时序分析(4) -- 移动平均模型(MA)

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。

```
        上一篇文章我们探讨了采用AR模型对时序数据进行建模，总的看来AR模型对金融序列建模并不是很理想。
这一节我们主要讨论时序数据移动平均模型。
   首先我们介绍MA模型的基本概念:
```

Moving Average Models - MA(q)

```
移动平均模型MA(q)其实和自回归模型有相似之处，不同之处在于移动平均是以过去的残差项也就是白噪声来做
线性组合，而AR模型是以过去的观察值来做线性组合。MA的出发点是通过组合残差项来观察残差的振动
MA(q)模型定义如下:
```

如果一个单变量时序数据$\{y_t; t = 0, 1, 2...\}$,

$$\begin{aligned} y_t &= \omega_t + \beta_1\omega_{t-1} + ... + \beta_p\omega_{t-p} \\ &= \omega_t + \sum_{i=1}^{p} \beta_i\omega_{t-i} \end{aligned}$$

In [16]:

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

1. 导入python包

In [4]:

```
import pandas as pd
import numpy as np
%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\S


%load_ext autoreload
%autoreload 2
```

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

2. 读入数据

In [5]:

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

In [6]:

```
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 [7]:

```
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 [8]:

```
indexs_sub.head()
```

Out[8]:

	国内债券	国内股票	国内货币	香港股票
日期				
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 [9]:

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

In [10]:

```
indexs_logret.head()
```

Out[10]:

	国内债券	国内股票	国内货币	香港股票
日期				
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 [11]:

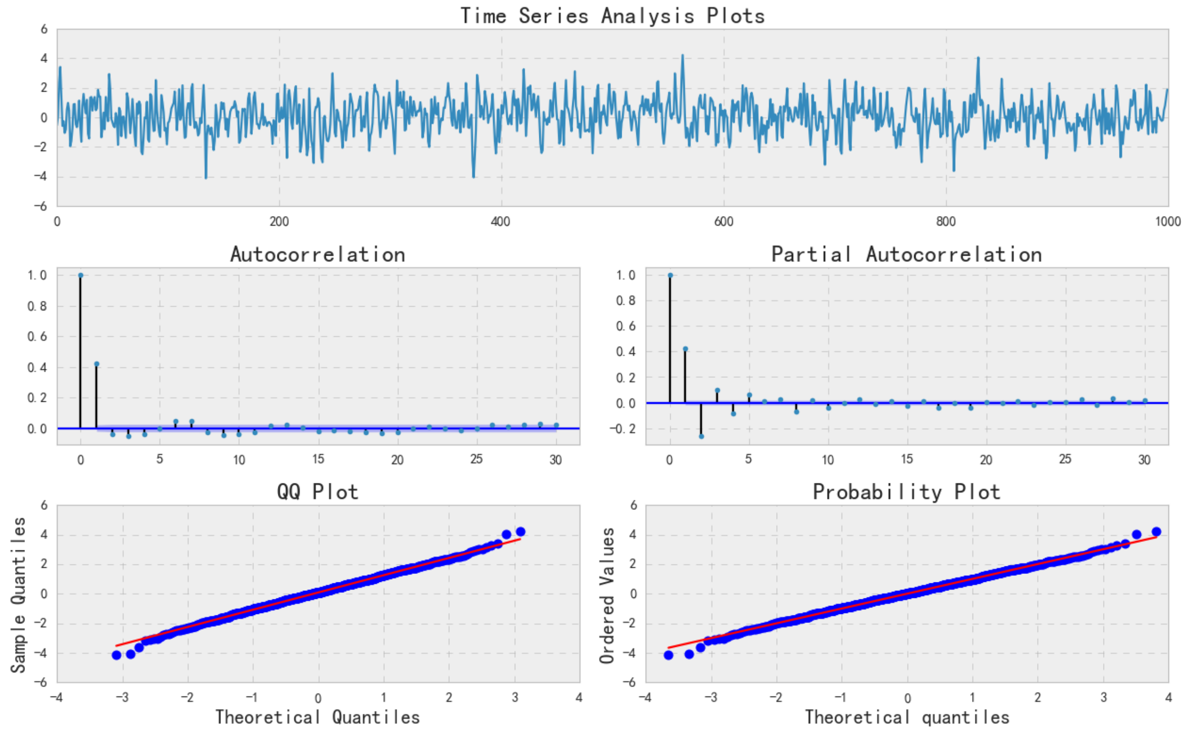
```
def tsplot(y, lags=None, figsize=(16, 10), style='bmh'):
    if not isinstance(y, pd.Series):
        y = pd.Series(y)
    with plt.style.context(style):
        fig = plt.figure(figsize=figsize)
        #mpl.rcParams['font.family'] = 'Ubuntu Mono'
        layout = (3, 2)
        ts_ax = plt.subplot2grid(layout,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

模拟MA(1)过程

下面我们以$\beta = 0.6$来模拟一个MA(1)过程

In [12]:

```
# Simulate an MA(1) process

n = int(1000)

# set the AR(p) alphas equal to 0
alphas = np.array([0.])
betas = np.array([0.6])

# add zero-lag and negate alphas
ar = np.r_[1, -alphas]
ma = np.r_[1, betas]

ma1 = smt.arma_generate_sample(ar=ar, ma=ma, nsample=n)
_ = tsplot(ma1, lags=30)
```



现在我们对此模拟数据用MA(3)进行建模

In [28]:

```
mdl_ma_sim = smt.ARMA(ma1, order=(0, 3)).fit(
    maxlag=max_lag, method='mle', trend='nc')
print(mdl_ma_sim.summary())
```

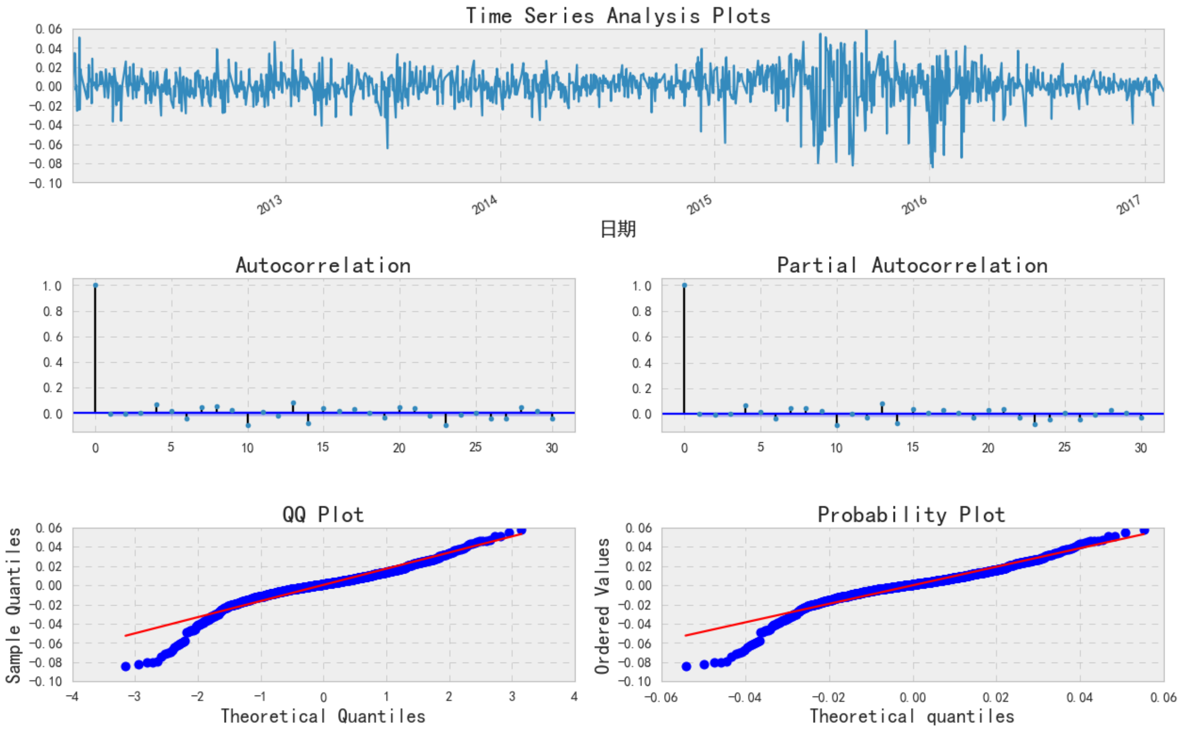
```
                              ARMA Model Results
==============================================================================
Dep. Variable:                      y   No. Observations:                 1000
Model:                     ARMA(0, 3)   Log Likelihood               -1429.380
Method:                           mle   S.D. of innovations              1.010
Date:                Wed, 11 Jul 2018   AIC                           2866.761
Time:                        20:02:03   BIC                           2886.392
Sample:                             0   HQIC                          2874.222

==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ma.L1.y        0.5822      0.032     18.344      0.000       0.520       0.644
ma.L2.y       -0.0220      0.038     -0.575      0.566      -0.097       0.053
ma.L3.y       -0.0143      0.032     -0.448      0.654      -0.077       0.048
                                    Roots
=============================================================================
                 Real           Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
MA.1          -1.7321           +0.0000j            1.7321            0.5000
MA.2           6.4467           +0.0000j            6.4467            0.0000
MA.3          -6.2511           +0.0000j            6.2511            0.5000
-----------------------------------------------------------------------------
```

- 国内股票收益率
  以MA建模，阶数设为3
  优化目标为Max Likelihood Estimation.

In [17]:

```
Y = indexs_logret['国内股票']
max_lag = 30
mdl_ma_gg = smt.ARMA(Y, order=(0, 3)).fit(
    maxlag=max_lag, method='mle', trend='nc')
print(mdl_ma_gg.summary())
_ = tsplot(mdl_ma_gg.resid, lags=max_lag)
```

```
                              ARMA Model Results
==============================================================================
Dep. Variable:                   国内股票   No. Observations:                 1234
Model:                     ARMA(0, 3)   Log Likelihood                3291.892
Method:                           mle   S.D. of innovations              0.017
Date:                Wed, 11 Jul 2018   AIC                          -6575.783
Time:                        19:41:35   BIC                          -6555.311
Sample:                    01-05-2012   HQIC                         -6568.082
                         - 02-03-2017
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ma.L1.国内股票     0.0806      0.029      2.797      0.005       0.024       0.137
ma.L2.国内股票    -0.0258      0.027     -0.964      0.335      -0.078       0.027
ma.L3.国内股票    -0.0066      0.030     -0.221      0.825      -0.066       0.052
                                    Roots
=============================================================================
                 Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
MA.1            4.8904           -0.0000j            4.8904           -0.0000
MA.2           -4.3865           -3.3955j            5.5471           -0.3952
MA.3           -4.3865           +3.3955j            5.5471            0.3952
-----------------------------------------------------------------------------
```



从拟合结果上看，只有一阶系数存在显著性。

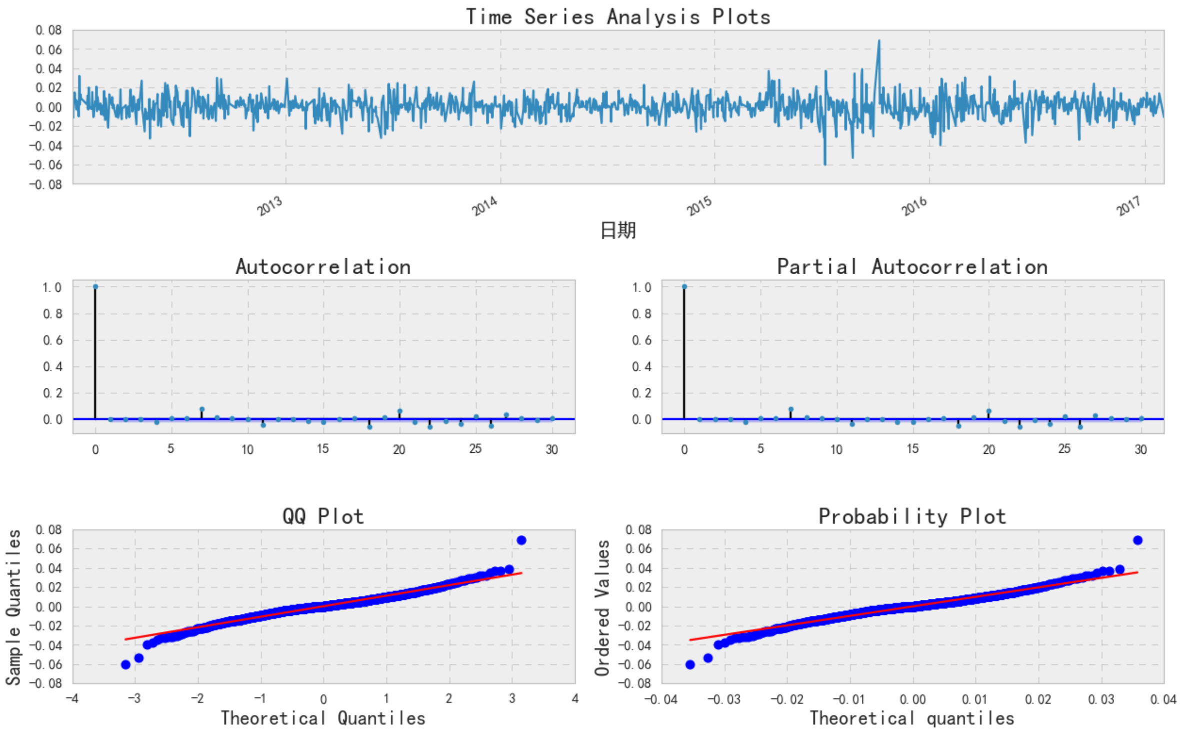
- 香港股票收益率
  以MA建模，阶数设为3

优化目标为Max Likelihood Estimation.

In [27]:

```
Y = indexs_logret['香港股票']
max_lag = 30
mdl_ma_gg = smt.ARMA(Y, order=(0, 3)).fit(
    maxlag=max_lag, method='mle', trend='nc')
print(mdl_ma_gg.summary())
_ = tsplot(mdl_ma_gg.resid, lags=max_lag)
```

```
                              ARMA Model Results
==============================================================================
Dep. Variable:                   香港股票   No. Observations:                 1234
Model:                     ARMA(0, 3)   Log Likelihood                3822.585
Method:                           mle   S.D. of innovations              0.011
Date:                Wed, 11 Jul 2018   AIC                          -7637.171
Time:                        19:58:12   BIC                          -7616.699
Sample:                    01-05-2012   HQIC                         -7629.470
                         - 02-03-2017
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ma.L1.香港股票     0.0242      0.028      0.849      0.396      -0.032       0.080
ma.L2.香港股票    -0.0057      0.029     -0.197      0.844      -0.062       0.051
ma.L3.香港股票    -0.0217      0.028     -0.763      0.445      -0.077       0.034
                                    Roots
=============================================================================
                 Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
MA.1          -1.9320          -3.0115j            3.5780           -0.3408
MA.2          -1.9320          +3.0115j            3.5780            0.3408
MA.3           3.6015          -0.0000j            3.6015           -0.0000
-----------------------------------------------------------------------------
```



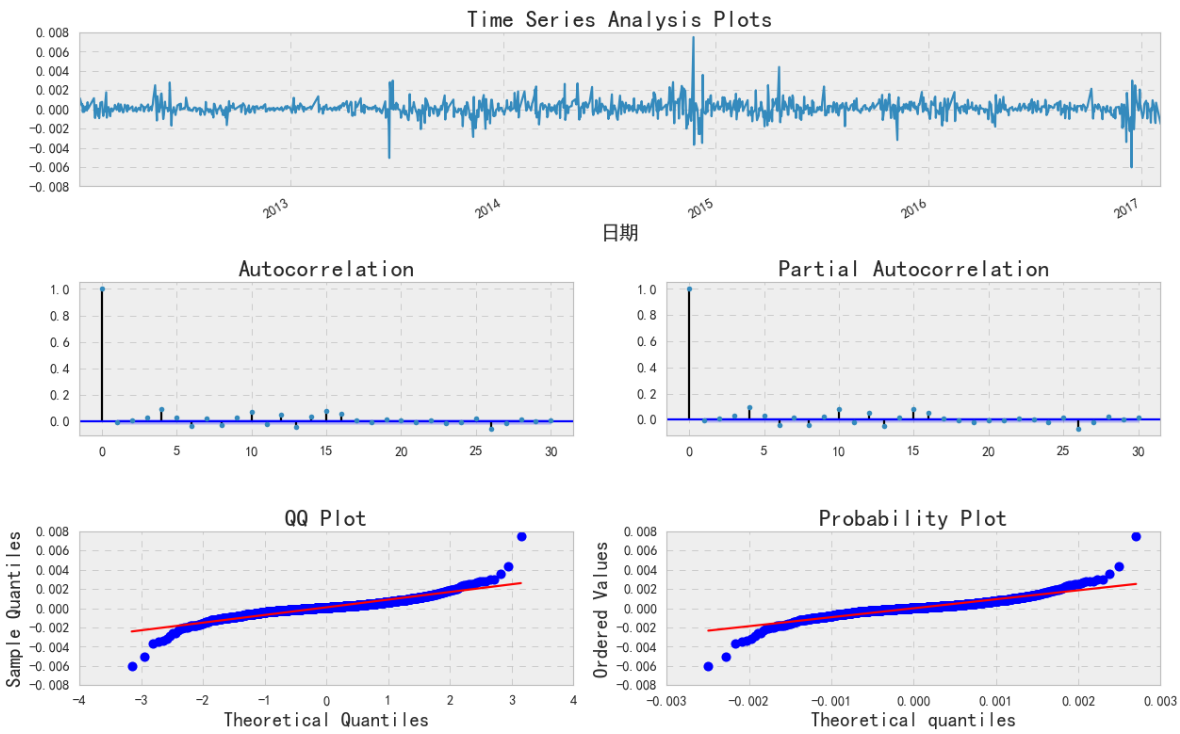
从第一阶到三阶系数都不存在显著性

- 国内债券收益率
  以MA建模，阶数设为3
  优化目标为Max Likelihood Estimation.

In [32]:

```
Y = indexs_logret['国内债券']
max_lag = 30
mdl_ma_gg = smt.ARMA(Y, order=(0, 3)).fit(
    maxlag=max_lag, method='mle', trend='nc')
print(mdl_ma_gg.summary())
_ = tsplot(mdl_ma_gg.resid, lags=max_lag)
```

```
                              ARMA Model Results
==============================================================================
Dep. Variable:                   国内债券   No. Observations:                 1234
Model:                     ARMA(0, 3)   Log Likelihood                7042.659
Method:                           mle   S.D. of innovations              0.001
Date:                Wed, 11 Jul 2018   AIC                         -14077.318
Time:                        20:05:25   BIC                         -14056.846
Sample:                    01-05-2012   HQIC                        -14069.617
                         - 02-03-2017
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ma.L1.国内债券     0.4182      0.029     14.659      0.000       0.362       0.474
ma.L2.国内债券     0.2185      0.028      7.933      0.000       0.165       0.273
ma.L3.国内债券     0.1122      0.028      4.003      0.000       0.057       0.167
                                    Roots
=============================================================================
                 Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
MA.1           0.0990           -2.0357j            2.0381           -0.2423
MA.2           0.0990           +2.0357j            2.0381            0.2423
MA.3          -2.1463           -0.0000j            2.1463           -0.5000
-----------------------------------------------------------------------------
```



总结

本文展示了采用Python语言为四个指数时序数据进行移动平均建模，介绍了MA模型等相关概念。