时序分析(7) -- ARCH(p)模型 ¶

如无特殊说明，本系列文章中的数据将使用2012~2017年，分别代表国内股票、香港股票、国内债卷和国内货币的四个指数数据。

前几篇文章我们探讨的模型主要是为了对时序数据的观测值进行预测，开篇中我们曾经讲过分析时序波动性也是时序分析的一个重要目的，本篇文章开始介绍时序波动分析预测技术：ARCH系列模型。 在金融时序分析中，我们研究条件异方差性(conditional heteroskedasticity)来为资产收益的波动率建模。波动性在金融中是个非常非常重要的概念，因为它与风险模型的关系紧密。波动性在金融应用中主要有：

- 期权定价(Option Pricing)： Black-Scholes定价模型依赖于底层资产的波动性。
- 风险管理(Risk Management)： 波动性在计算投资组合的在险价值(VaR Value at Risk)和夏普比率中、确定杠杆比例中起着关键作用。
- 可交易证券： CBOE波动指数(VIX)、远期合同和ETF都可以通过波动率来交易。 如果我们可以很有效的预测波动率，我们就可以生成更为精确的定价模型和风险管理方法。

  首先我们介绍ARCH模型的基本概念：

Autoregressive Conditionally Heteroskedastic Models - ARCH(p)

- Heteroskedastic (异方差性)
  想象我们有一些随机变量，如果这些随机变量中存在某个组或者某个子集，其方差与剩下的随机变量的方差不同，我们就称这个随机变量集合是具备异方差性的。 举例来说，一个非平稳时序数据经常展现出趋势性和周期性，我们会发现时序的波动性随着周期或者趋势而增加，这种形式的波动性就称为异方差性。在金融中，有许多原因可以解释方差的增加与未来方差的增加是非常相关的。例如，只能做多的基金经理常会采用资产组合的下行风险保护策略，如果证卷市场遭遇大幅下跌时，会自动触发风险管理下的卖出下单，而这个操作会进一步使组合中的证券的价格下跌。因为大的投资组合一般都具备高度相关性，所以就会引发更大的下行波动。金融界的这些波动现象导致了串行相关的异方差性，且建立在方差增长期间的条件下的。我们称这样的时序是具备异方差性的。 鉴于刚才所描述的这戏特性，自回归异方差模型(Autoregressive Conditonally Heteroskedastic Model)就是用过于方差的变化来对方差进行建模。

ARCH模型可以被理解为AR(p)模型应用在时序的方差上，还可以理解为当前的方差是建立在过去的观测值的方差的条件上。

如果时序数据$\{y_t\}$可以表示为

$$y_t = \sigma_t w_t$$

其中$\{w_t\}$是高斯白噪声，均值为0，方差为1，这里σ_t为

$$\sigma_t^2 = \alpha_0 + \alpha_1 y_{t-1}^2$$

我们说时序$\{y_t\}$是一个一阶自回归异方差模型

$$\sigma_t = w_t \sqrt{\alpha_0 + \alpha_1 y_{t-1}^2}$$

$$Var(y_t) = \alpha_0 + \alpha_1 Var(y_{t-1})$$

那么什么样的时序适合ARCH(1)模型呢?

回想我们在判断AR(1)模型是否适合一个时序数据时，我们检查时序的一阶自相关性。同样的逻辑也适合于判断是否适合ARCH模型，只不过我们这时候是关注残差的平方。事实上，ARCH模型只适合于无趋势性和无周期性时序。

ARCH(p)模型:

$$y_t = w_t \sqrt{\alpha_0 + \sum_{i=1}^{p} \alpha_i y_{t-i}^2}$$

In [2]:

```
import warnings
warnings.simplefilter('ignore')
```

1. 导入python包

In [3]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

from fintechtools.backtest import *
from fintechtools.datasource import *
#from fintechtools.SimuMultiTest import *
#from lib.portfolio import DailySimulator
#from lib.experiment import Experiment

import statsmodels.formula.api as smf
import statsmodels.tsa.api as smt
import statsmodels.api as sm
import scipy.stats as scs
from arch import arch_model
#sns.set_context("talk")
import matplotlib
import matplotlib as mpl
from matplotlib.ticker import FuncFormatter
mpl.style.use('classic')

plt.rcParams['font.sans-serif'] = ['SimHei']
plt.rcParams['font.serif'] = ['SimHei']
plt.rcParams['axes.unicode_minus'] = False
import seaborn as sns
sns.set_style("whitegrid",{"font.sans-serif":['simhei', 'Arial']})
sns.set_context("talk")

#zhfont1 = matplotlib.font_manager.FontProperties(fname='C:\Users\ktwc37\Documents\ZNTG\notebooks\SI


%load_ext autoreload
%autoreload 2
```

```
The autoreload extension is already loaded. To reload it, use:
  %reload_ext autoreload
```

2. 读入数据

In [4]:

```
start = '2012-01-01'
end = '2017-02-05'
```

In [5]:

```
indexs = pd.read_excel('./data/华夏指数.xlsx')
indexs_pv = indexs.pivot_table(index='日期', columns='简称', values='收盘价(元)')
indexs_pv.index = pd.to_datetime(indexs_pv.index, unit='d')
```

In [6]:

```
indexs_pv.columns = ['国内债券', '国内股票', '香港股票', '国内货币']
indexs_pv = indexs_pv[['国内债券', '国内股票', '国内货币', '香港股票']]
indexs_pv.fillna(axis=0,method='bfill',inplace=True)
indexs_sub = indexs_pv.loc[start:end,]
```

国内债卷：中债综合财富(总值)指数
国内股票：中证全指
香港股票：恒生指数
国内货币：货币基金

In [7]:

```
indexs_sub.head()
```

Out[7]:

日期	国内债券	国内股票	国内货币	香港股票
2012-01-04	141.5160	2571.951	1166.7726	18727.31
2012-01-05	141.5501	2513.699	1166.9696	18813.41
2012-01-06	141.7277	2527.247	1167.1185	18593.06
2012-01-09	141.8669	2619.638	1167.5058	18865.72
2012-01-10	142.0118	2713.529	1167.6330	19004.28

In [8]:

```
indexs_logret = indexs_sub.apply(log_return).dropna()
```

In [9]:

```
indexs_logret.head()
```

Out[9]:

日期	国内债券	国内股票	国内货币	香港股票
2012-01-05	0.000241	-0.022909	0.000169	0.004587
2012-01-06	0.001254	0.005375	0.000128	-0.011782
2012-01-09	0.000982	0.035906	0.000332	0.014558
2012-01-10	0.001021	0.035214	0.000109	0.007318
2012-01-11	0.000188	-0.002115	0.000113	0.007740

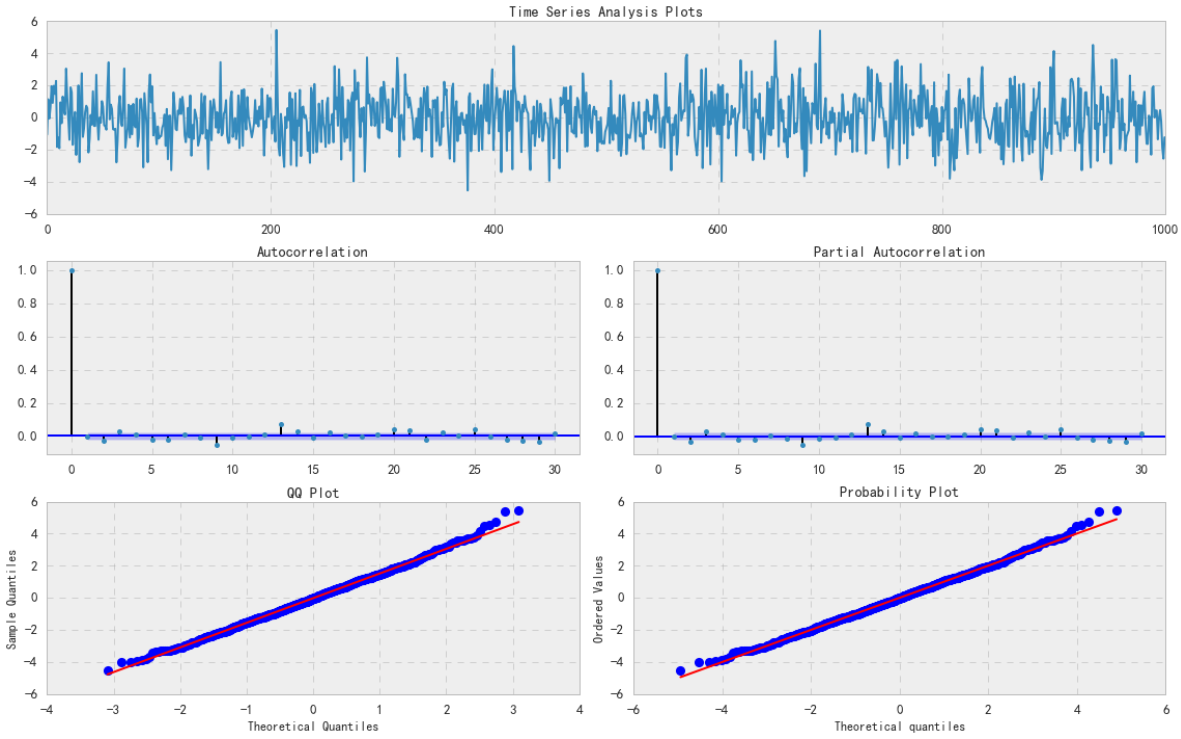
In [10]:

```
def tsplot(y, lags=None, figsize=(16, 10), style='bmh'):
    if not isinstance(y, pd.Series):
        y = pd.Series(y)
    with plt.style.context(style):
        fig = plt.figure(figsize=figsize)
        #mpl.rcParams['font.family'] = 'Ubuntu Mono'
        layout = (3, 2)
        ts_ax = plt.subplot2grid(layout, (0, 0), colspan=2)
        acf_ax = plt.subplot2grid(layout, (1, 0))
        pacf_ax = plt.subplot2grid(layout, (1, 1))
        qq_ax = plt.subplot2grid(layout, (2, 0))
        pp_ax = plt.subplot2grid(layout, (2, 1))

        y.plot(ax=ts_ax)
        ts_ax.set_title('Time Series Analysis Plots')
        smt.graphics.plot_acf(y, lags=lags, ax=acf_ax, alpha=0.5)
        smt.graphics.plot_pacf(y, lags=lags, ax=pacf_ax, alpha=0.5)
        sm.qqplot(y, line='s', ax=qq_ax)
        qq_ax.set_title('QQ Plot')
        scs.probplot(y, sparams=(y.mean(), y.std()), plot=pp_ax)

        plt.tight_layout()
    return
```

我们先模拟一个ARCH(1)时序

In [11]:

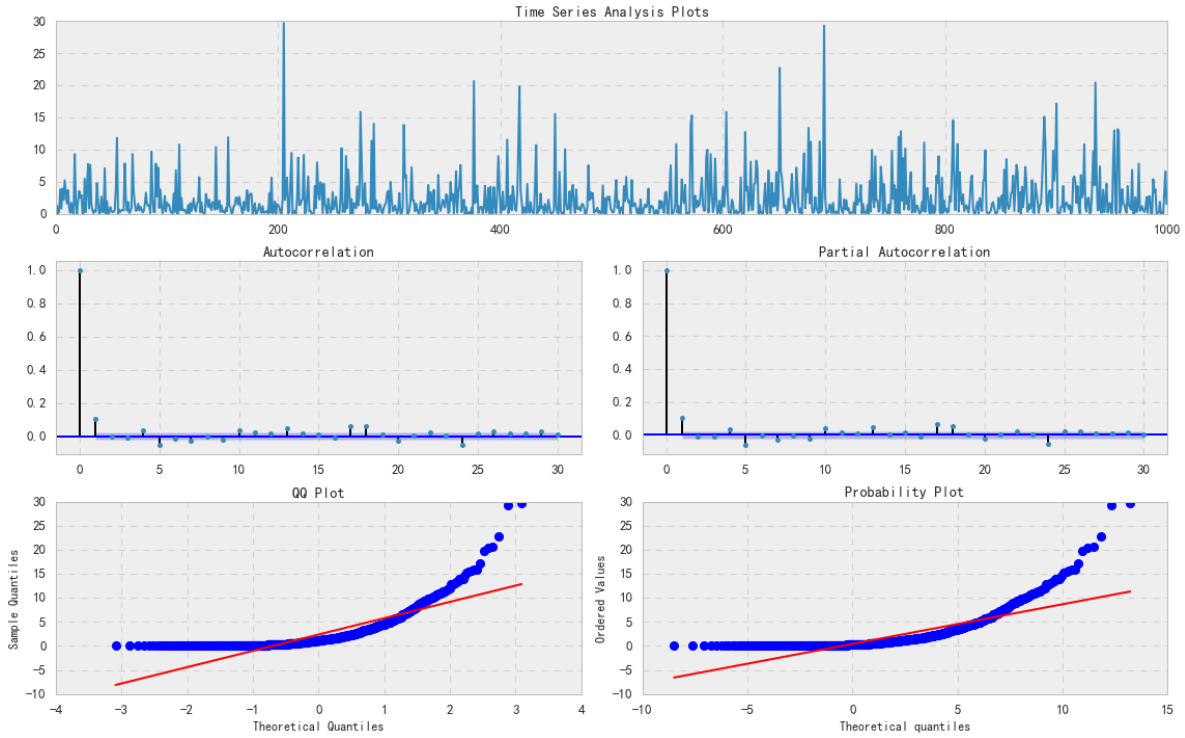
```
# Simulate ARCH(1) series
# Var(yt) = a_0 + a_1*y{t-1}**2
# if a_1 is between 0 and 1 then yt is white noise

np.random.seed(13)

a0 = 2
a1 = .5

y = w = np.random.normal(size=1000)
Y = np.empty_like(y)

for t in range(len(y)):
    Y[t] = w[t] * np.sqrt((a0 + a1*y[t-1]**2))

# simulated ARCH(1) series, looks like white noise
tsplot(Y, lags=30)
```



ARCH(1)时序的平方

In [13]:

```
tsplot(Y*Y, lags=30)
```



从ACF,PACF中显示一阶显著相关性，所有AR(1)很可能是适用于此时序数据的模型。

以T分布建模国内股票收益率

阶数为1

In [15]:

```
TS = indexs_logret['国内股票']
am = arch_model(TS, p=1, dist='StudentsT')
arch_gg = am.fit(update_freq=5, disp='off')
print(arch_gg.summary())
```

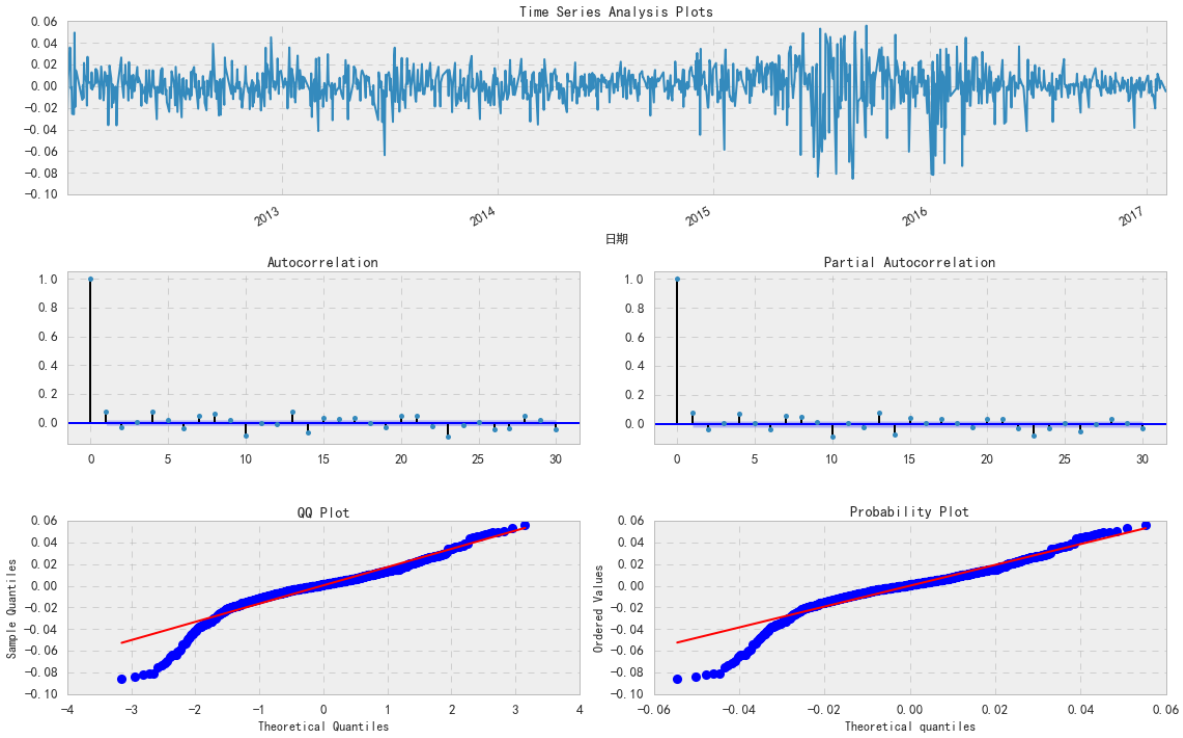
```
                     Constant Mean - GARCH Model Results
==============================================================================
Dep. Variable:                    国内股票   R-squared:                       -
0.000
Mean Model:             Constant Mean   Adj. R-squared:                 -0.000
Vol Model:                      GARCH   Log-Likelihood:                2448.22
Distribution:      Standardized Student's t   AIC:                    -4886.44
Method:                 Maximum Likelihood   BIC:                    -4860.85
                                             No. Observations:             1234
Date:                  Sat, Jul 14 2018   Df Residuals:                 1229
Time:                          14:57:58   Df Model:                        5
                              Mean Model
============================================================================
                 coef    std err          t      P>|t|       95.0% Conf. Int.
----------------------------------------------------------------------------
mu          1.2378e-04  8.052e-04      0.154      0.878 [-1.454e-03,1.702e-03]
                           Volatility Model
============================================================================
                 coef    std err          t      P>|t|       95.0% Conf. Int.
----------------------------------------------------------------------------
omega       2.7701e-03  1.023e-04     27.077 1.841e-161  [2.570e-03,2.971e-03]
alpha[1]    9.6721e-07      0.585  1.654e-06      1.000      [ -1.146,  1.146]
beta[1]     3.4966e-03  1.375e-02      0.254      0.799 [-2.346e-02,3.045e-02]
                            Distribution
============================================================================
                 coef    std err          t      P>|t|  95.0% Conf. Int.
----------------------------------------------------------------------------
nu            55.6821      0.569     97.919      0.000 [ 54.568, 56.797]
============================================================================

Covariance estimator: robust

WARNING: The optimizer did not indicate successful convergence. The message was
Positive directional derivative for linesearch. See convergence_flag.

d:\Anaconda3\lib\site-packages\arch\univariate\base.py:524: ConvergenceWarning:
The optimizer returned code 8. The message is:
Positive directional derivative for linesearch
See scipy.optimize.fmin_slsqp for code meaning.

  ConvergenceWarning)
```

α_1不具备显著性

残差Plot

In [16]:

```
_ = tsplot(arch_gg.resid, lags=30)
```



残差并非正态分布，且显示出串行相关性，说明模型拟合并不是很好。

以T分布建模香港股票收益率

阶数为1

In [17]:

```
TS = indexs_logret['香港股票']
am = arch_model(TS, p=1, dist='StudentsT')
arch_xg = am.fit(update_freq=5, disp='off')
print(arch_xg.summary())
```

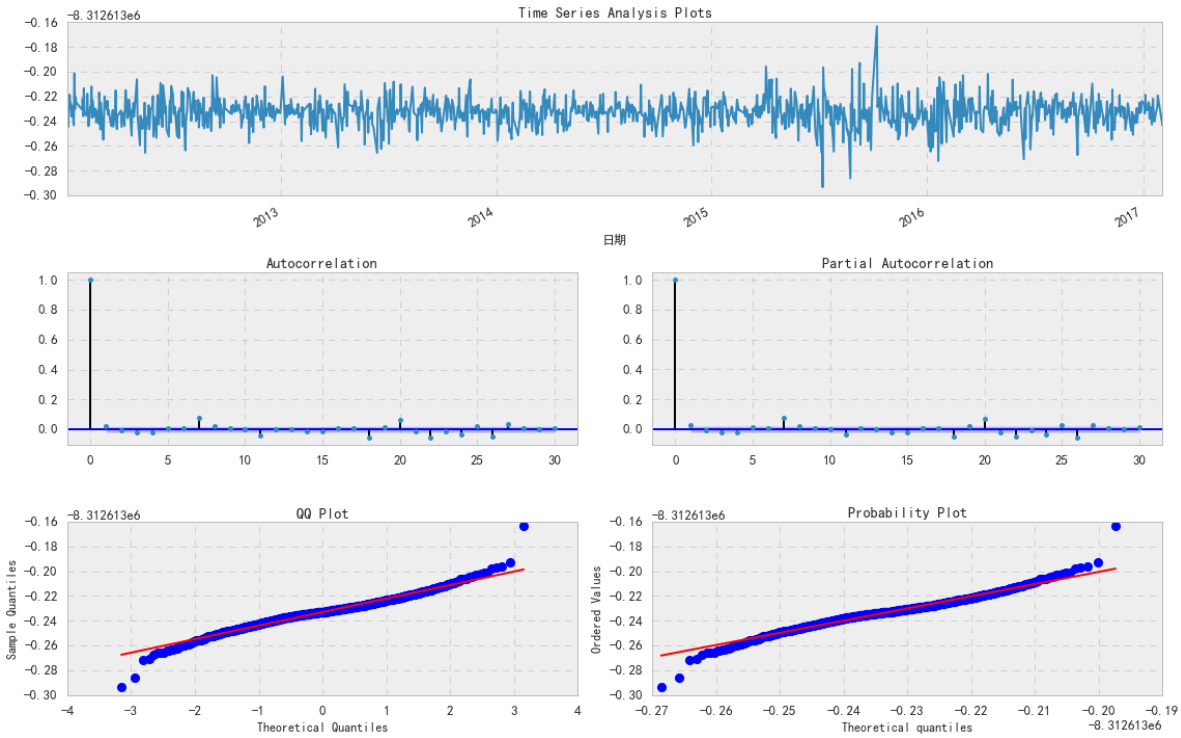
```
                       Constant Mean - GARCH Model Results

==============================================================================
=====
Dep. Variable:                          香港股票   R-squared:          -5783752803467
76448.000
Mean Model:                  Constant Mean   Adj. R-squared:     -57837528034677644
8.000
Vol Model:                           GARCH   Log-Likelihood:               -1.5551
9e+06
Distribution:      Standardized Student's t   AIC:                           3.1104
0e+06
Method:                  Maximum Likelihood   BIC:                           3.1104
2e+06
                                              No. Observations:
1234
Date:                     Sat, Jul 14 2018   Df Residuals:
1229
Time:                             17:52:27   Df Model:
    5
                                 Mean Model
==============================================================================
                 coef    std err          t      P>|t|      95.0% Conf. Int.
------------------------------------------------------------------------------
mu         8.3126e+06  1.329e+05     62.561      0.000 [8.052e+06,8.573e+06]
                           Volatility Model
==============================================================================
                 coef    std err          t      P>|t|       95.0% Conf. Int.
------------------------------------------------------------------------------
omega          0.0000  2.561e-04      0.000      1.000 [-5.020e-04,5.020e-04]
alpha[1]       0.9994  1.081e-02     92.469      0.000      [  0.978,   1.021]
beta[1]        0.0000      1.181      0.000      1.000      [ -2.314,   2.314]
                              Distribution
==============================================================================
                 coef    std err          t      P>|t|      95.0% Conf. Int.
------------------------------------------------------------------------------
nu           111.8346      1.797     62.226      0.000 [1.083e+02,1.154e+02]
==============================================================================

Covariance estimator: robust

WARNING: The optimizer did not indicate successful convergence. The message was
Positive directional derivative for linesearch. See convergence_flag.

d:\Anaconda3\lib\site-packages\arch\univariate\base.py:524: ConvergenceWarning:
The optimizer returned code 8. The message is:
Positive directional derivative for linesearch
See scipy.optimize.fmin_slsqp for code meaning.

  ConvergenceWarning)
```

α_1具备显著性

残差Plot

In [18]:

```
_ = tsplot(arch_xg.resid, lags=30)
```



模型拟合度要优于国内股票

总结

本文展示了采用Python语言为指数时序数据进行ARCH建模，介绍了异方差性、ARCH模型的基本概念。