[python] 时间序列分析之ARIMA - 走那条小路 - 博客频道 - CSDN

[python] 时间序列分析之ARIMA

1 时间序列与时间序列分析

在生产和科学研究中，对某一个或者一组变量 $x(t)$ 进行观察测量，将在一系列时刻 $t_1,t_2,\cdots,t_n$ 所得到的离散数字组成的序列集合，称之为时间序列。
时间序列分析是根据系统观察得到的时间序列数据，通过曲线拟合和参数估计来建立数学模型的理论和方法。时间序列分析常用于国民宏观经济控制、市场潜力预测、气象预测、农作物害虫灾害预报等各个方面。

2 时间序列建模基本步骤

1. 获取被观测系统时间序列数据；
2. 对数据绘图，观测是否为平稳时间序列；对于非平稳时间序列要先进行**d阶差分运算**，化为平稳时间序列；
3. 经过第二步处理，已经得到平稳时间序列。要对平稳时间序列分别求得其**自相关系数ACF** 和**偏自相关系数PACF** ，通过对自相关图和偏自相关图的分析，得到最佳的**阶层 p** 和**阶数 q**
4. 由以上得到的 d、q、p ，得到ARIMA模型。然后开始对得到的模型进行模型检验。

3 ARIMA实战解剖

原理大概清楚，实践却还是会有诸多问题。相比较R语言，Python在做时间序列分析的资料相对少很多。下面就通过Python语言详细解析后三个步骤的实现过程。
文中使用到这些基础库： $pandas,numpy,scipy,matplotlib,statsmodels$。 对其调用如下

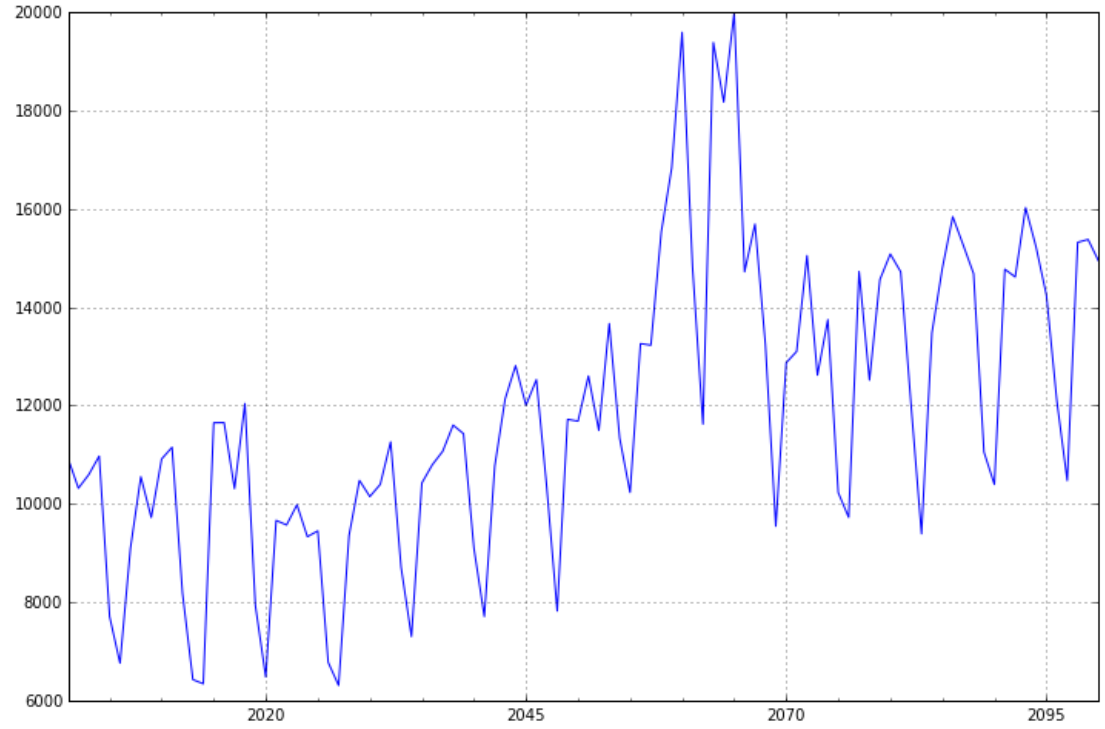
```
from __future__ import print_function

import pandas as pd

import numpy as np

from scipy import  stats

import matplotlib.pyplot as plt

import statsmodels.api as sm

from statsmodels.graphics.api import qqplot
```

3.1 获取数据

这里我们使用一个具有周期性的测试数据，进行分析。
数据如下：

```
dta=[10930,10318,10595,10972,7706,6756,9092,10551,9722,10913,11151,8186,6422,
6337,11649,11652,10310,12043,7937,6476,9662,9570,9981,9331,9449,6773,6304,9355,
10477,10148,10395,11261,8713,7299,10424,10795,11069,11602,11427,9095,7707,10767,
12136,12812,12006,12528,10329,7818,11719,11683,12603,11495,13670,11337,10232,
13261,13230,15535,16837,19598,14823,11622,19391,18177,19994,14723,15694,13248,
9543,12872,13101,15053,12619,13749,10228,9725,14729,12518,14564,15085,14722,
11999,9390,13481,14795,15845,15271,14686,11054,10395]
```

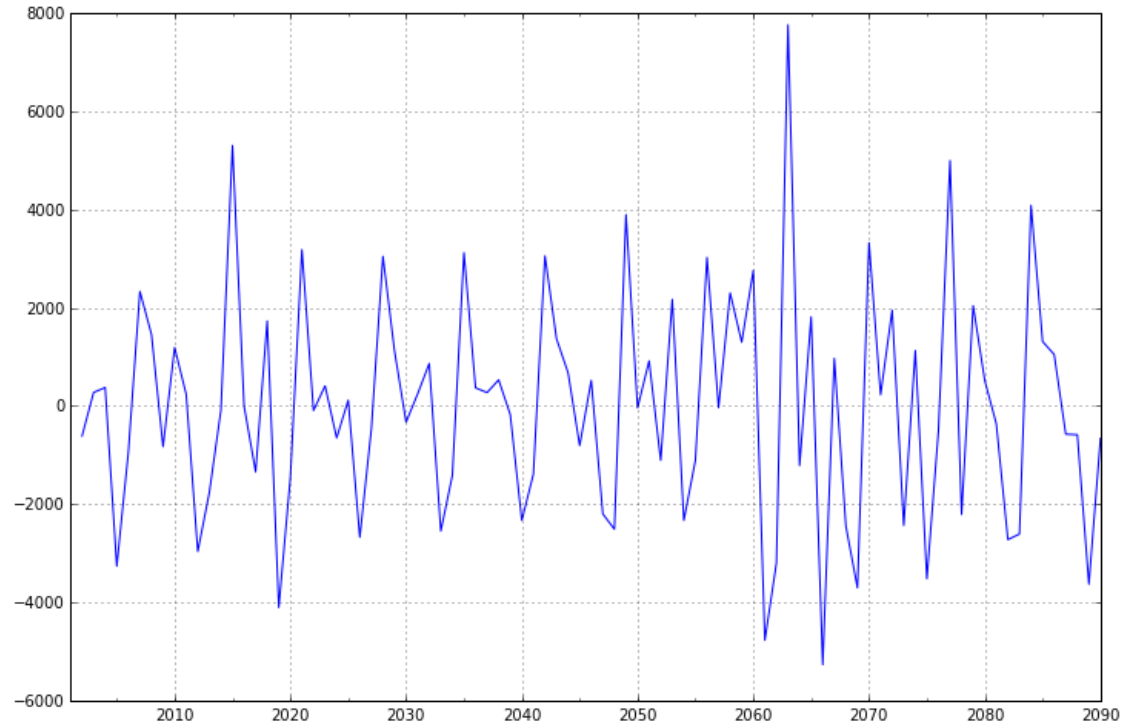
```
dta=pd.Series(dta)

dta.index = pd.Index(sm.tsa.datetools.dates_from_range('2001','2100'))

dta.plot(figsize=(12,8))
```



3.2 时间序列的差分d

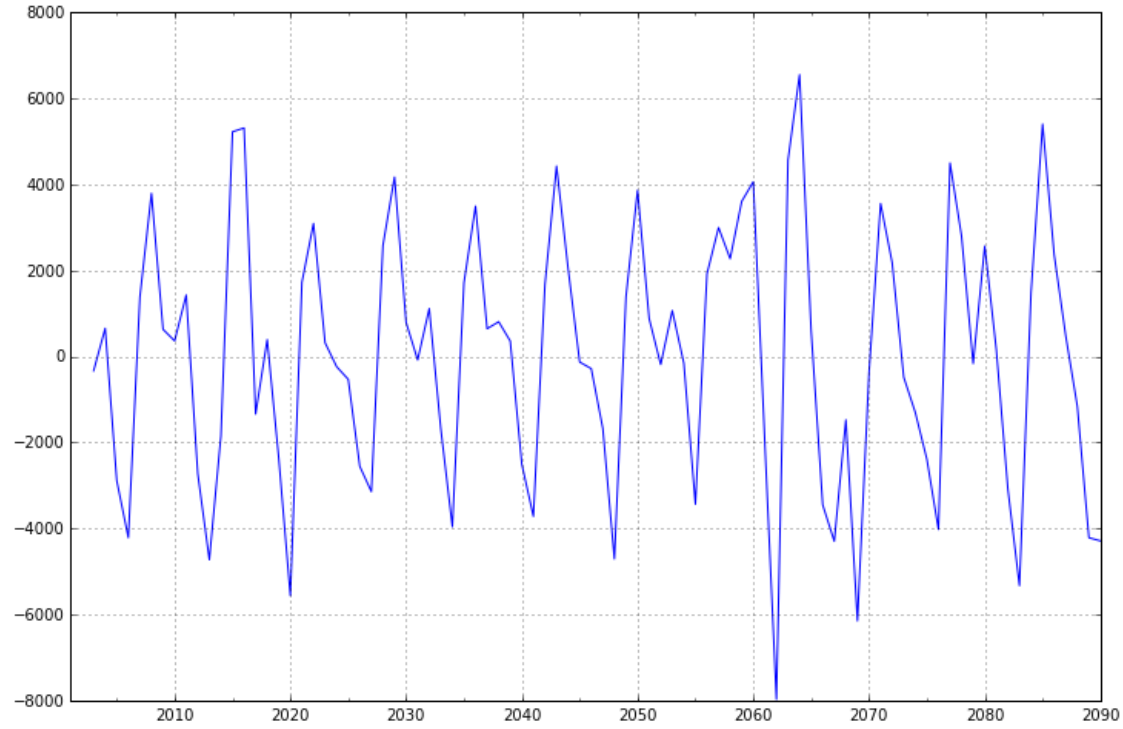
ARIMA 模型对时间序列的要求是平稳型。因此，当你得到一个非平稳的时间序列时，首先要做的即是做时间序列的差分，直到得到一个平稳时间序列。如果你对时间序列做d次差分才能得到一个平稳序列，那么可以使用ARIMA(p,d,q)模型，其中d是差分次数。

```
fig = plt.figure(figsize=(12,8))

ax1= fig.add_subplot(111)

diff1 = dta.diff(1)

diff1.plot(ax=ax1)
```



一阶差分的时间序列的均值和方差已经基本平稳，不过我们还是可以比较一下二阶差分的效果

```
fig = plt.figure(figsize=(12,8))

ax2= fig.add_subplot(111)

diff2 = dta.diff(2)

diff2.plot(ax=ax2)
```



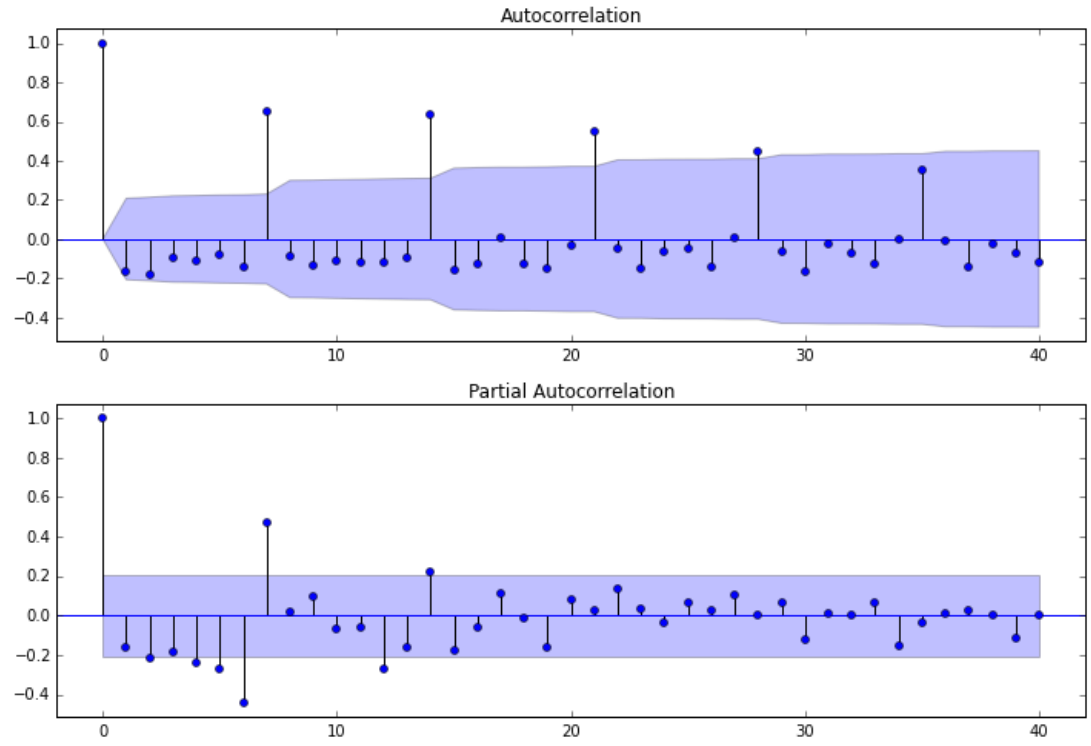
可以看出二阶差分后的时间序列与一阶差分相差不大，并且二者随着时间推移，时间序列的均值和方差保持不变。因此可以将差分次数d设置为1。
其实还有针对平稳的检验，叫“ADF单位根平稳型检验”，以后再更。

3.3 合适的p,q

现在我们已经得到一个平稳的时间序列，接来下就是选择合适的ARIMA模型，即ARIMA模型中合适的p,q。
第一步我们要先检查平稳时间序列的自相关图和偏自相关图。

```
dta= dta.diff(1)#我们已经知道要使用一阶差分的时间序列，之前判断差分的程序可以注释掉

fig = plt.figure(figsize=(12,8))

ax1=fig.add_subplot(211)

fig = sm.graphics.tsa.plot_acf(dta,lags=40,ax=ax1)

ax2 = fig.add_subplot(212)

fig = sm.graphics.tsa.plot_pacf(dta,lags=40,ax=ax2)
```

其中lags 表示滞后的阶数，以上分别得到acf 图和pacf 图



通过两图观察得到：
* 自相关图显示滞后有三个阶超出了置信边界；
* 偏相关图显示在滞后1至7阶（lags 1,2,...，7）时的偏自相关系数超出了置信边界，从lag 7之后偏自相关系数值缩小至0
则有以下模型可以供选择：
1. ARMA(0,1)模型：即自相关图在滞后1阶之后缩小为0，且偏自相关缩小至0，则是一个阶数q=1的移动平均模型；
2. ARMA(7,0)模型：即偏自相关图在滞后7阶之后缩小为0，且自相关缩小至0，则是一个阶层p=3的自回归模型；
3. ARMA(7,1)模型：即使得自相关和偏自相关都缩小至零。则是一个混合模型。
4. ...还可以有其他供选择的模型
现在有以上这么多可供选择的模型，我们通常采用ARMA模型的AIC法则。我们知道：增加自由参数的数目提高了拟合的优良性，AIC鼓励数据拟合的优良性但是尽量避免出现过度拟合(Overfitting)的情况。所以优先考虑的模型应是AIC值最小的那一个。赤池信息准则的方法是寻找可以最好地解释数据但包含最少自由参数的模型。不仅仅包括AIC准则，目前选择模型常用如下准则：
* AIC=-2 ln(L) + 2 k 中文名字：赤池信息量 akaike information criterion
* BIC=-2 ln(L) + ln(n)*k 中文名字：贝叶斯信息量 bayesian information criterion
* HQ=-2 ln(L) + ln(ln(n))*k hannan-quinn criterion
构造这些统计量所遵循的统计思想是一致的，就是在考虑拟合残差的同时，依自变量个数施加“惩罚”。但要注意的是，这些准则不能说明某一个模型的精确度，也即是说，对于三个模型A，B，C，我们能够判断出C模型是最好的，但不能保证C模型能够很好地刻画数据，因为有可能三个模型都是糟糕的。

```
arma_mod20 = sm.tsa.ARMA(dta,(7,0)).fit()

print(arma_mod20.aic,arma_mod20.bic,arma_mod20.hqic)

arma_mod30 = sm.tsa.ARMA(dta,(0,1)).fit()

print(arma_mod30.aic,arma_mod30.bic,arma_mod30.hqic)

arma_mod40 = sm.tsa.ARMA(dta,(7,1)).fit()

print(arma_mod40.aic,arma_mod40.bic,arma_mod40.hqic)

arma_mod50 = sm.tsa.ARMA(dta,(8,0)).fit()

print(arma_mod50.aic,arma_mod50.bic,arma_mod50.hqic)
```

```
1579.70255481 1602.10028214 1588.73043594
1632.32037328 1639.78628239 1635.32966699
1581.09160559 1605.97796929 1591.12258462
1581.39578369 1606.28214739 1591.42676273
```

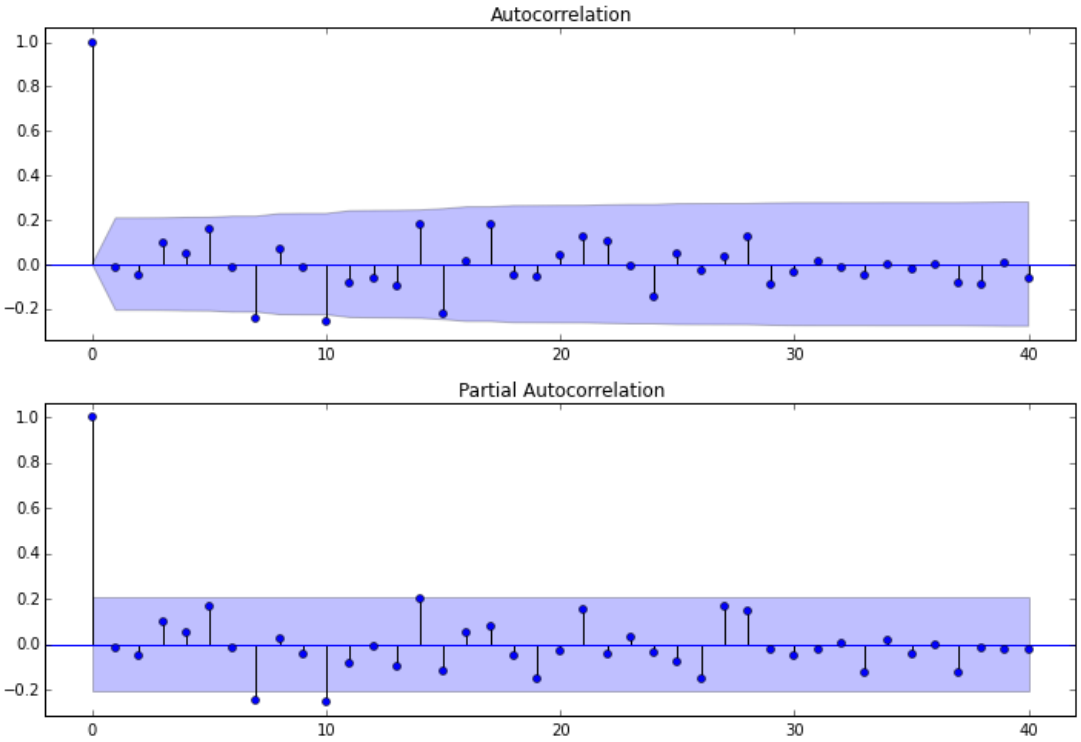
可以看到ARMA(7,0)的aic，bic，hqic均最小，因此是最佳模型。

3.4 模型检验

在指数平滑模型下，观察ARIMA模型的残差是否是平均值为0且方差为常数的正态分布（服从零均值、方差不变的正态分布），同时也要观察连续残差是否（自）相关。

3.4.1 我们对ARMA(7,0)模型所产生的残差做自相关图

```
fig = plt.figure(figsize=(12,8))

ax1 = fig.add_subplot(211)

fig = sm.graphics.tsa.plot_acf(resid.values.squeeze(), lags=40, ax=ax1)

ax2 = fig.add_subplot(212)

fig = sm.graphics.tsa.plot_pacf(resid, lags=40, ax=ax2)
```



3.4.2 做D-W检验

德宾-沃森（Durbin-Watson）检验。德宾-沃森检验,简称D-W检验，是目前检验自相关性最常用的方法，但它只使用于检验一阶自相关性。因为自相关系数ρ的值介于-1和1之间，所以 0≤DW≤4。并且DW=O= >ρ=1　　即存在正自相关性
DW=4 < = >ρ=-1　即存在负自相关性
DW=2 < = >ρ=0　　即不存在（一阶）自相关性
因此，当DW值显著的接近于O或4时，则存在自相关性，而接近于2时，则不存在（一阶）自相关性。这样只要知道DW统计量的概率分布，在给定的显著水平下，根据临界值的位置就可以对原假设H 0进行检验。

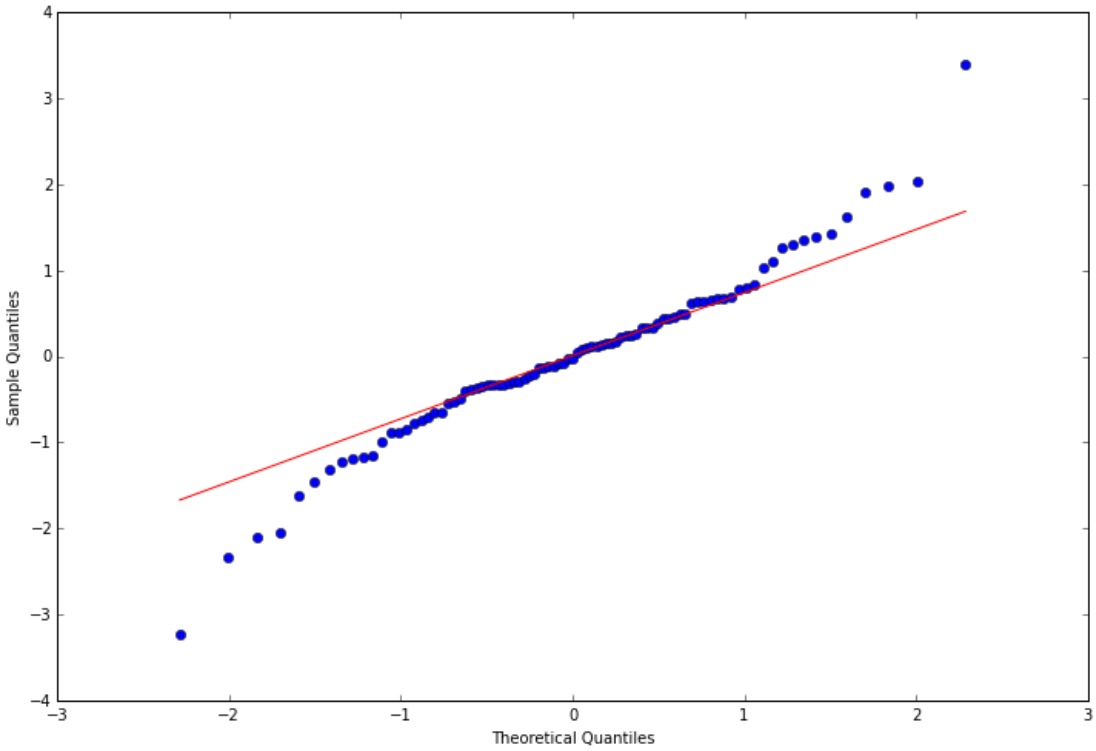
```
print(sm.stats.durbin_watson(arma_mod20.resid.values))
```

检验结果是2.02424743723，说明不存在自相关性。

3.4.3 观察是否符合正态分布

这里使用QQ图，它用于直观验证一组数据是否来自某个分布，或者验证某两组数据是否来自同一（族）分布。在教学和软件中常用的是检验数据是否来自于正态分布。QQ图细节，下次再更。

```
resid = arma_mod20.resid#残差

fig = plt.figure(figsize=(12,8))

ax = fig.add_subplot(111)

fig = qqplot(resid, line='q', ax=ax, fit=True)
```



3.4.4 Ljung-Box检验

Ljung-Box test是对randomness的检验,或者说是对时间序列是否存在滞后相关的一种统计检验。对于滞后相关的检验，我们常常采用的方法还包括计算ACF和PCAF并观察其图像，但是无论是ACF还是PACF都仅仅考虑是否存在某一特定滞后阶数的相关。LB检验则是基于一系列滞后阶数，判断序列总体的相关性或者说随机性是否存在。
时间序列中一个最基本的模型就是高斯白噪声序列。而对于ARIMA模型，其残差被假定为高斯白噪声序列，所以当我们用ARIMA模型去拟合数据时，拟合后我们要对残差的估计序列进行LB检验，判断其是否是高斯白噪声，如果不是，那么就说明ARIMA模型也许并不是一个适合样本的模型。

```
r,q,p = sm.tsa.acf(resid.values.squeeze(), qstat=True)

data = np.c_[range(1,41), r[1:], q, p]

table = pd.DataFrame(data, columns=['lag', "AC", "Q", "Prob(>Q)"])

print(table.set_index('lag'))
```

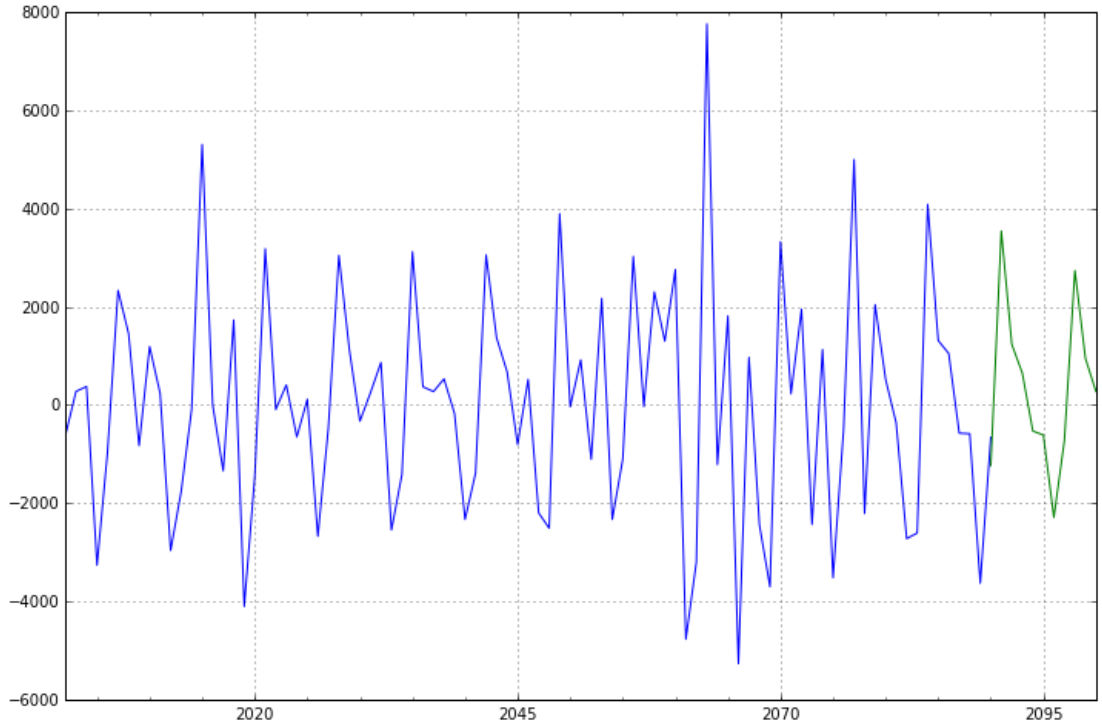
lag	AC	Q	Prob(>Q)
1	-0.014445	0.019204	0.889782
2	-0.047441	0.228723	0.891935
3	0.097777	1.129061	0.770064
4	0.047513	1.344160	0.853840
5	0.156219	3.697134	0.593790
6	-0.017856	3.728245	0.713396
7	-0.241228	9.475682	0.220283
8	0.068078	9.939090	0.269327
9	-0.012041	9.953768	0.354241
10	-0.256684	16.708387	0.081071
11	-0.085178	17.461731	0.094940
12	-0.063576	17.886873	0.119169
13	-0.096511	18.879475	0.126888
14	0.181120	22.421903	0.070351
15	-0.223097	27.869257	0.022402
16	0.012916	27.887766	0.032609
17	0.176768	31.402615	0.017834
18	-0.053140	31.724738	0.023695
19	-0.057704	32.109990	0.030375
20	0.037426	32.274398	0.040461
21	0.120520	34.004381	0.036200
22	0.102662	35.278404	0.036226
23	-0.007829	35.285926	0.048712
24	-0.148547	38.035376	0.034384
25	0.046254	38.306116	0.043174

检验的结果就是看最后一列前十二行的检验概率（一般观察滞后1~12阶），如果检验概率小于给定的显著性水平，比如0.05、0.10等就拒绝原假设，其原假设是相关系数为零。就结果来看，如果取显著性水平为0.05，那么相关系数与零没有显著差异，即为白噪声序列。

3.5 模型预测

模型确定之后，就可以开始进行预测了，我们对未来十年的数据进行预测。

```
predict_sunspots = arma_mod20.predict('2090', '2100', dynamic=True)print(predict_sunspots)

fig, ax = plt.subplots(figsize=(12, 8))

ax = dta.ix['2001':].plot(ax=ax)

predict_sunspots.plot(ax=ax)
```



前面90个数据为测试数据，最后10个为预测数据；从图形来，预测结果较为合理。至此，本案例的时间序列分析也就结束了。

参考文献与推荐阅读

1. statsmodels--statistics in python
2. 时间序列分析—(ARIMA模型)
3. Arima预测模型（R语言）
4. 介绍QQplot
5. LBQ检验
6. 经管之家