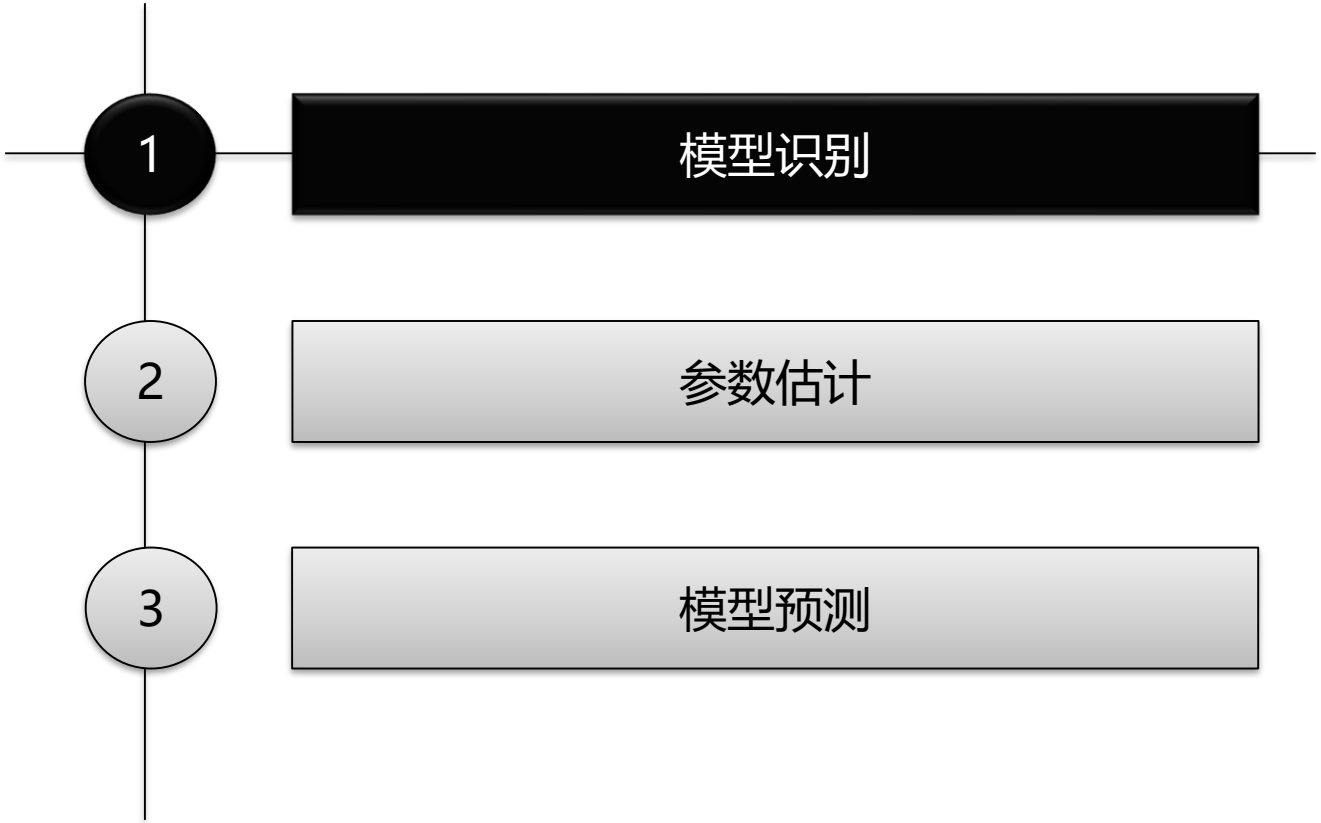
《Python与数据挖掘》

第11章 时间序列分析

讲师：武永亮

目录



时间序列

- 常用的时间序列模型见下表，本章以ARIMA模型为例介绍时间序列算法在python中是如何实现的。

模型名称	描述
ARIMA模型	可以实现AR模型、MA模型、ARMA模型及ARIMA模型
GARCH模型	也称为条件异方差模型，适用于金融时间序列。
时间序列分解	时间序列的变化主要受到长期趋势、季节变动、周期变动和不规则变动这四个因素的影响。根据序列的特点，可以构建加法模型和乘法模型。
指数平滑法	可以实现简单指数平滑法、Holt双参数线性指数平滑法和Winters线性和季节性指数平滑法

ARIMA模型

- 下面应用python语言建模步骤，对表中2013年1月到2016年1月某餐厅的营业数据进行建模，部分数据如下：

日期	销量	日期	销量
2013年1月	3023	2014年8月	3443
2013年2月	3039	2014年9月	3428
2013年3月	3056	2014年10月	3554
2013年4月	3138	2014年11月	3615
2013年5月	3188	2014年12月	3646
2013年6月	3224	2015年1月	3614
2013年7月	3226	2015年2月	3574
2013年8月	3029	2015年3月	3635
2013年9月	2859	2015年4月	3738
2013年10月	2870	2015年5月	3707
2013年11月	2910	2015年6月	3827
2013年12月	3012	2015年7月	4039
2014年1月	3142	2015年8月	4210
2014年2月	3252	2015年9月	4493
…	…	…	…

时间序列对象

- 加载基础库：pandas,numpy,scipy,matplotlib,statsmodels对其调用如下：

✓ import pandas as pd

- 从xls文件中读取数据

✓ # 参数初始化

✓ discfile = '../data/arima_data.xls' # 读取数据，指定日期列为指标，Pandas自动将“日期”列识别为Datetime格式

✓ data = pd.read_excel(discfile,index_col=0)

✓ print(data.head())

✓ print('\n Data Types:')

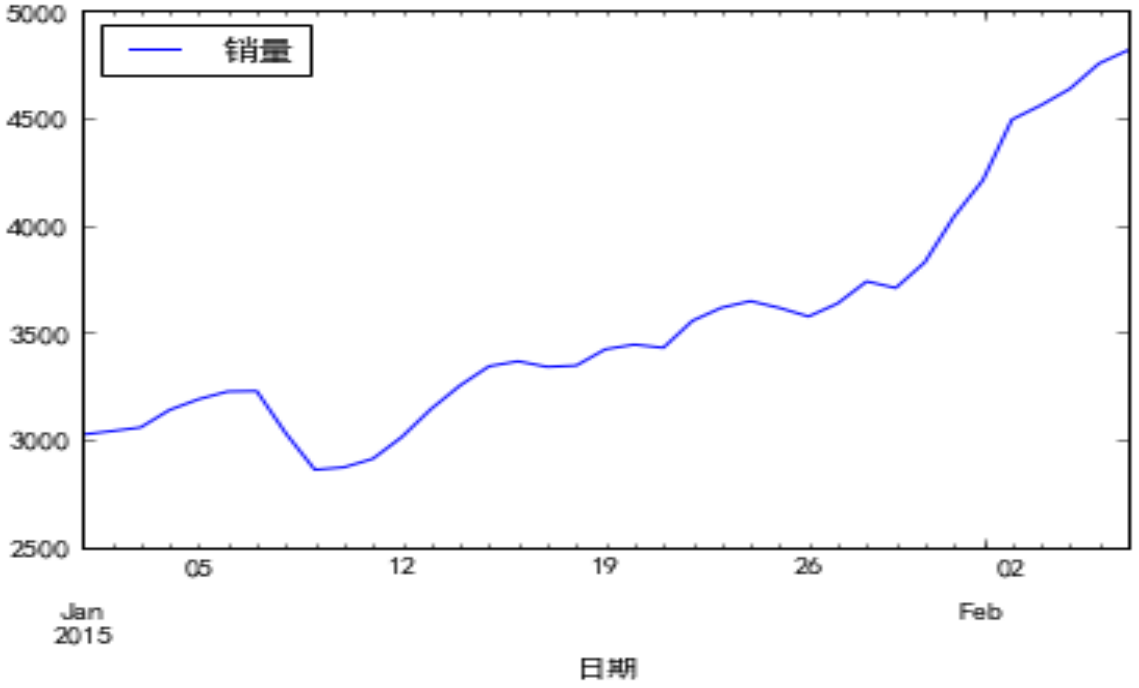
✓ print(data.dtypes)

绘制时间序列图

- 对读取的数据绘制时间序列图，观察图形的特征，所得图形见如下：

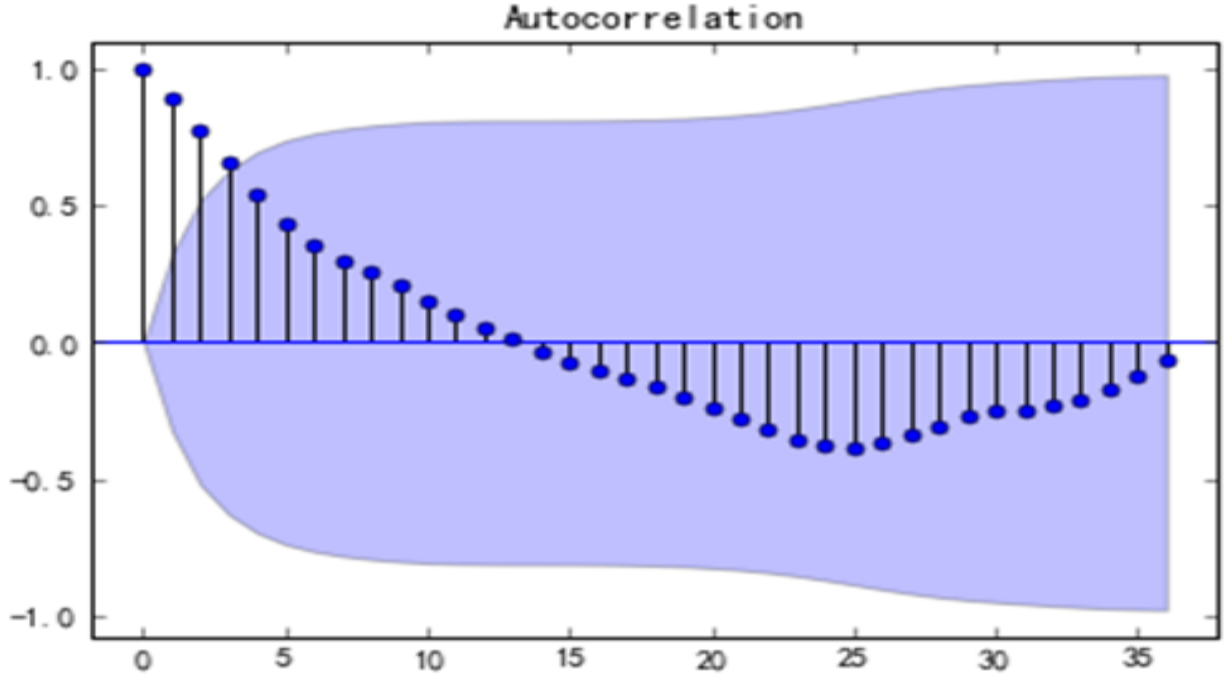
✓ # 时序图
✓ import matplotlib.pyplot as plt
✓ plt.rcParams['font.sans-serif'] = ['SimHei'] #用来正常显示中文标签
✓ plt.rcParams['axes.unicode_minus'] = False #用来正常显示负号
✓ data.plot()
✓ plt.show()

- 运行结果如下图：



自相关

- 对时间序列作自相关图，判断序列是否自相关
  - ✓ from statsmodels.graphics.tsaplots import #自相关图
  - ✓ plot_acfplot_acf(data).show()



- 由自相关图可以看出，在4阶后才落入区间内，并且自相关系数长期大于零，显示出很强的自相关性。

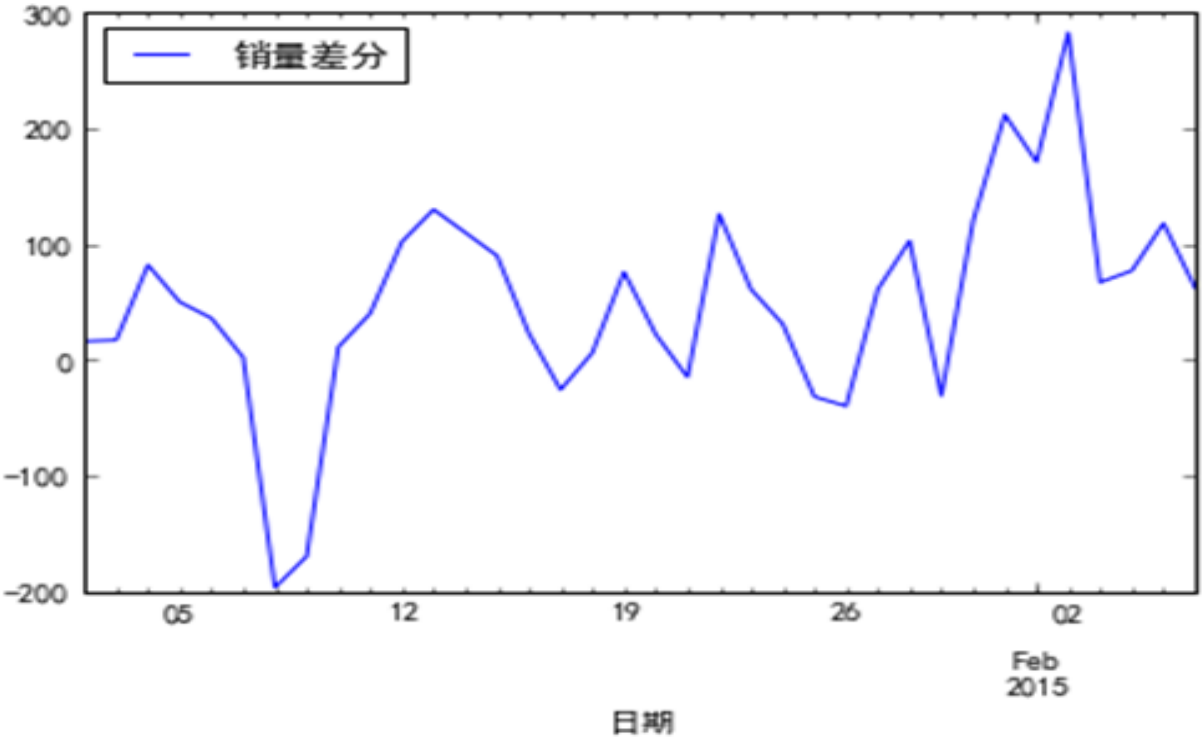
时间序列的差分

- 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。

时间序列的差分

- 首先对时间序列做差分，并观察差分后的时序图，见下图 ：

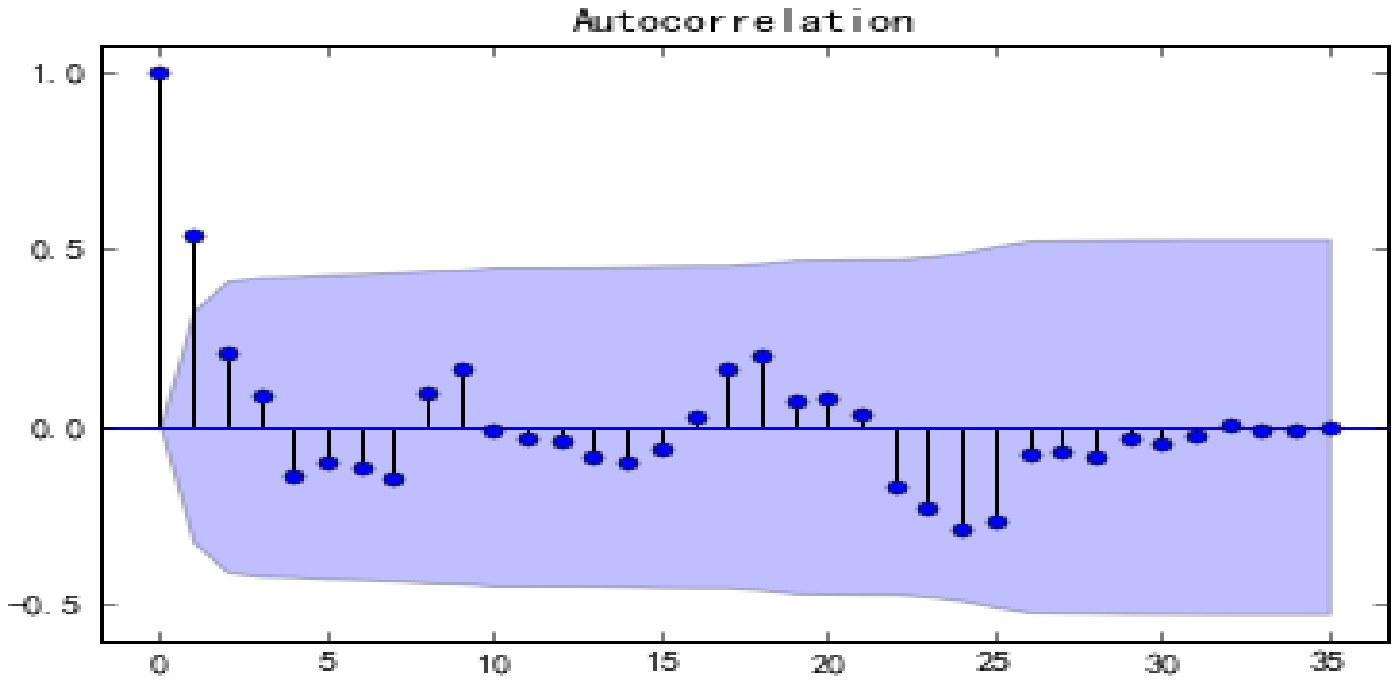
- ✓ D_data = data.diff().
- ✓ dropna()
- ✓ D_data.columns = [u'销量差分']D_data.plot() #时序图
- ✓ plt.show()



时间序列的差分

- 对差分后的序列做自相关检验，见图 12-4，观察是否自相关

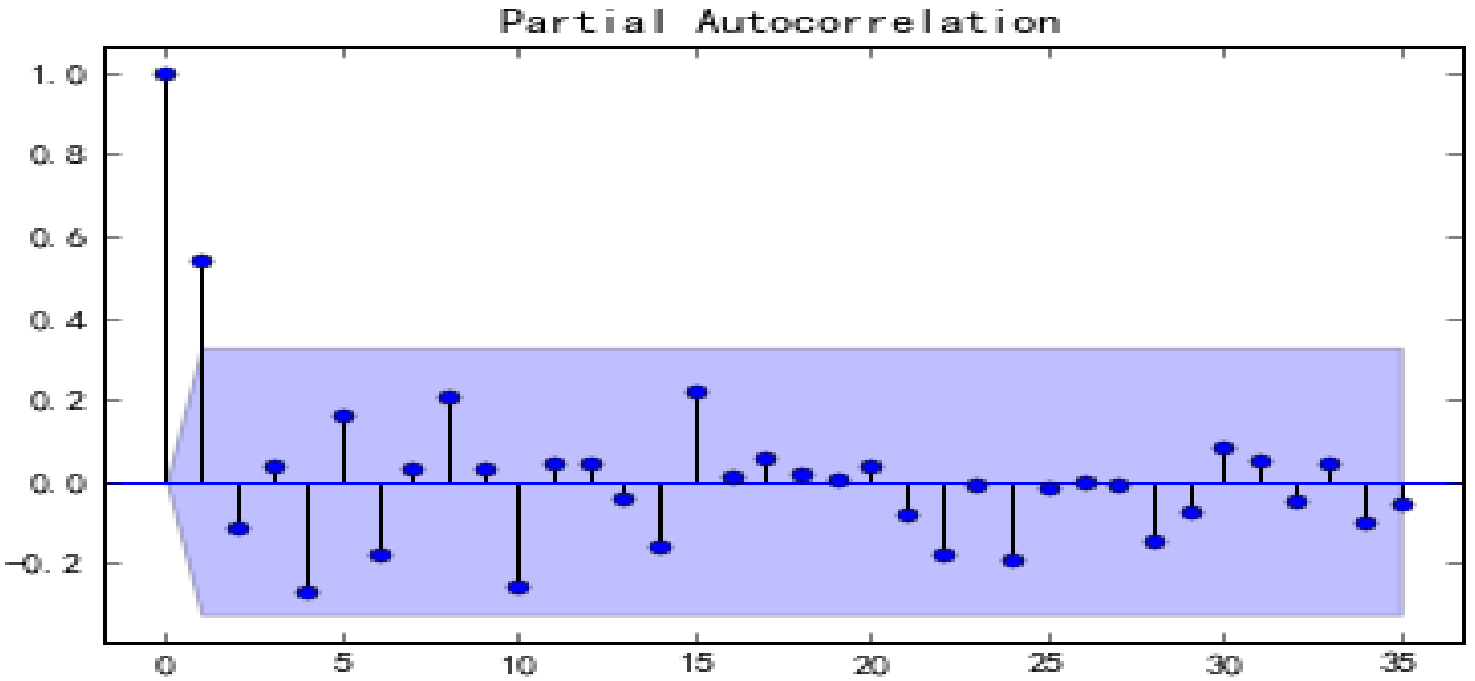
✓ plot_acf(D_data).show() #自相关图



- 由图可以看出，差分后的序列迅速落入区间内，并呈现出向0靠拢的趋势，序列没有自相关性。

相关性检验

- 对差分后的序列做偏自相关检验，见图 12-5，观察是否偏自相关
  - ✓ from statsmodels.graphics.tsaplots import
  - ✓ plot_pacfplot_pacf(D_data).show() #偏自相关图



- 由偏自相关图可以看出，差分后的序列也没有显示出偏自相关性

平稳性检验

- 再对差分后的序列做平稳性检测

✓ #平稳性检测

✓ print(u'差分序列的ADF检验结果为：', ADF(D_data[u'销量差分']))

✓ #result:差分序列的ADF检验结果为： (-3.156056236672353 7, 0.022673435440048798, 0L, 35L, {'5%': -2.948510204081632 7, '1%': -3.632742664723031 6, '10%': -2.613017346938775 6}, 287.59090907803341)

✓ 从返回的ADF检验结果得到，p值为0.022673435440048798，小于0.05

- 还需要对差分后的序列做白噪声检验

✓ #白噪声检验

✓ from statsmodels.stats.diagnostic import acorr_ljungbox

✓ print(u'差分序列的白噪声检验结果为：', acorr_ljungbox(D_data, lags=1))

✓ #返回统计量和p值

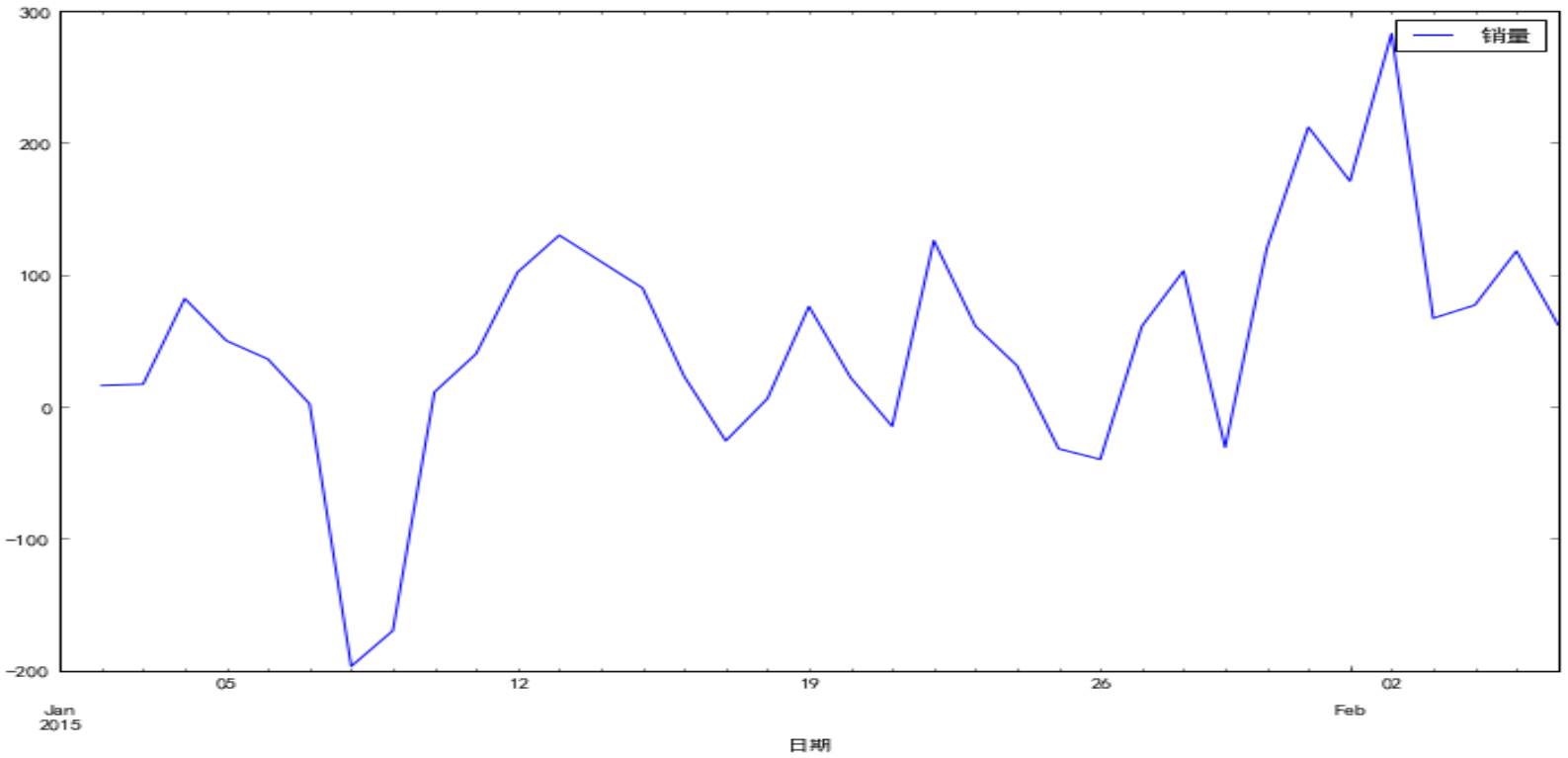
✓ #result:差分序列的白噪声检验结果为： (array([11.30402222]), array([0.00077339]))

- 从得到的白噪声检验结果可以看出，检验的p值为0.00077339，小于0.05，通过白噪声检验，序列为白噪声序列。

一阶差分

- 接下来我们比较下一阶差分后的序列和二阶差分后的序列

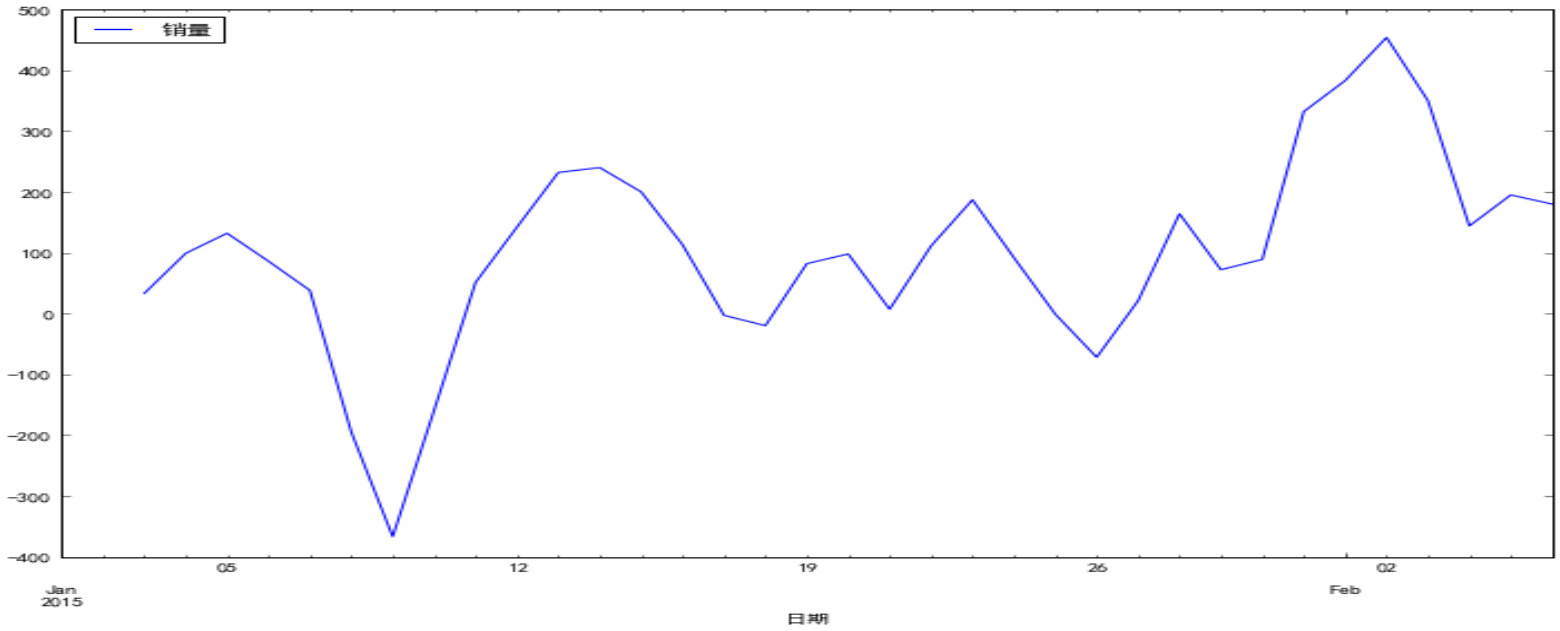
✓ fig = plt.figure(figsize=(12,8))ax1= fig.add_subplot(111)diff1 = data.diff(1)diff1.plot(ax=ax1) # 一阶差分



- 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

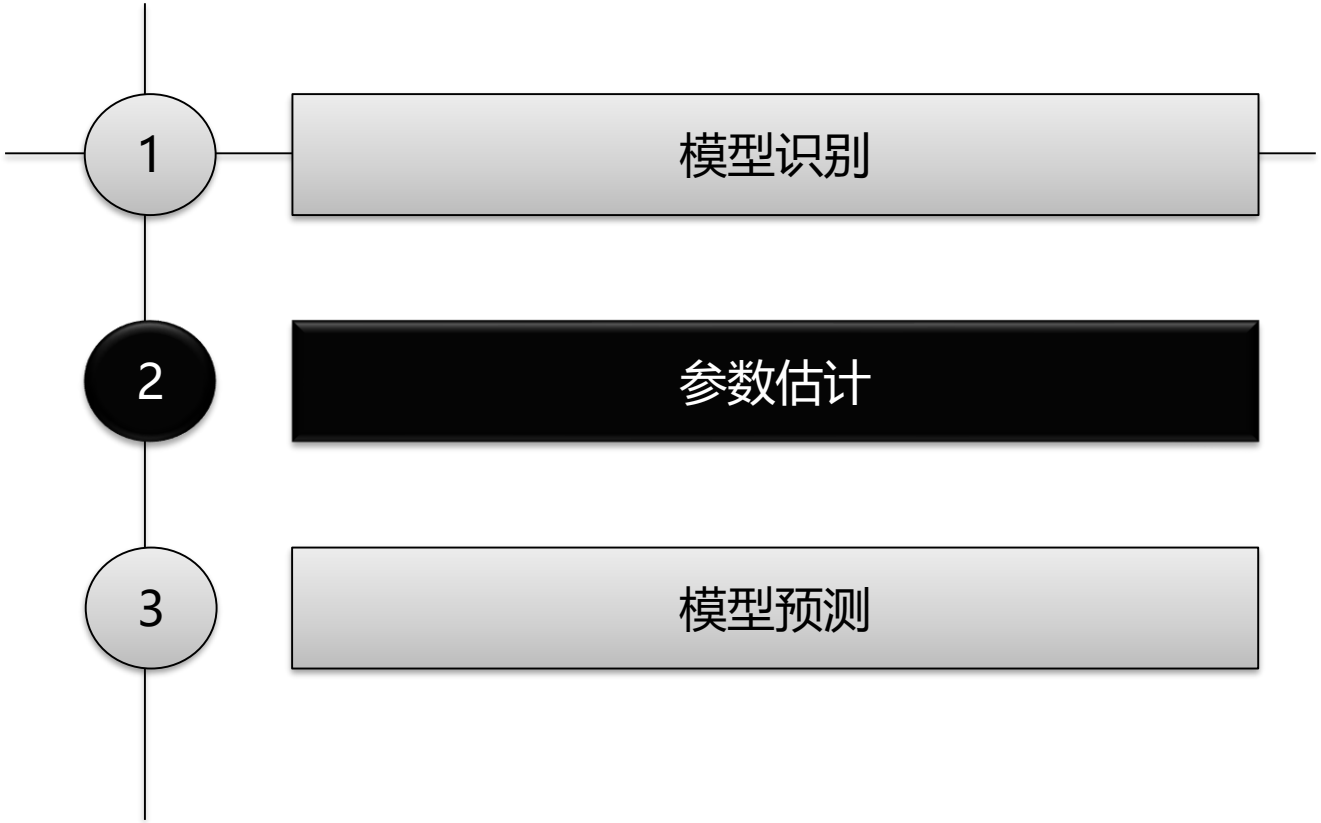
二阶差分

- ✓ # 二阶差分
- ✓ 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。

目录



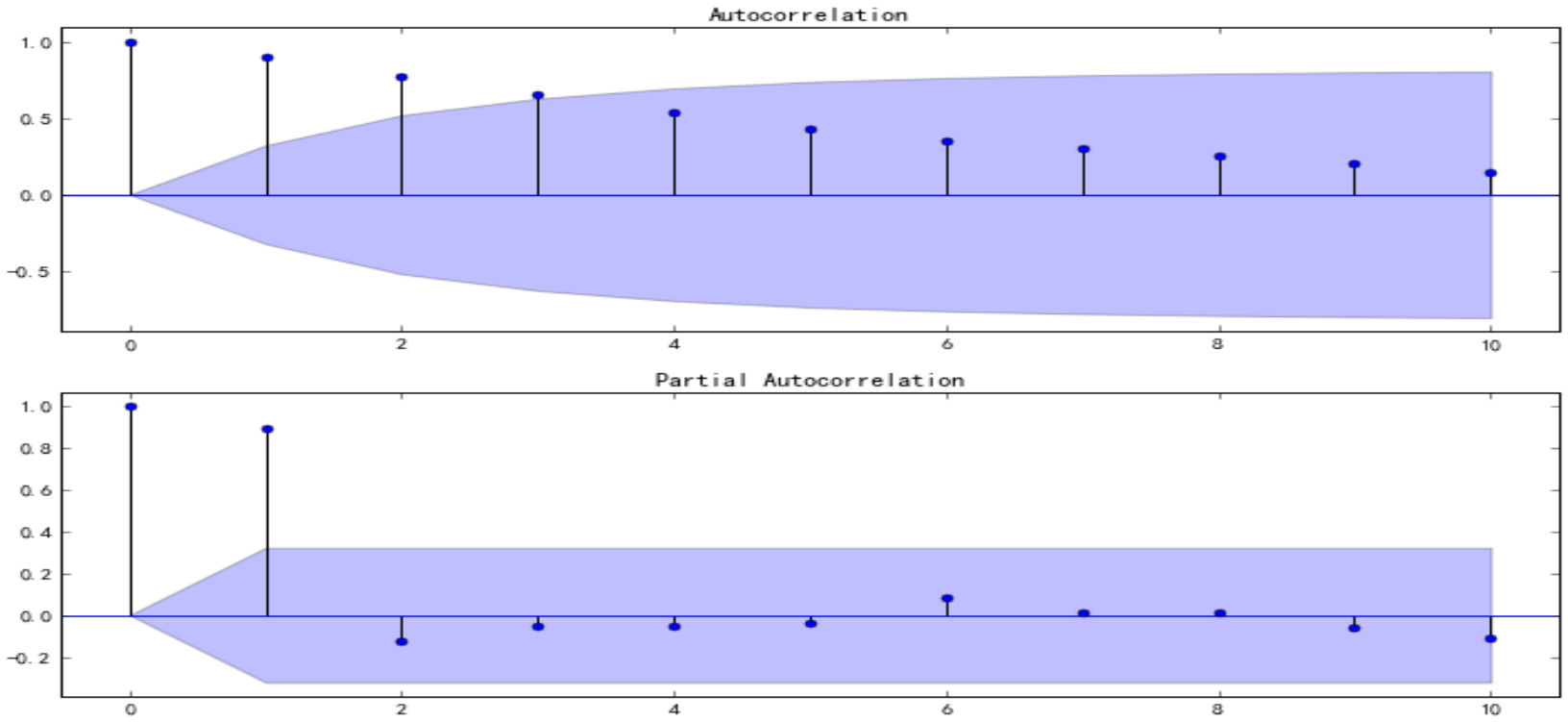
定阶

- 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：

✓ # 合适的p,q
✓ dta = data.diff(1)[1:]
✓ fig = plt.figure(figsize=(12,8))
✓ ax1=fig.add_subplot(211)
✓ fig1 = sm.graphics.tsa.plot_acf(dta[u'销量'],lags=10,ax=ax1)
✓ ax2 = fig.add_subplot(212)
✓ fig2 = sm.graphics.tsa.plot_pacf(dta[u'销量'],lags=10,ax=ax2)
✓ 其中lags 表示滞后的阶数，以上分别得到acf 图和pacf 图

定阶

- 通过两图观察得到：
- * 自相关图显示滞后有2个阶超出了置信边界；
- * 偏相关图显示在滞后1阶时的偏自相关系数超出了置信边界，从lag 1之后偏自相关系数值缩小至0



定阶

- 则有以下模型可以供选择：

a) ARMA(0,2)模型：即自相关图在滞后2阶之后缩小为0，且偏自相关缩小至0，则是一个阶数q=2的移动平均模型；

b) ARMA(1,0)模型：即偏自相关图在滞后1阶之后缩小为0，且自相关缩小至0，则是一个阶层p=1的自回归模型；

c) 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，则是一个阶层q=1的自回归模型。

- 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。

定阶

- 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

定阶

- 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。

定阶实现代码

- 对三个模型分别做AIC，BIC，HQ统计量检验：

```
#模型
arma_mod20 = sm.tsa.ARMA(dta,(2,0)).fit()
print(arma_mod20.aic,arma_mod20.bic,arma_mod20.hqic)
arma_mod01 = sm.tsa.ARMA(dta,(0,1)).fit()
print(arma_mod01.aic,arma_mod01.bic,arma_mod01.hqic)
arma_mod10 = sm.tsa.ARMA(dta,(1,0)).fit()
print(arma_mod10.aic,arma_mod10.bic,arma_mod10.hqic)
#result:
print(arma_mod20.aic,arma_mod20.bic,arma_mod20.hqic)
  420.440 748036 426.77482379 422.651510127
print(arma_mod01.aic,arma_mod01.bic,arma_mod01.hqic)
  417.759525387 422.510082203 419.417596956
print(arma_mod10.aic,arma_mod10.bic,arma_mod10.hqic)
  418.877719334 423.628276149 420.535790902
```

- 对比3个模型，可以看到ARMA(0,1)的aic，bic，hqic均最小，因此是最佳模型

目录

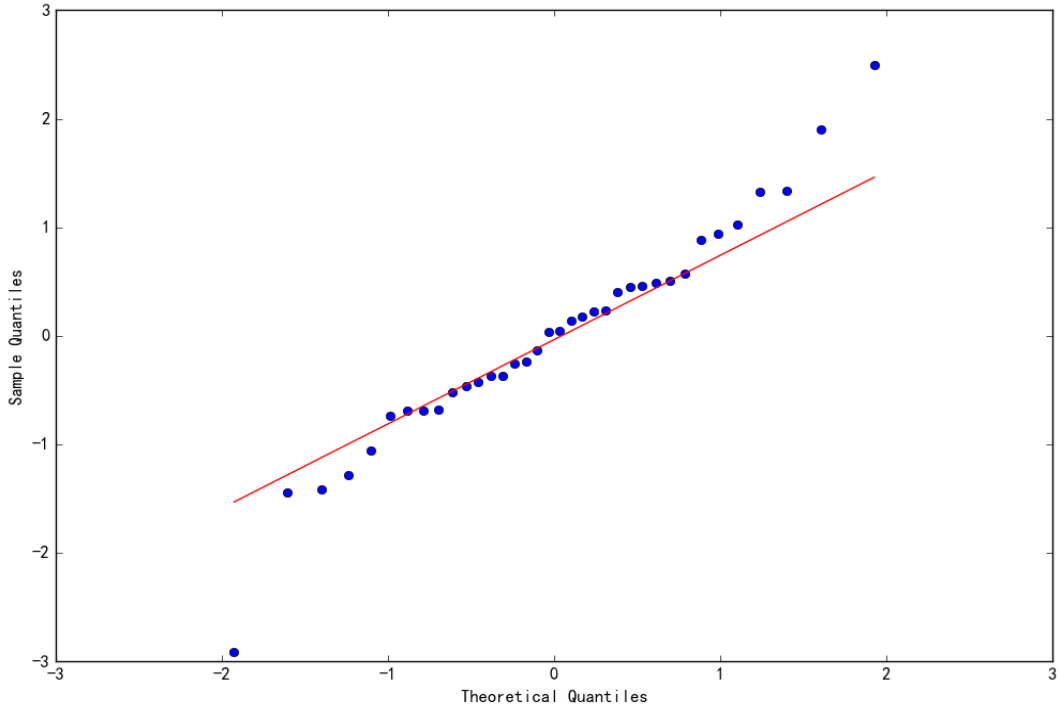
正态性检验

- 对于选择的模型，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

正态性检验

- #残差QQ图

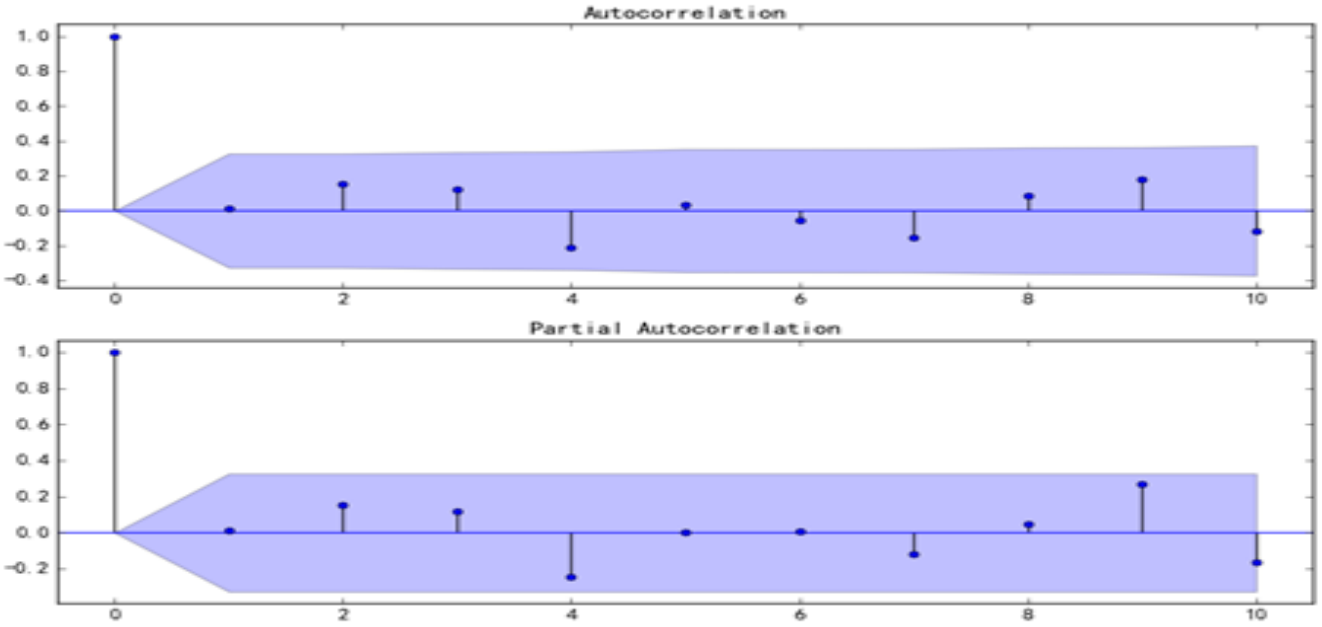
✓ resid = arma_mod01.resid
✓ fig = plt.figure(figsize=(12,8))
✓ ax = fig.add_subplot(111)
✓ fig = qqplot(resid, line='q', ax=ax, fit=True)



自相关性检验

- 对ARMA(0,1)模型所产生的残差做自相关图

✓ #残差自相关检验
✓ fig = plt.figure(figsize=(12,8))
✓ ax1 = fig.add_subplot(211)
✓ fig = sm.graphics.tsa.plot_acf(arma_mod01.resid.values.squeeze(), lags=10, ax=ax1)
✓ ax2 = fig.add_subplot(212)
✓ fig = sm.graphics.tsa.plot_pacf(arma_mod01.resid, lags=10, ax=ax2)



D-W检验

- 还需要对残差做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，但它只适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进行检验。

D-W检验

- ✓ #D-W检验
- ✓ print(sm.stats.durbin_watson(arma_mod01.resid.values))
- ✓ #result:1.95414900233

- 检验结果是1.95414900233，说明不存在自相关性。
- 最后还需要对残差做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。

L-B检验

- 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。
- L-B检验代码如下：

```
# Ljung-Box检验
import numpy as np
r,q,p = sm.tsa.acf(resid.values.squeeze(), qstat=True)
datap = np.c_[range(1,36), r[1:], q, p]
table = pd.DataFrame(datap, columns=['lag', "AC", "Q", "Prob(>Q)"])
print(table.set_index('lag'))
```

结果分析

- 部分检验结果如下：

#result:

lag	AC	Q	Prob(>Q)
1.0	0.009994	0.003904	0.950179
2.0	0.151097	0.922489	0.630498
3.0	0.119392	1.513404	0.679180
4.0	-0.212564	3.445002	0.486290
5.0	0.034075	3.496239	0.623957
6.0	-0.053349	3.626021	0.727135
7.0	-0.157088	4.790085	0.685562
8.0	0.082868	5.125590	0.744072
9.0	0.180436	6.775153	0.660516
10.0	-0.119683	7.528822	0.674754
11.0	0.051306	7.672864	0.742274
12.0	-0.062678	7.896792	0.793143
13.0	-0.020659	7.922177	0.848633

- 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。

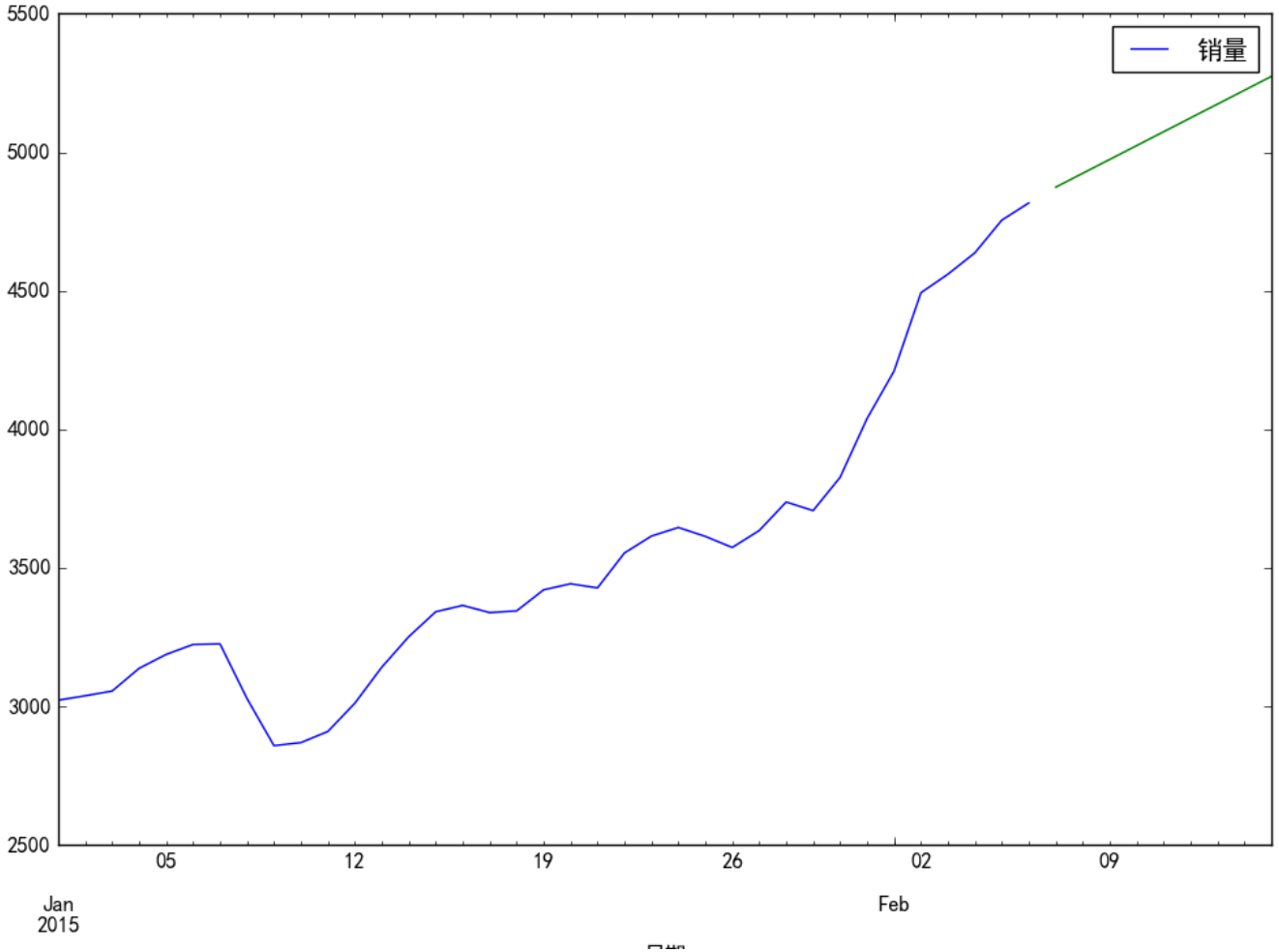
预测代码

- 模型确定之后，就可以开始进行预测了，我们对未来9日的数据进行预测，代码如下:

```
#预测
predict_sunspots = arma_mod01.predict('2015-2-07', '2015-2-15', dynamic=True)
fig, ax = plt.subplots(figsize=(12, 8))
print(predict_sunspots)
predict_sunspots[0] += data['2015-02-06':][u'销量']
data=pd.DataFrame(data)
for i in range(len(predict_sunspots)-1):
    predict_sunspots[i+1]=predict_sunspots[i]+predict_sunspots[i+1]
print(predict_sunspots)
ax = data.ix['2015':].plot(ax=ax)
predict_sunspots.plot(ax=ax)
plt.show()
```

预测结果

- 模型预测结果图如下：



Thank You!