时序分析(6) -- ARIMA(p,d,q)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。

> 上一篇文章我们探讨了ARIMA模型时序数据进行建模，这一节我们主要讨论ARMA模型的一个非常重要的扩展模型ARIMA。

首先我们介绍ARIMA模型的基本概念:

Autoregressive Integrated Moving Average Models - ARIMA(p, d, q)

> 我们以前提到过金融时序大多数都不是平稳时序，也就是说其统计特性随着时间的推移而变化，但是通过差分我们有时可以将其变成平稳时序。在前面文章中我们看到了把指数数据变换成收益率后就是平稳过程了；把随机步行序列进行一阶差分以后就变成了白噪声。

ARIMA模型中的参数d就代表我们对原始序列进行差分的次数。

In [1]:

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

1. 导入python包

In [2]:

```
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 [3]:

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

In [4]:

```
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 [5]:

```
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 [6]:

```
indexs_sub.head()
```

Out[6]:

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 [7]:

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

In [8]:

```
indexs_logret.head()
```

Out[8]:

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 [9]:

```
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
```

下面我们将对四个指数数据进行ARIMA建模，注意：不是对收益率数据建模。

- 国内股票
  以ARIMA建模， 模型比较准则为AIC，得到阶数为(3,1,2)

In [26]:

```
best_aic = np.inf
best_order = None
best_mdl_gg = None
Y = indexs_sub['国内股票']
pq_rng = range(5) # [0,1,2,3,4]
d_rng = range(2) # [0,1]
for i in pq_rng:
    for d in d_rng:
        for j in pq_rng:
            try:
                tmp_mdl = smt.ARIMA(Y, order=(i,d,j)).fit(method='mle', trend='nc')
                tmp_aic = tmp_mdl.aic
                if tmp_aic < best_aic:
                    best_aic = tmp_aic
                    best_order = (i, d, j)
                    best_mdl_gg = tmp_mdl
            except: continue

print('aic: {:6.5f} | order: {}'.format(best_aic, best_order))
```

```
d:\Anaconda3\lib\site-packages\statsmodels\base\model.py:496: ConvergenceWarning: Ma
ximum Likelihood optimization failed to converge. Check mle_retvals
  "Check mle_retvals", ConvergenceWarning)
d:\Anaconda3\lib\site-packages\statsmodels\base\model.py:496: ConvergenceWarning: Ma
ximum Likelihood optimization failed to converge. Check mle_retvals
  "Check mle_retvals", ConvergenceWarning)
d:\Anaconda3\lib\site-packages\statsmodels\base\model.py:496: ConvergenceWarning: Ma
ximum Likelihood optimization failed to converge. Check mle_retvals
  "Check mle_retvals", ConvergenceWarning)

aic: 14191.66013 | order: (3, 1, 2)

d:\Anaconda3\lib\site-packages\statsmodels\base\model.py:496: ConvergenceWarning: Ma
ximum Likelihood optimization failed to converge. Check mle_retvals
  "Check mle_retvals", ConvergenceWarning)
```

得到p,d,q分别为3,1,2，这是符合我们的预期的。

In [27]:

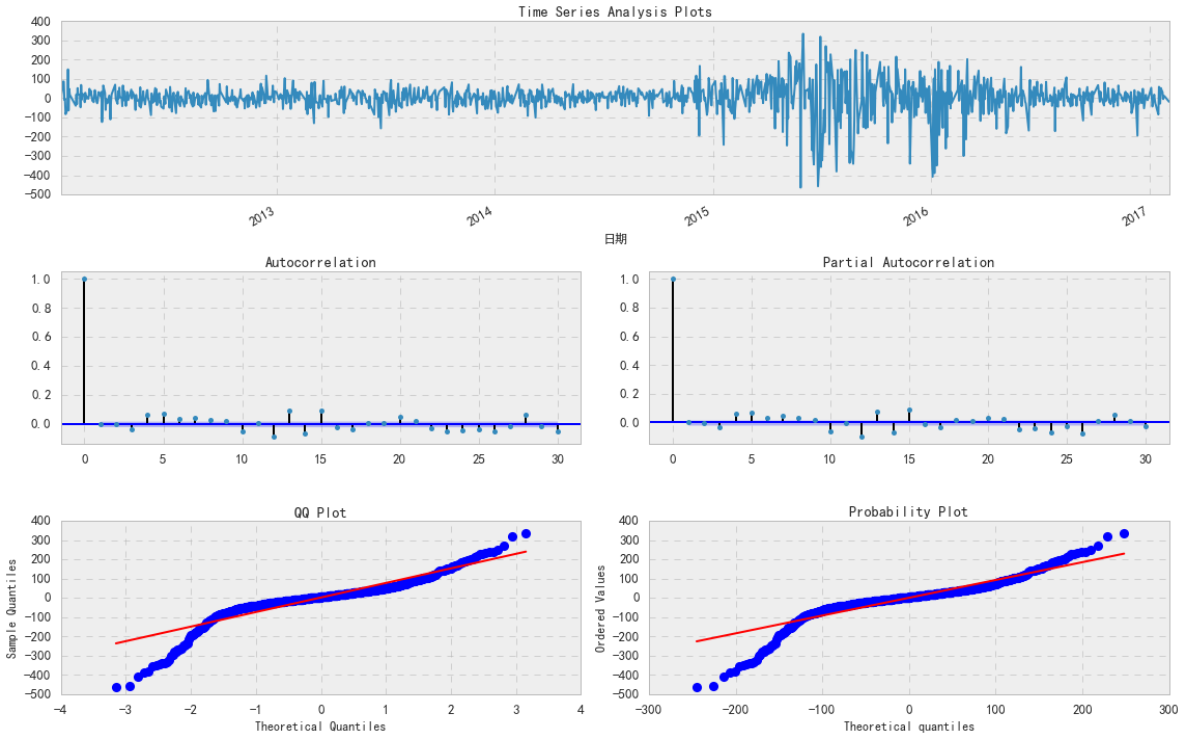
```
print(best_mdl_gg.summary())
```

```
                             ARIMA Model Results
==============================================================================
Dep. Variable:                 D.国内股票   No. Observations:                 1234
Model:                 ARIMA(3, 1, 2)   Log Likelihood               -7089.830
Method:                           mle   S.D. of innovations             75.642
Date:                Thu, 12 Jul 2018   AIC                          14191.660
Time:                        20:48:31   BIC                          14222.368
Sample:                    01-05-2012   HQIC                         14203.212
                         - 02-03-2017
=================================================================================
                    coef    std err          z      P>|z|      [0.025      0.975]
---------------------------------------------------------------------------------
ar.L1.D.国内股票     0.3577      0.030     11.910      0.000       0.299       0.417
ar.L2.D.国内股票    -0.9978      0.012    -85.108      0.000      -1.021      -0.975
ar.L3.D.国内股票     0.1832      0.029      6.390      0.000       0.127       0.239
ma.L1.D.国内股票    -0.2333      0.012    -18.782      0.000      -0.258      -0.209
ma.L2.D.国内股票     0.9558      0.016     58.997      0.000       0.924       0.988
                                    Roots
=============================================================================
                  Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
AR.1            0.0870           -1.0138j            1.0175           -0.2364
AR.2            0.0870           +1.0138j            1.0175            0.2364
AR.3            5.2732           -0.0000j            5.2732           -0.0000
MA.1            0.1220           -1.0156j            1.0229           -0.2310
MA.2            0.1220           +1.0156j            1.0229            0.2310
-----------------------------------------------------------------------------
```

残差Plot

In [28]:

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



从QQ-plot上看，残差并非正态分布。

- 香港股票
  以ARIMA建模
  比较准则为AIC，拟合参数为(3,1,2)

In [18]:

```
best_aic = np.inf
best_order = None
best_mdl = None
Y = indexs_sub['香港股票']
pq_rng = range(5) # [0,1,2,3,4]
d_rng = range(2) # [0,1]
for i in pq_rng:
    for d in d_rng:
        for j in pq_rng:
            try:
                tmp_mdl = smt.ARIMA(Y, order=(i,d,j)).fit(method='mle', trend='nc')
                tmp_aic = tmp_mdl.aic
                if tmp_aic < best_aic:
                    best_aic = tmp_aic
                    best_order = (i, d, j)
                    best_mdl = tmp_mdl
            except: continue


print('aic: {:6.5f} | order: {}'.format(best_aic, best_order))
```

```
d:\Anaconda3\lib\site-packages\statsmodels\base\model.py:496: ConvergenceWarning: Ma
ximum Likelihood optimization failed to converge. Check mle_retvals
  "Check mle_retvals", ConvergenceWarning)
```

```
aic: 17054.58001 | order: (3, 1, 2)
```

```
d:\Anaconda3\lib\site-packages\statsmodels\base\model.py:496: ConvergenceWarning: Ma
ximum Likelihood optimization failed to converge. Check mle_retvals
  "Check mle_retvals", ConvergenceWarning)
```

In [19]:

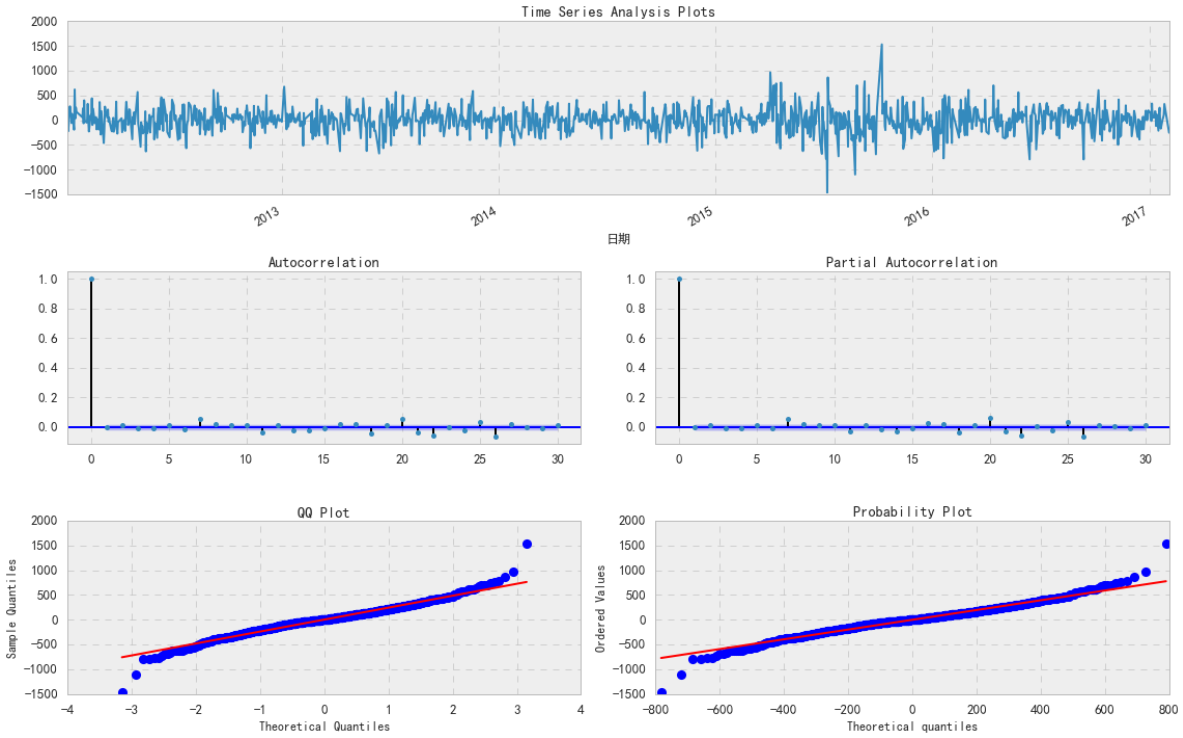
```
print(best_mdl.summary())
```

```
                             ARIMA Model Results
==============================================================================
Dep. Variable:                 D.香港股票   No. Observations:                 1234
Model:                 ARIMA(3, 1, 2)   Log Likelihood               -8521.290
Method:                           mle   S.D. of innovations            241.361
Date:                Thu, 12 Jul 2018   AIC                          17054.580
Time:                        20:26:31   BIC                          17085.288
Sample:                    01-05-2012   HQIC                         17066.132
                         - 02-03-2017
==================================================================================
                     coef    std err          z      P>|z|      [0.025      0.975]
----------------------------------------------------------------------------------
ar.L1.D.香港股票     1.2614      0.032     38.914      0.000       1.198       1.325
ar.L2.D.香港股票    -1.0232      0.037    -27.868      0.000      -1.095      -0.951
ar.L3.D.香港股票     0.0264      0.029      0.895      0.371      -0.031       0.084
ma.L1.D.香港股票    -1.2381      0.015    -79.963      0.000      -1.268      -1.208
ma.L2.D.香港股票     0.9824      0.011     91.504      0.000       0.961       1.003
                                    Roots
=============================================================================
                  Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
AR.1            0.6234           -0.7881j            1.0049           -0.1435
AR.2            0.6234           +0.7881j            1.0049            0.1435
AR.3           37.5841           -0.0000j           37.5841           -0.0000
MA.1            0.6301           -0.7879j            1.0089           -0.1426
MA.2            0.6301           +0.7879j            1.0089            0.1426
-----------------------------------------------------------------------------
```

残差Plot

In [20]:

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



- 国内债券 以ARIMA建模
  比较准则AIC，得到(3,0,1).

In [21]:

```
Y = indexs_logret['国内债券']
best_aic = np.inf
best_order = None
best_mdl = None
pq_rng = range(5) # [0,1,2,3,4]
d_rng = range(2) # [0,1]
for i in pq_rng:
    for d in d_rng:
        for j in pq_rng:
            try:
                tmp_mdl = smt.ARIMA(Y, order=(i,d,j)).fit(method='mle', trend='nc')
                tmp_aic = tmp_mdl.aic
                if tmp_aic < best_aic:
                    best_aic = tmp_aic
                    best_order = (i, d, j)
                    best_mdl = tmp_mdl
            except: continue


print('aic: {:6.5f} | order: {}'.format(best_aic, best_order))
```

```
d:\Anaconda3\lib\site-packages\statsmodels\base\model.py:496: ConvergenceWarning: Ma
ximum Likelihood optimization failed to converge. Check mle_retvals
  "Check mle_retvals", ConvergenceWarning)
d:\Anaconda3\lib\site-packages\statsmodels\base\model.py:496: ConvergenceWarning: Ma
ximum Likelihood optimization failed to converge. Check mle_retvals
  "Check mle_retvals", ConvergenceWarning)
d:\Anaconda3\lib\site-packages\statsmodels\base\model.py:496: ConvergenceWarning: Ma
ximum Likelihood optimization failed to converge. Check mle_retvals
  "Check mle_retvals", ConvergenceWarning)
```

```
aic: -14105.28291 | order: (3, 0, 1)
```

In [22]:

```
print(best_mdl.summary())
```

```
                              ARMA Model Results
==============================================================================
Dep. Variable:                   国内债券   No. Observations:                 1234
Model:                     ARMA(3, 1)   Log Likelihood                7057.641
Method:                           mle   S.D. of innovations              0.001
Date:                Thu, 12 Jul 2018   AIC                         -14105.283
Time:                        20:30:50   BIC                         -14079.693
Sample:                    01-05-2012   HQIC                        -14095.657
                         - 02-03-2017
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1.国内债券     1.3945      0.031     44.862      0.000       1.334       1.455
ar.L2.国内债券    -0.3340      0.048     -6.961      0.000      -0.428      -0.240
ar.L3.国内债券    -0.0629      0.029     -2.138      0.033      -0.121      -0.005
ma.L1.国内债券    -0.9845      0.012    -79.206      0.000      -1.009      -0.960
                                    Roots
=============================================================================
                  Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
AR.1            1.0045           +0.0000j            1.0045            0.0000
AR.2            1.9219           +0.0000j            1.9219            0.0000
AR.3           -8.2389           +0.0000j            8.2389            0.5000
MA.1            1.0157           +0.0000j            1.0157            0.0000
-----------------------------------------------------------------------------
```

- 国内货币
  以ARIMA建模
  比较准则AIC，得到(4,1,4).

In [23]:

```
Y = indexs_logret['国内货币']
best_aic = np.inf
best_order = None
best_mdl = None
pq_rng = range(5) # [0,1,2,3,4]
d_rng = range(2) # [0,1]
for i in pq_rng:
    for d in d_rng:
        for j in pq_rng:
            try:
                tmp_mdl = smt.ARIMA(Y, order=(i,d,j)).fit(method='mle', trend='nc')
                tmp_aic = tmp_mdl.aic
                if tmp_aic < best_aic:
                    best_aic = tmp_aic
                    best_order = (i, d, j)
                    best_mdl = tmp_mdl
            except: continue


print('aic: {:6.5f} | order: {}'.format(best_aic, best_order))
```

```
d:\Anaconda3\lib\site-packages\statsmodels\base\model.py:496: ConvergenceWarning: Ma
ximum Likelihood optimization failed to converge. Check mle_retvals
  "Check mle_retvals", ConvergenceWarning)
d:\Anaconda3\lib\site-packages\statsmodels\base\model.py:496: ConvergenceWarning: Ma
ximum Likelihood optimization failed to converge. Check mle_retvals
  "Check mle_retvals", ConvergenceWarning)
d:\Anaconda3\lib\site-packages\statsmodels\base\model.py:496: ConvergenceWarning: Ma
ximum Likelihood optimization failed to converge. Check mle_retvals
  "Check mle_retvals", ConvergenceWarning)

aic: -19130.92496 | order: (4, 1, 4)
```

In [24]:

```
print(best_mdl.summary())
```

```
                             ARIMA Model Results
==============================================================================
Dep. Variable:                 D.国内货币   No. Observations:                 1233
Model:                 ARIMA(4, 1, 4)   Log Likelihood                9574.462
Method:                           mle   S.D. of innovations              0.000
Date:                Thu, 12 Jul 2018   AIC                         -19130.925
Time:                        20:32:52   BIC                         -19084.870
Sample:                    01-06-2012   HQIC                        -19113.600
                         - 02-03-2017
==================================================================================
                     coef    std err          z      P>|z|      [0.025      0.975]
----------------------------------------------------------------------------------
ar.L1.D.国内货币    -0.7151      0.054    -13.153      0.000      -0.822      -0.609
ar.L2.D.国内货币    -0.7725      0.036    -21.445      0.000      -0.843      -0.702
ar.L3.D.国内货币    -0.8323      0.038    -21.788      0.000      -0.907      -0.757
ar.L4.D.国内货币    -0.4458      0.030    -14.637      0.000      -0.505      -0.386
ma.L1.D.国内货币    -0.4655      0.060     -7.815      0.000      -0.582      -0.349
ma.L2.D.国内货币     0.0386      0.057      0.675      0.500      -0.073       0.151
ma.L3.D.国内货币     0.0556      0.064      0.863      0.388      -0.071       0.182
ma.L4.D.国内货币    -0.4890      0.049     -9.916      0.000      -0.586      -0.392
                                    Roots
=============================================================================
                  Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
AR.1            0.2846           -1.0215j            1.0604           -0.2068
AR.2            0.2846           +1.0215j            1.0604            0.2068
AR.3           -1.2181           -0.7150j            1.4124           -0.4155
AR.4           -1.2181           +0.7150j            1.4124            0.4155
MA.1            1.0595           -0.0000j            1.0595           -0.0000
MA.2            0.1951           -1.1860j            1.2019           -0.2240
MA.3            0.1951           +1.1860j            1.2019            0.2240
MA.4           -1.3361           -0.0000j            1.3361           -0.5000
-----------------------------------------------------------------------------
```

最后，我们尝试对国内股票收益率用ARIMA模型建模。

In [10]:

```python
best_aic = np.inf
best_order = None
best_mdl_gg = None
Y = indexs_logret['国内股票']
pq_rng = range(5) # [0,1,2,3,4]
d_rng = range(2) # [0,1]
for i in pq_rng:
    for d in d_rng:
        for j in pq_rng:
            try:
                tmp_mdl = smt.ARIMA(Y, order=(i,d,j)).fit(method='mle', trend='nc')
                tmp_aic = tmp_mdl.aic
                if tmp_aic < best_aic:
                    best_aic = tmp_aic
                    best_order = (i, d, j)
                    best_mdl_gg = tmp_mdl
            except: continue

print('aic: {:6.5f} | order: {}'.format(best_aic, best_order))
```

```
d:\Anaconda3\lib\site-packages\statsmodels\base\model.py:496: ConvergenceWarning: Ma
ximum Likelihood optimization failed to converge. Check mle_retvals
  "Check mle_retvals", ConvergenceWarning)
d:\Anaconda3\lib\site-packages\statsmodels\base\model.py:496: ConvergenceWarning: Ma
ximum Likelihood optimization failed to converge. Check mle_retvals
  "Check mle_retvals", ConvergenceWarning)
d:\Anaconda3\lib\site-packages\statsmodels\base\model.py:496: ConvergenceWarning: Ma
ximum Likelihood optimization failed to converge. Check mle_retvals
  "Check mle_retvals", ConvergenceWarning)
d:\Anaconda3\lib\site-packages\statsmodels\base\model.py:496: ConvergenceWarning: Ma
ximum Likelihood optimization failed to converge. Check mle_retvals
  "Check mle_retvals", ConvergenceWarning)
d:\Anaconda3\lib\site-packages\statsmodels\base\model.py:496: ConvergenceWarning: Ma
ximum Likelihood optimization failed to converge. Check mle_retvals
  "Check mle_retvals", ConvergenceWarning)

aic: -6601.86081 | order: (3, 0, 2)
```

如我们所料，得到了参数(3,0,2)

其实，我们采用了如此多的模型来为时序数据进行建模，最主要的目的是为了能够预测。下面，我们需要评估一下采用ARIMA模型预测数据的效果。

预测国内股票收益率21天数据，并分别估算其95%和99%置信水平。

In [12]:

```
n_steps = 21

f, err95, ci95 = best_mdl_gg.forecast(steps=n_steps) # 95% CI
_, err99, ci99 = best_mdl_gg.forecast(steps=n_steps, alpha=0.01) # 99% CI

idx = pd.date_range(Y.index[-1], periods=n_steps, freq='D')
fc_95 = pd.DataFrame(np.column_stack([f, ci95]),
                     index=idx, columns=['forecast', 'lower_ci_95', 'upper_ci_95'])
fc_99 = pd.DataFrame(np.column_stack([ci99]),
                     index=idx, columns=['lower_ci_99', 'upper_ci_99'])
fc_all = fc_95.combine_first(fc_99)
fc_all.tail()
```

Out[12]:

	forecast	lower_ci_95	lower_ci_99	upper_ci_95	upper_ci_99
2017-02-19	-0.000601	-0.033384	-0.043686	0.032182	0.042483
2017-02-20	-0.001168	-0.033953	-0.044255	0.031617	0.041919
2017-02-21	0.000365	-0.032434	-0.042741	0.033164	0.043470
2017-02-22	0.001195	-0.031608	-0.041916	0.033999	0.044306
2017-02-23	-0.000131	-0.032945	-0.043256	0.032682	0.042993

In [13]:

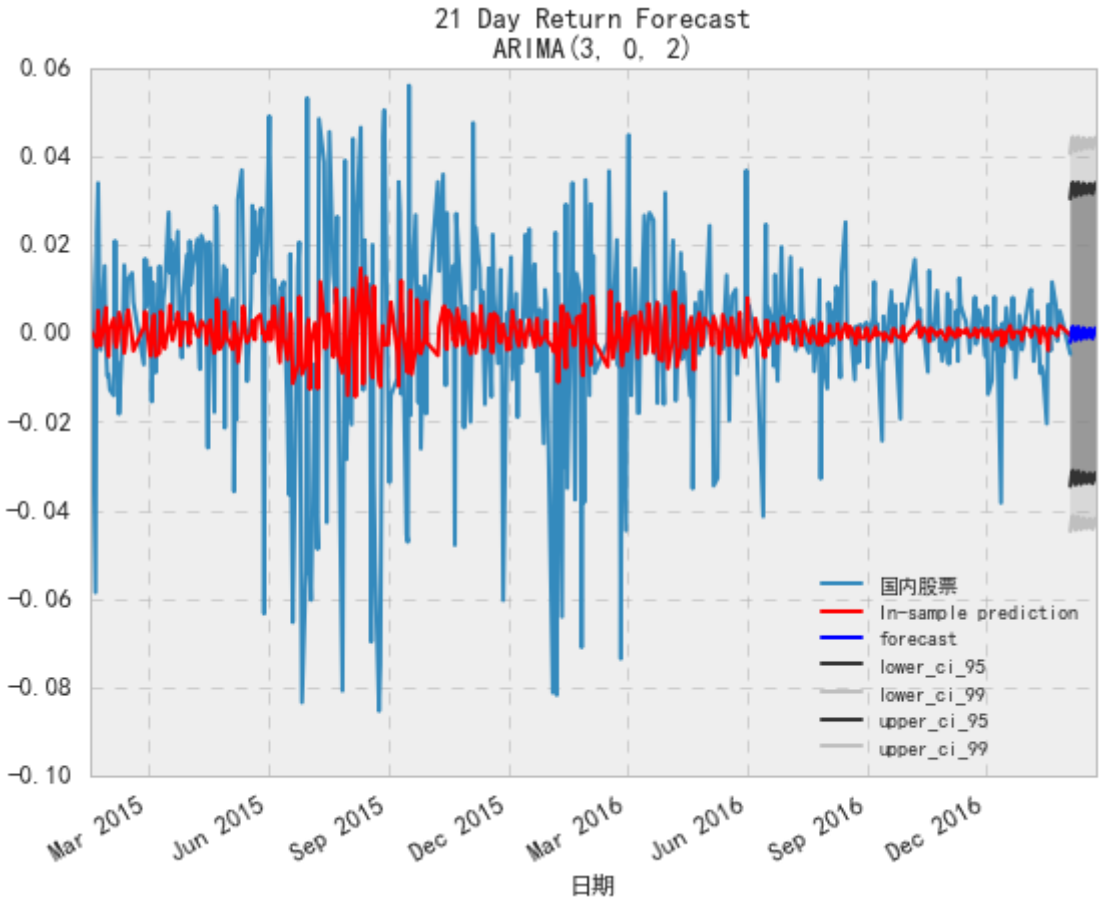
```
plt.style.use('bmh')
fig = plt.figure(figsize=(9,7))
ax = plt.gca()

ts = indexs_logret[['国内股票']].iloc[-500:].copy()
ts.plot(ax=ax, label='GG Returns')
# in sample prediction
pred = best_mdl_gg.predict(ts.index[0], ts.index[-1])
pred.plot(ax=ax, style='r-', label='In-sample prediction')

styles = ['b-', '0.2', '0.75', '0.2', '0.75']
fc_all.plot(ax=ax, style=styles)
plt.fill_between(fc_all.index, fc_all.lower_ci_95, fc_all.upper_ci_95, color='gray', alpha=0.7)
plt.fill_between(fc_all.index, fc_all.lower_ci_99, fc_all.upper_ci_99, color='gray', alpha=0.2)
plt.title('{} Day Return Forecast\nARIMA{}'.format(n_steps, best_order))
plt.legend(loc='best', fontsize=10)
```

Out[13]:

<matplotlib.legend.Legend at 0x22a738e1780>



总结

本文展示了采用Python语言为四个指数时序数据进行ARIMA建模，介绍了ARIMA模型的基本概念，并使用ARIMA模型对四指数数据进行建模其对国内股票收益率的预测进行了评估。