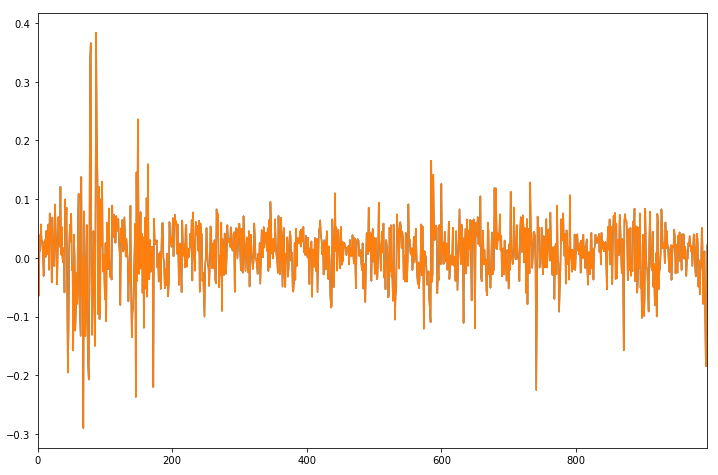
## 2．

基本步骤：

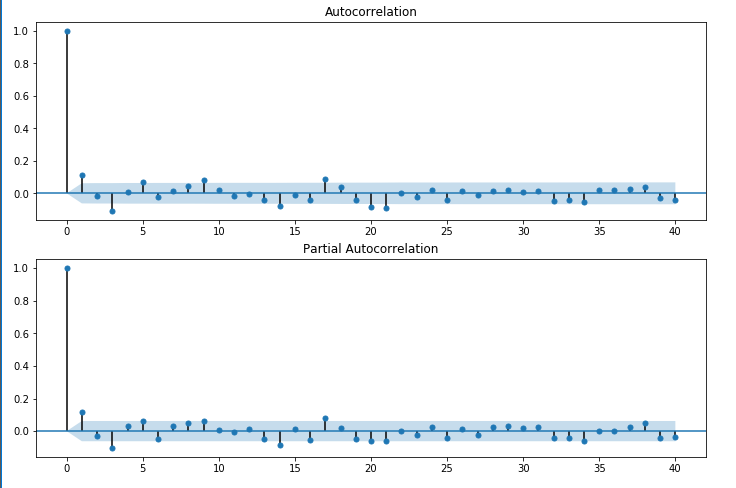
1. 初步对数据绘图，观测是否为平稳时间序列；如果为非平稳时间序列，先进行**d阶差分运算**，化为平稳时间序列；
2. 对已得到的平稳时间序列，分别求得其**自相关系数ACF** 和**偏自相关系数PACF** ，通过对自相关图和偏自相关图的分析，得到最佳的**阶层 p**和**阶数 q**
3. 由以上得到的d、q、p ，得到ARIMA模型。对得到的模型进行模型检验。

首先进行初步绘图分析：（我们使用python完成整体的分析工作，关键代码过程中附或者后附）



为便于数据处理和绘图，其中横轴代表月份，从1926-01 至 2008 – 12，共996单位数据，我们可以初步得到以下结论：该时间序列基本上是平稳的，没有明显的趋势，同时也没有明显的周期现象。同时可以看到，除了少数几个极端值，多数值是稳定的。我们可以初步认为是不需要进行差分的，事实上，通过后续的分析可以进一步验证这一点。

### 接下来进行第二步，分析ACF和PACF，从而确定p和q：

其中横轴为Lags。从样本的ACF and PACF之中，我们可以进一步验证事实上并不需要进行进一步的差分，ACF宏观上存在着随着Lag增加逐渐减小的趋势，符合平稳过程的特征。同时ACF的多数值均落在95%置信区间之内（即图中的蓝色区域）

为了确定ARMA中的p，q，进一步观察上图，我们发现在滞后1阶和3阶时，ACF和PACF均超出了置信区间，但滞后二阶时ACF与PACF均落回置信区间，且缩小至接近零。

因此我们最可能的选择为ARMA（2,2）模型，及自相关图在滞后2阶之后缩小为0，且偏自相关图在滞后2阶后同样缩小为零。

我们不妨根据AIC，BIC以及HQ信息准则进行检验，和备选的(4,4)模型，以及常用的低阶ARMA模型进行比较：

此处代码为：

dta = Vwrtn

arma\_mod10 = sm.tsa.ARMA(dta,(1,0)).fit()

print(arma\_mod10.aic,arma\_mod10.bic,arma\_mod10.hqic)

arma\_mod01 = sm.tsa.ARMA(dta,(0,1)).fit()

print(arma\_mod01.aic,arma\_mod01.bic,arma\_mod01.hqic)

arma\_mod11 = sm.tsa.ARMA(dta,(1,1)).fit()

print(arma\_mod11.aic,arma\_mod11.bic,arma\_mod11.hqic)

arma\_mod22 = sm.tsa.ARMA(dta,(2,2)).fit()

print(arma\_mod22.aic,arma\_mod22.bic,arma\_mod22.hqic)

arma\_mod44 = sm.tsa.ARMA(dta,(4,4)).fit()

print(arma\_mod44.aic,arma\_mod44.bic,arma\_mod44.hqic)

得到的信息格式为（AIC，BIC，HQ），结果为：

(-2984.0869262386755, -2969.3756844659215, -2978.49454016869)

(-2984.3894095883356, -2969.6781678155817, -2978.7970235183502)

(-2982.413254182797, -2962.798265152459, -2974.9567394228166)

(-2997.0194094336034, -2967.5969258880959, -2985.8346372936326)

(-2991.8872468362783, -2942.8497742604322, -2973.2459599363269)

我们发现综合比较之后，显然是(2,2)模型的三个信息数值均较小，与（0,1）和（1,0）模型相比，其AIC和HQ也均较小，BIC相差不大，因此我们认为ARMA(2,2)模型是最佳模型。

ARMA拟合结果为：

当然，其中是测量值，是残差。

## 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。

首先拟合出残差的ACF和PACF图像：

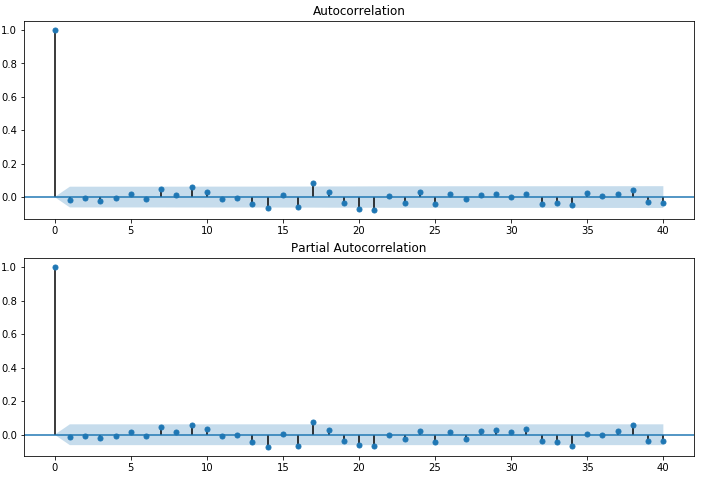
fig = plt.figure(figsize=(12,8))

ax1 = fig.add\_subplot(211)

fig = sm.graphics.tsa.plot\_acf(resid.squeeze(), lags=40, ax=ax1)

ax2 = fig.add\_subplot(212)

fig = sm.graphics.tsa.plot\_pacf(resid, lags=40, ax=ax2)



## 显然在几个滞后阶中，ACF和PACF均较为接近零，存在这很大的改善，说明残差相关性极低。几乎不存在自相关。对于一些ACF和PACF较大甚至超过置信区间的情况，我们可以认为是一些异常值的影响。

### DW检验

对于D-W检验，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＝0＝＞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0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Ｈ０进行检验。

代码为

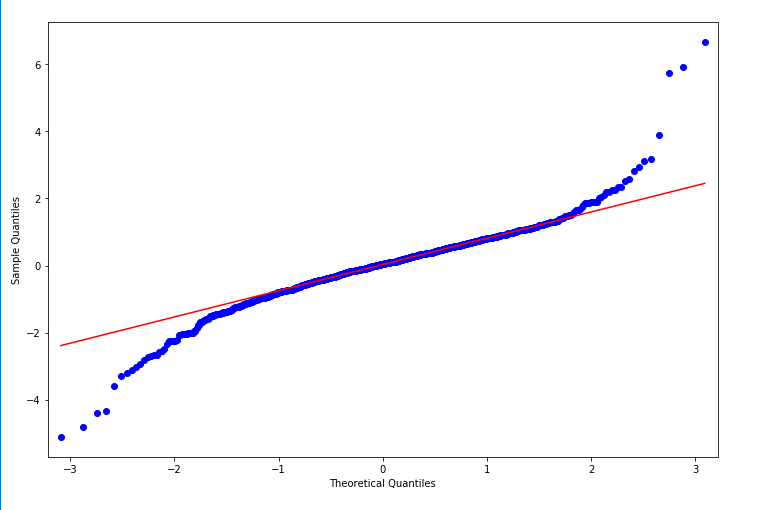
print(sm.stats.durbin\_watson(arma\_mod22.resid))

输出结果为：2.02901317618

这个值正在2 的附近，说明残差不存在自相关性。

### 观察是否符合正态分布

我们直接使用QQ图来验证是否为正态分布，图像为：



这里横坐标为正太分布的理论值，纵坐标为样本值，我们发现多数值均落在拟合直线附近

，相当程度上可以认为是正太分布的。

代码为：

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

ax = fig.add\_subplot(111)

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

### L-B检验

对于滞后相关的检验，我们之前已经计算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。 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，那么就说明该模型也许并不是一个适合样本的模型。

代码为：

r,q,p = sm.tsa.acf(resid.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'))

输出结果为：

AC Q Prob(>Q)

lag

1.0 -0.014539 0.211162 0.645858

2.0 -0.008193 0.278284 0.870104

3.0 -0.021105 0.724175 0.867503

4.0 -0.008301 0.793214 0.939356

5.0 0.017339 1.094773 0.954565

6.0 -0.008610 1.169204 0.978378

7.0 0.048300 3.513885 0.833753

8.0 0.013387 3.694196 0.883607

9.0 0.059470 7.256039 0.610481

10.0 0.028359 8.066795 0.622312

11.0 -0.010533 8.178758 0.697207

12.0 -0.005380 8.207996 0.768672

13.0 -0.043060 10.082914 0.687141

14.0 -0.065684 14.450093 0.416745

15.0 0.012209 14.601129 0.480511

16.0 -0.057965 18.009077 0.323366

17.0 0.083580 25.101703 0.092456

18.0 0.030726 26.061224 0.098366

19.0 -0.037746 27.510775 0.093307

20.0 -0.068431 32.280026 0.040404

21.0 -0.074961 38.008685 0.012858

22.0 0.004239 38.027027 0.018194

23.0 -0.034427 39.237863 0.018680

24.0 0.030425 40.184483 0.020431

25.0 -0.043417 42.114175 0.017472

26.0 0.019529 42.504980 0.021763

27.0 -0.014343 42.716001 0.027938

28.0 0.014952 42.945571 0.035229

29.0 0.017043 43.244159 0.043217

30.0 0.002064 43.248543 0.055702

31.0 0.015485 43.495545 0.067415

32.0 -0.039798 45.128742 0.061831

33.0 -0.034528 46.359337 0.061376

34.0 -0.048754 48.815402 0.047934

35.0 0.026053 49.517460 0.052844

36.0 0.007079 49.569351 0.065540

37.0 0.019618 49.968253 0.075498

38.0 0.042482 51.840826 0.066525

39.0 -0.030565 52.811188 0.068971

40.0 -0.036832 54.221715 0.066074

我们取显著性水平为0.05，观察最后一列的检验概率，发现多数均大于0.05，一些小于0.05，但这些已经是滞后阶较高的结果，对于前12行，我们发现最后一列值均大于0.0，因此可以认为自相关系数与零没有显著差异，即为高斯白噪声序列

经过以上检验，我们可以认为ARMA(2,2)模型是较为理想的模型，通过了以上所有检验，残差是不相关的Gauss白噪声序列。

# 附录

使用模块：

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

读入数据：（为方便读入，删去了数据集中第一行以及末尾的空行）

Path = 'C:/Users/DaiYi/Desktop/2\_data.txt'

Date = []

SplitedLine = []

for OriginalLine in open(Path):

Splited = OriginalLine.replace(' ', ',').split(',')

SplitedLine.append(Splited[0])

SplitedLine = SplitedLine[0:-1]

Date = [int(i) for i in SplitedLine]

Date.append(20081231)

SplitedLine = []

for OriginalLine in open(Path):

Splited = OriginalLine.replace(' ', ',').split(',')

SplitedLine.append(Splited[2])

Vwrtn = []

for i in SplitedLine:

Si = i.split()

Vwrtn = [float(i) for i in SplitedLine]

初步分析中的绘图：

dta = pd.Series(Vwrtn)

dta.index = pd.Index(Date)

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

plt.show(dta)

ACF和PACF的绘图：

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

ax1=fig.add\_subplot(211)

fig = sm.graphics.tsa.plot\_acf(Vwrtn,lags=40,ax=ax1)

ax2 = fig.add\_subplot(212)

fig = sm.graphics.tsa.plot\_pacf(Vwrtn,lags=40,ax=ax2)