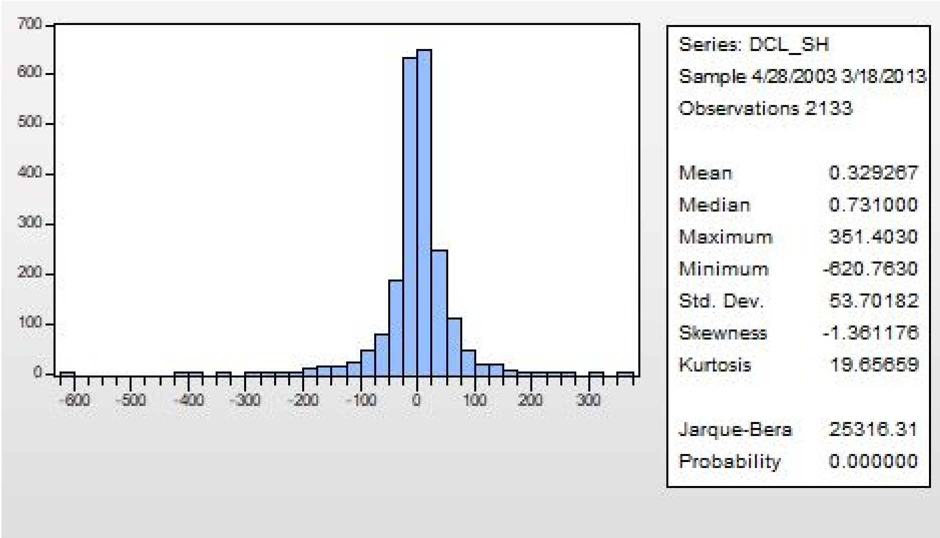
由于序列不存在显著的相关性，因此将均值方程设定为白噪声。

首先我们获得一阶差分的结果：

$d_t = r_t - r_{t-1}$，存入新的序列dcl_sh中，具体操作为Quick/Generate Series，输入如下表达式



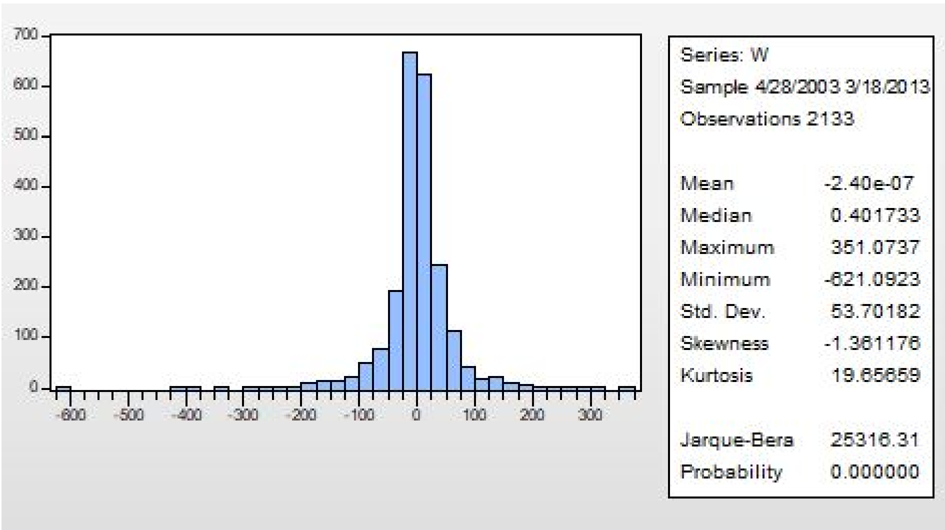
新序列dcl_sh的综合信息如下



设立模型 $d_t = \pi_t + \varepsilon_t$

CL_SH去均值化，得到w, 具体操作为Quick/Generate Series输入

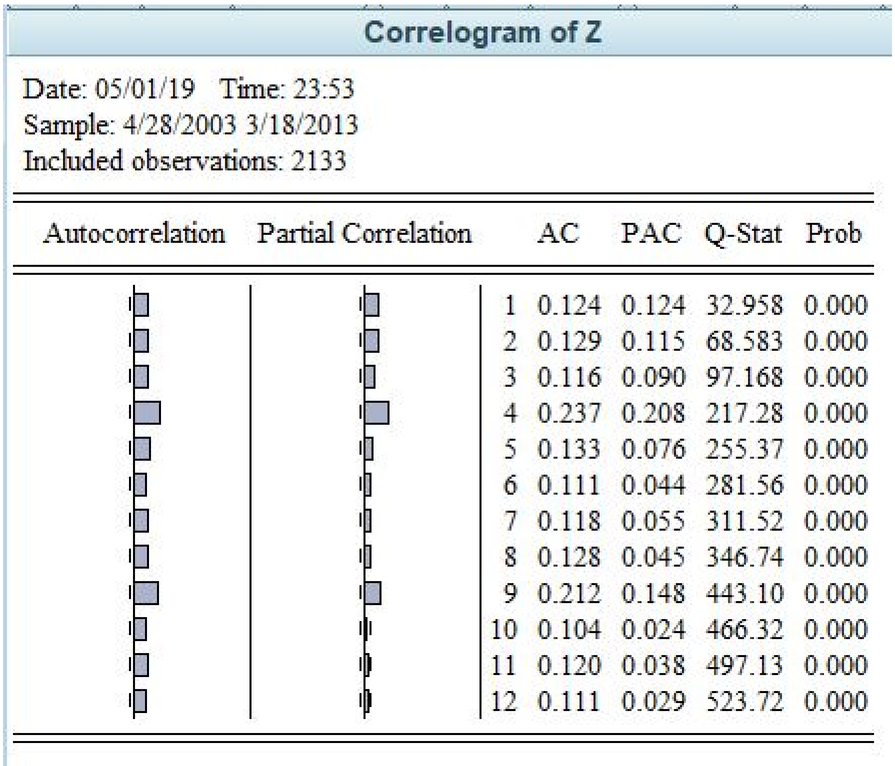
w=dcl_sh-0.329267，新序列w的综合信息如下



4. 检验ARCH效应

   检验ARCH效应有两种方法：LM法（拉格朗日乘数检验法）和对残差的平方相关图检验。在此我们采用第二种方法。

   首先建立w的平分方程z，在Quick/Generate Series输入z=w*w，然后在视图中点击view-correlogram，得到如下结果



Correlogram of Z

Date: 05/01/19 Time: 23:53
Sample: 4/28/2003 3/18/2013
Included observations: 2133

Autocorrelation	Partial Correlation		AC	PAC	Q-Stat	Prob
		1	0.124	0.124	32.958	0.000
		2	0.129	0.115	68.583	0.000
		3	0.116	0.090	97.168	0.000
		4	0.237	0.208	217.28	0.000
		5	0.133	0.076	255.37	0.000
		6	0.111	0.044	281.56	0.000
		7	0.118	0.055	311.52	0.000
		8	0.128	0.045	346.74	0.000
		9	0.212	0.148	443.10	0.000
		10	0.104	0.024	466.32	0.000
		11	0.120	0.038	497.13	0.000
		12	0.111	0.029	523.72	0.000

序列存在自相关，所以有ARCH效应。

5. 建立GARCH模型

   常用的GARCH模型包括GARCH(1,1)，GARCH(1,2)，GARCH(2,1)

   尝试GARCH(1,1), 使用Quick/Estimate Equation，设置Method为ARCH，Model为GARCH, Order中ARCH与GARCH填（1,1），可得结果如下

Dependent Variable: W
Method: ML ARCH - Normal distribution (BFGS / Marquardt steps)
Date: 05/02/19 Time: 00:02
Sample (adjusted): 4/29/2003 3/18/2013
Included observations: 2133 after adjustments
Convergence achieved after 25 iterations
Coefficient covariance computed using outer product of gradients
Presample variance: backcast (parameter = 0.7)
GARCH = C(1) + C(2)*RESID(-1)^2 + C(3)*GARCH(-1)

Variable	Coefficient	Std. Error	z-Statistic	Prob.
		Variance Equation		
C	3.381203	0.836777	4.040747	0.0001
RESID(-1)^2	0.060981	0.005054	12.06604	0.0000
GARCH(-1)	0.941036	0.004111	228.8886	0.0000

R-squared	0.000000	Mean dependent var	-2.40E-07
Adjusted R-squared	0.000469	S.D. dependent var	53.70182
S.E. of regression	53.68923	Akaike info criterion	10.01344
Sum squared resid	6148445.	Schwarz criterion	10.02141
Log likelihood	-10676.33	Hannan-Quinn criter.	10.01635
Durbin-Watson stat	2.072018		

所有系数都通过检验，因此GARCH（1,1）可以用于建模

6. 建立E-GARCH模型

建立EGARCH(1,1), 使用Quick/Estimate Equation，设置Method为ARCH，Model为EGARCH, 得到如下结果

Dependent Variable: W
Method: ML ARCH - Normal distribution (BFGS / Marquardt steps)
Date: 05/02/19 Time: 00:10
Sample (adjusted): 4/29/2003 3/18/2013
Included observations: 2133 after adjustments
Convergence achieved after 48 iterations
Coefficient covariance computed using outer product of gradients
Presample variance: backcast (parameter = 0.7)
LOG(GARCH) = C(1) + C(2)*ABS(RESID(-1)/@SQRT(GARCH(-1))) + C(3)*RESID(-1)/@SQRT(GARCH(-1)) + C(4)*LOG(GARCH(-1))

Variable	Coefficient	Std. Error	z-Statistic	Prob.
Variance Equation				
C(1)	-0.064463	0.008087	-7.971435	0.0000
C(2)	0.121921	0.009882	12.33729	0.0000
C(3)	0.022633	0.004801	4.713750	0.0000
C(4)	0.996678	0.000628	1585.823	0.0000

R-squared	0.000000	Mean dependent var	-2.40E-07
Adjusted R-squared	0.000469	S.D. dependent var	53.70182
S.E. of regression	53.68923	Akaike info criterion	10.01131
Sum squared resid	6148445.	Schwarz criterion	10.02193
Log likelihood	-10673.06	Hannan-Quinn criter.	10.01520
Durbin-Watson stat	2.072018		

EGARCH(1,1)模型的参数均显著，说明序列具有杠杆性。

7. 检验ARCH-M过程

进一步加入“ARCH-M”检验，配置时将ARCH-M设置为“std-dev”或者“variance”, “std-dev”结果如下

Dependent Variable: W
Method: ML ARCH - Normal distribution (BFGS / Marquardt steps)
Date: 05/02/19 Time: 00:14
Sample (adjusted): 4/29/2003 3/18/2013
Included observations: 2133 after adjustments
Convergence achieved after 41 iterations
Coefficient covariance computed using outer product of gradients
Presample variance: backcast (parameter = 0.7)
LOG(GARCH) = C(2) + C(3)*ABS(RESID(-1)/@SQRT(GARCH(-1))) + C(4)*RESID(-1)/@SQRT(GARCH(-1)) + C(5)*LOG(GARCH(-1))

Variable	Coefficient	Std. Error	z-Statistic	Prob.
@SQRT(GARCH)	0.001013	0.021130	0.047963	0.9617
Variance Equation				
C(2)	-0.064489	0.008311	-7.759352	0.0000
C(3)	0.121910	0.009879	12.34079	0.0000
C(4)	0.022729	0.005547	4.097548	0.0000
C(5)	0.996686	0.000724	1376.729	0.0000

R-squared	-0.000018	Mean dependent var	-2.40E-07
Adjusted R-squared	-0.000018	S.D. dependent var	53.70182
S.E. of regression	53.70231	Akaike info criterion	10.01225
Sum squared resid	6148555.	Schwarz criterion	10.02553
Log likelihood	-10673.06	Hannan-Quinn criter.	10.01711
Durbin-Watson stat	2.071978		

“variance”结果如下

Dependent Variable: W
Method: ML ARCH - Normal distribution (BFGS / Marquardt steps)
Date: 05/02/19 Time: 00:16
Sample (adjusted): 4/29/2003 3/18/2013
Included observations: 2133 after adjustments
Convergence achieved after 56 iterations
Coefficient covariance computed using outer product of gradients
Presample variance: backcast (parameter = 0.7)
LOG(GARCH) = C(2) + C(3)*ABS(RESID(-1)/@SQRT(GARCH(-1))) + C(4)*RESID(-1)/@SQRT(GARCH(-1)) + C(5)*LOG(GARCH(-1))

Variable	Coefficient	Std. Error	z-Statistic	Prob.
GARCH	-0.000243	0.000368	-0.659428	0.5096
Variance Equation				
C(2)	-0.063133	0.008794	-7.178878	0.0000
C(3)	0.121324	0.009898	12.25703	0.0000
C(4)	0.021884	0.005147	4.252156	0.0000
C(5)	0.996511	0.000856	1164.433	0.0000

R-squared	-0.000065	Mean dependent var	-2.40E-07
Adjusted R-squared	-0.000065	S.D. dependent var	53.70182
S.E. of regression	53.70356	Akaike info criterion	10.01201
Sum squared resid	6148843.	Schwarz criterion	10.02529
Log likelihood	-10672.81	Hannan-Quinn criter.	10.01687
Durbin-Watson stat	2.072008		

可以看出其结果均不显著，说明不存在ARCH-M过程

8. 模型验证

对建立的EARCH(1，1)模型进行残差ARCH效应检验，点击EARCH(1，1)结果输出窗口View /Residual Test /ARCH LM Test，将Lag依次设置为，1,4,8,12得到如下结果

Lag=1

Heteroskedasticity Test: ARCH

F-statistic	0.339287	Prob. F(1,2130)	0.5603
Obs*R-squared	0.339552	Prob. Chi-Square(1)	0.5601

Lag=4

Heteroskedasticity Test: ARCH

F-statistic	1.660160	Prob. F(4,2124)	0.1566
Obs*R-squared	6.635525	Prob. Chi-Square(4)	0.1564

Lag=8

Heteroskedasticity Test: ARCH

F-statistic	0.910193	Prob. F(8,2116)	0.5068
Obs*R-squared	7.287436	Prob. Chi-Square(8)	0.5060

Lag=12

Heteroskedasticity Test: ARCH

F-statistic	1.009127	Prob. F(12,2108)	0.4374
Obs*R-squared	12.11461	Prob. Chi-Square(12)	0.4365

在各种lag值情形下，F统计量均不显著，说明模型已经不存在ARCH效应。

因而我们最终得到的模型为

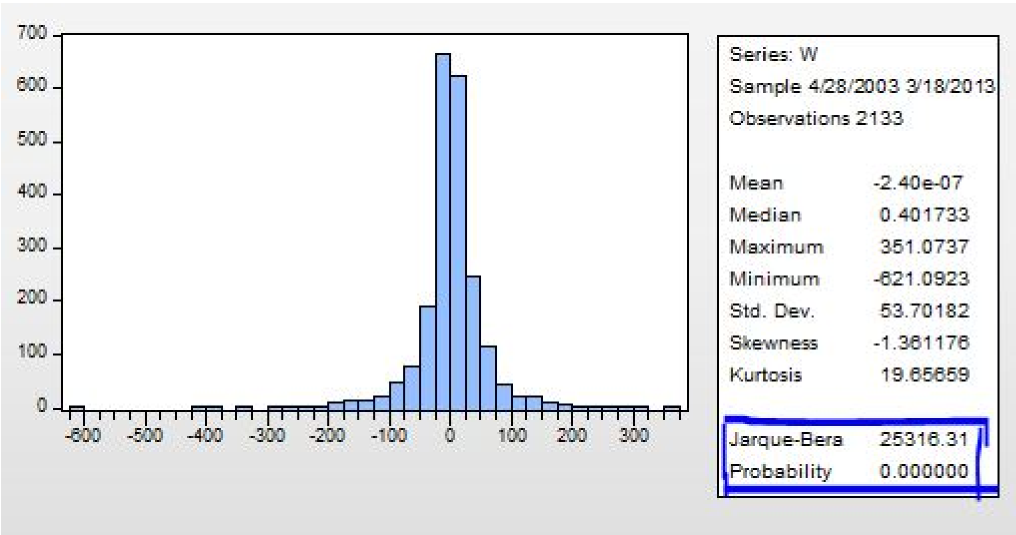
$$r_t - r_{t-1} = 0.329267 + a_t$$

$$a_t = \sigma_t \varepsilon_t, \quad \varepsilon_t \sim N(\mathrm{O}, \sigma_\varepsilon)$$

$$\ln \sigma_t^2 = -0.064 + 0.122 \left| \varepsilon_{t-1} / \sqrt{\sigma_{t-1}^2} \right| + 0.023 \varepsilon_{t-1} / \sqrt{\sigma_{t-1}^2} + 0.997^* \ln \sigma_{t-1}^2$$

9. 残差混合正态分布建模

现在我们回到 w的综合信息这里，发现Jarque-Bera 统计量过大，对应概率为0，因此统计的正态分布检验没有通过，为此我们需要对厚尾特征进行建模，此处我们采用混合正态分布。



我们将w导出到txt文件中，使用python进行建模, arch_model的参数dist传入误差分布的名字

```
from itertools import islice
file = open("./w.txt","r")
series = []
for line in islice(file,2,None):
    series.append(float(line.strip()))
# series
from scipy import  stats
import statsmodels.api as sm  # 统计相关的库
import numpy as np
import pandas as pd
```

```python
import matplotlib.pyplot as plt
import arch  # 条件异方差模型相关的库
am = arch.arch_model(series, mean='zero', p=1, o=1,dist='StudentsT', q=1,vol = 
"EGARCH")
print(am.distribution('StudentsT'))
res = am.fit()
```

由于arch_model中没有关于混合正太分布的误差分布，因此我们对arch_model所在的mean.py文件以及它所调用的distribution文件进行修改

其中mean.py修改如下

```python
# append
from arch.univariate.distribution import (GeneralizedError, Normal,
                                          SkewStudent, StudentsT, MixNormal)
# decorate
def arch_model(y, x=None, mean='Constant', lags=0, vol='Garch', p=1, o=0, 
q=1, p_ = None,power=2.0, dist='Normal', hold_back=None ):# p_
    if dist in ('skewstudent', 'skewt'):
        d = SkewStudent()
    elif dist in ('studentst', 't'):
        d = StudentsT()
    elif dist in ('ged', 'generalized error'):
        d = GeneralizedError()
    elif dist in ('mix', 'mix norm'):
        d = MixNormal(p_)
    else:  # ('gaussian', 'normal')
        d = Normal()
```

distribution.py修改如下

$$\sigma^2=(1-p+\xi p)^{-1}=>\xi=\frac{(\sigma^2)^{-1}-1}{p}+1=>\xi\sigma^2=\frac{1-\sigma^2}{p}+\sigma^2$$

```python
class MixNormal(Distribution):
    """
    Standard normal distribution for use with ARCH models
    """

    def __init__(self, p_ , random_state=None):
        super(MixNormal, self).__init__('MixNormal', random_state=random_state)
        self.name = 'MixNormal' # 修改名称
        self.p_ = p_

    def constraints(self):
        return empty(0), empty(0)

    def bounds(self, resids):
        return tuple([])

    def loglikelihood(self, parameters, resids, sigma2, individual=False):
        # newsigma2 为 乘以xi之后新的sigma2，公式见上
```

```
        newsigma2 = abs(((1-sigma2)/self.p_)+sigma2)
        ll1 = (log(self.p_) -0.5 * (log(2 * pi) + log(newsigma2) + resids ** 2.0 /
newsigma2))
        ll2 = (log(1-self.p_) -0.5 * (log(2 * pi) + log(sigma2) + resids ** 2.0 /
sigma2))
        lls = log(exp(ll1)+exp(ll2))
        if individual:
            return lls
        else:
            return sum(lls)

    def starting_values(self, std_resid):
        return empty(0)

    def _simulator(self, size):
        return self._random_state.standard_normal(size)

    def simulate(self, parameters):
        return self._simulator

    def parameter_names(self):
        return []

    def cdf(self, resids, parameters=None):#过程不涉及此处
        return None

    def ppf(self, pits, parameters=None): # 过程不涉及此处
        return None
```

定义混合正太分布后，使用以下命令调用混合正态模型

```
am = arch.arch_model(series, mean='zero', dist="mix", p=1, o=1, q=1, p_=0.5, vol =
"EGARCH")
```

```
                    Zero Mean - EGARCH Model Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.000
Mean Model:                 Zero Mean   Adj. R-squared:                  0.000
Vol Model:                     EGARCH   Log-Likelihood:               -10672.8
Distribution:               MixNormal   AIC:                           21353.6
Method:            Maximum Likelihood   BIC:                           21376.3
                                        No. Observations:                 2133
Date:                Sun, May 05 2019   Df Residuals:                     2129
Time:                        22:45:53   Df Model:                            4
                            Volatility Model
=============================================================================
                 coef    std err          t      P>|t|       95.0% Conf. Int.
-----------------------------------------------------------------------------
omega          0.0329  2.180e-02      1.510      0.131 [-9.804e-03,7.564e-02]
alpha[1]       0.1194  3.505e-02      3.406  6.603e-04    [5.067e-02,  0.188]
gamma[1]       0.0228  1.256e-02      1.818  6.911e-02 [-1.787e-03,4.747e-02]
beta[1]        0.9967  2.840e-03    350.988      0.000      [  0.991,  1.002]
=============================================================================
```

为了做对比，下面展示了正态分布所得结果

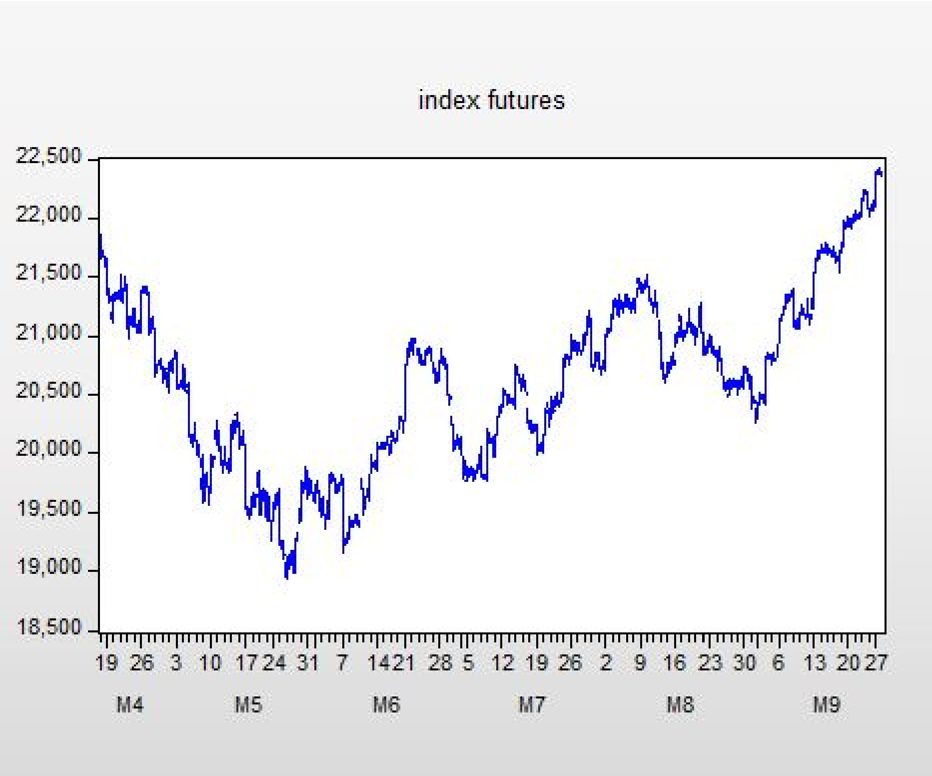
```
am = arch.arch_model(series, mean='zero', dist="mix", p=1, o=1, q=1, vol = "EGARCH")
```

```
                    Zero Mean - EGARCH Model Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.000
Mean Model:                 Zero Mean   Adj. R-squared:                  0.000
Vol Model:                     EGARCH   Log-Likelihood:               -10672.8
Distribution:                  Normal   AIC:                           21353.5
Method:            Maximum Likelihood   BIC:                           21376.2
                                        No. Observations:                 2133
Date:                Sun, May 05 2019   Df Residuals:                     2129
Time:                        22:53:49   Df Model:                            4
                            Volatility Model
=============================================================================
                 coef    std err          t      P>|t|       95.0% Conf. Int.
-----------------------------------------------------------------------------
omega          0.0324  2.170e-02      1.495      0.135 [-1.009e-02,7.497e-02]
alpha[1]       0.1194  3.503e-02      3.408  6.538e-04    [5.074e-02,  0.188]
gamma[1]       0.0229  1.257e-02      1.823  6.834e-02 [-1.724e-03,4.754e-02]
beta[1]        0.9967  2.829e-03    352.351      0.000      [  0.991,  1.002]
=============================================================================
```

）可以看到，混合正太分布的系数与正态分布的结果仅有微小差别，然而考虑了厚尾性的混合正态分布的结果t值更小（alpha、gamma、beta）。

高频数据以台湾指数期货为例

1. 序列描述性分析



2. 考察序列的平稳性，原序列不平稳，进行一阶差分，下图为一阶差分结果，一阶差分平稳

Augmented Dickey-Fuller Unit Root Test on D(INDEX_FUTURES)

Null Hypothesis: D(INDEX_FUTURES) has a unit root
Exogenous: Constant
Lag Length: 0 (Automatic - based on SIC, maxlag=33)

		t-Statistic	Prob.*
Augmented Dickey-Fuller test statistic		-76.42745	0.0001
Test critical values:	1% level	-3.431305	
	5% level	-2.861847	
	10% level	-2.566976	

*MacKinnon (1996) one-sided p-values.

Augmented Dickey-Fuller Test Equation
Dependent Variable: D(INDEX_FUTURES,2)
Method: Least Squares
Date: 05/05/19 Time: 23:09
Sample (adjusted): 4/16/2010 10:00 9/27/2010 15:50
Included observations: 5760 after adjustments

Variable	Coefficient	Std. Error	t-Statistic	Prob.
D(INDEX_FUTURES(-1))	-1.007003	0.013176	-76.42745	0.0000
C	0.096026	0.446887	0.214878	0.8299

3. 序列的自相关性和偏自相关性检验

Correlogram of D(INDEX_FUTURES)

Date: 05/05/19 Time: 23:11
Sample: 4/16/2010 09:50 9/27/2010 15:50
Included observations: 5761

Autocorrelation	Partial Correlation		AC	PAC	Q-Stat	Prob
		1	-0.007	-0.007	0.2827	0.595
		2	-0.026	-0.026	4.0761	0.130
		3	-0.013	-0.013	5.0276	0.170
		4	0.006	0.005	5.2023	0.267
		5	-0.002	-0.002	5.2202	0.390
		6	0.010	0.010	5.8283	0.443
		7	0.007	0.007	6.1102	0.527
		8	-0.016	-0.015	7.5924	0.474
		9	-0.004	-0.003	7.6698	0.568
		10	0.005	0.004	7.7877	0.650
		11	0.008	0.008	8.1847	0.697
		12	-0.000	0.000	8.1849	0.771

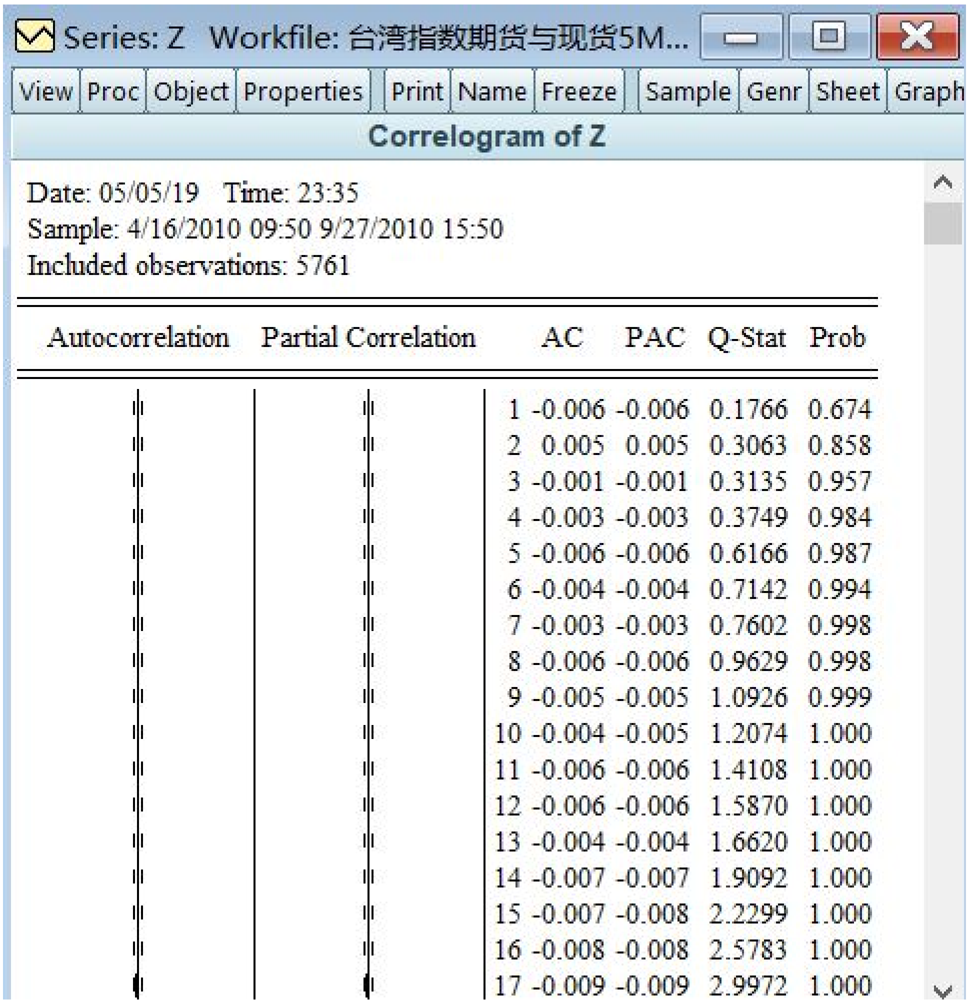
存在自相关性。

首先我们获得一阶差分的结果：

$d_t = r_t - r_{t-1}$, 存入新的序列dfuture中, 由于随机误差项之间有自相关性，因此不能进行去均值化

4. 检验ARCH效应

dfuture的平方的相关图检验： z=dfuture*dfuture


Series: Z Workfile: 台湾指数期货与现货5M...

View | Proc | Object | Properties | Print | Name | Freeze | Sample | Genr | Sheet | Graph

Correlogram of Z

Date: 05/05/19 Time: 23:35
Sample: 4/16/2010 09:50 9/27/2010 15:50
Included observations: 5761

Autocorrelation	Partial Correlation		AC	PAC	Q-Stat	Prob
		1	-0.006	-0.006	0.1766	0.674
		2	0.005	0.005	0.3063	0.858
		3	-0.001	-0.001	0.3135	0.957
		4	-0.003	-0.003	0.3749	0.984
		5	-0.006	-0.006	0.6166	0.987
		6	-0.004	-0.004	0.7142	0.994
		7	-0.003	-0.003	0.7602	0.998
		8	-0.006	-0.006	0.9629	0.998
		9	-0.005	-0.005	1.0926	0.999
		10	-0.004	-0.005	1.2074	1.000
		11	-0.006	-0.006	1.4108	1.000
		12	-0.006	-0.006	1.5870	1.000
		13	-0.004	-0.004	1.6620	1.000
		14	-0.007	-0.007	1.9092	1.000
		15	-0.007	-0.008	2.2299	1.000
		16	-0.008	-0.008	2.5783	1.000
		17	-0.009	-0.009	2.9972	1.000

不存在自相关性，没有ARCH效应，不适合建立ARCH类模型

综上，我们在低频数据沪深100指数上建立了基于混合分布的EGARCH模型，在高频数据台湾指数期货中，我们发现其没有ARCH效应，因此不再建立ARCH类模型。