学号：19072302008 姓名：饶华城 专业：应用统计

**[习题4.2]** 某城市过去四年每个月人口净流入数量如表4-25所示（行数据）。

表4-25

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 4.101 | 3.297 | 3.533 | 5.687 | 6.778 | 4.873 | 3.592 | 3.973 | 2.731 | 3.557 | 2.863 | 4.170 |
| 4.225 | 2.581 | 1.965 | 4.257 | 4.373 | 3.573 | 3.320 | 2.257 | 3.110 | 4.574 | 5.328 | 2.645 |
| 2.859 | 3.721 | 3.836 | 2.417 | 3.074 | 3.483 | 3.847 | 3.250 | 3.735 | 4.842 | 3.564 | 3.109 |
| 2.463 | 1.778 | 1.450 | 1.956 | 2.196 | 4.584 | 3.715 | 1.853 | 2.543 | 2.123 | 2.756 | 3.690 |

（1）绘制该序列的时序图。

Python代码：

import numpy as np

import pandas as pd

import pylab as plt

#写入数据

dta = np.array([4.101,3.297,3.533,5.687,6.778,4.873,3.592,3.973,2.731,3.557,2.863,4.170,

4.225,2.581,1.965,4.257,4.373,3.573,3.320,2.257,3.110,4.574,5.328,2.645,

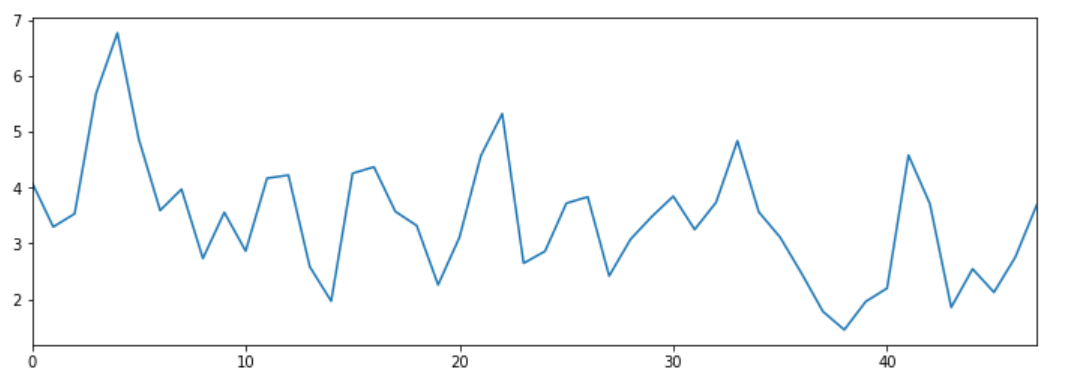
2.859,3.721,3.836,2.417,3.074,3.483,3.847,3.250,3.735,4.842,3.564,3.109,

2.463,1.778,1.450,1.956,2.196,4.584,3.715,1.853,2.543,2.123,2.756,3.690])

data = pd.Series(dta)

# 画时间序列线图

data.plot(figsize=[12,4])



绘制的时间序列线图如上所示。

（2）判断该序列的平稳性和纯随机性。

Python代码：

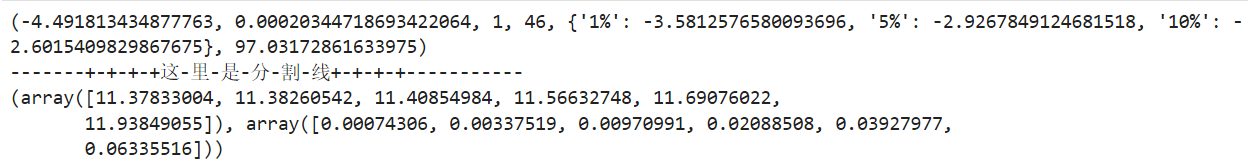
from statsmodels.stats.diagnostic import unitroot\_adf # 平稳性检验

from statsmodels.stats.diagnostic import acorr\_ljungbox #白噪声检验

print(unitroot\_adf(data))

print(""-------+-+-+-+这-里-是-分-割-线+-+-+-+-----------"")

print(acorr\_ljungbox(data,lags=1))



分析代码运行结果，该时间序列在进行一阶差分计算后，p值为0.000203小于给定显著性水平，认为此时平稳。在白噪声检验中，之后阶数为5阶的时候p值小于给定的显著性水平，认为此时为非白噪声数据。综述可得，该序列为平稳非白噪声数据，可以使用ARMA模型来拟合预测。

（3）考察该序列的自相关系数和偏自相关系数的性质。

Python代码：

import statsmodels.api as sm

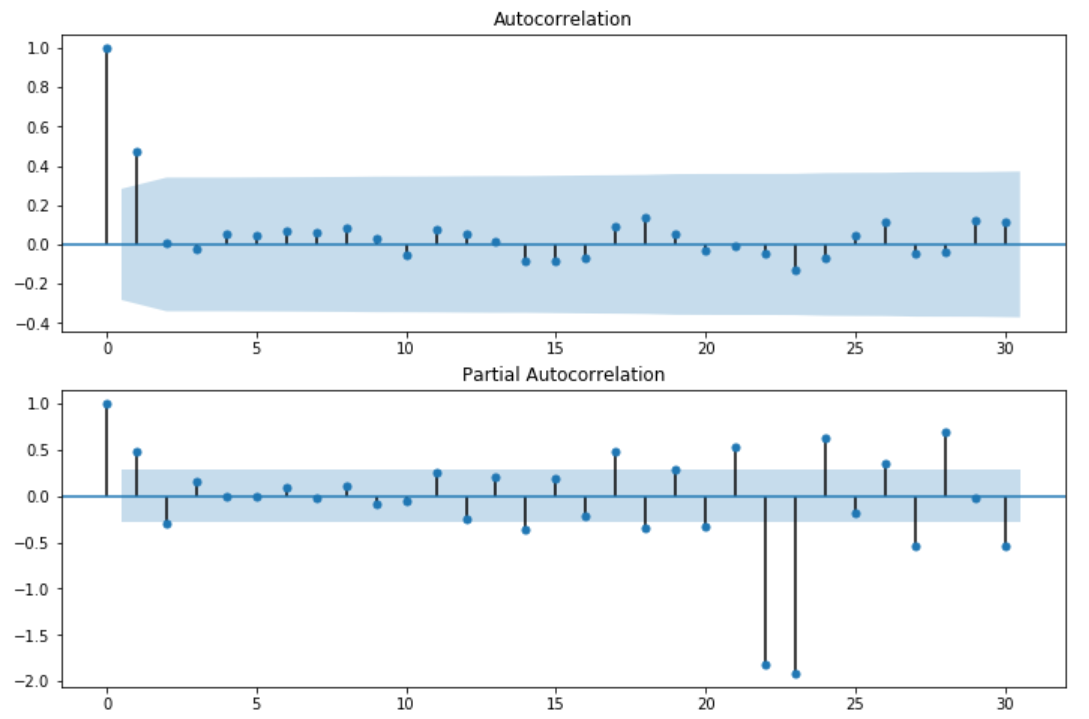
fig = plt.figure(figsize=(12,8))

ax1 = fig.add\_subplot(211)

ax2 = fig.add\_subplot(212)

fig = sm.graphics.tsa.plot\_acf(data.values.squeeze(),lags=30,ax=ax1)

fig = sm.graphics.tsa.plot\_pacf(data,lags=30,ax=ax2)



代码运行结果，如上图所示。

（4）选择合适的模型拟合该序列的发展。

分析第（3）们自相关和偏自相关图，自相关图拖尾，偏自相关图拖尾，因此，我们先假设使用ARMA(1,1)模型来拟合该序列的发展。

Python代码：

# 根据bic准则编写函数来推荐最优的模型：

def SelectModel(data):

best\_model = sm.tsa.ARMA(data,(0,0)).fit()

for i in range(2):

for j in range(2):

arma\_mod = sm.tsa.ARMA(data,(i,j)).fit()

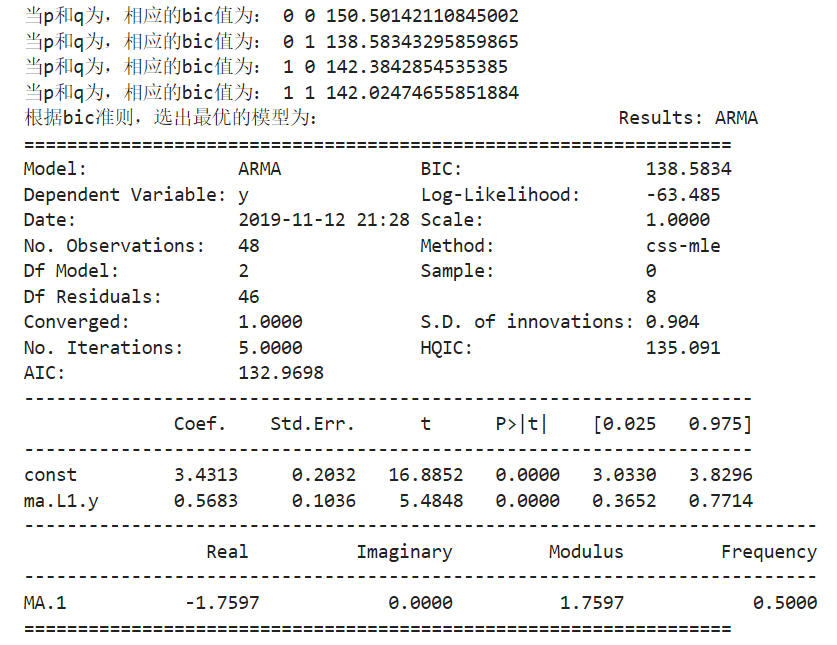
print('当p和q为，相应的bic值为：',i,j,arma\_mod.bic)

if best\_model.bic > arma\_mod.bic :

best\_model = arma\_mod

print('根据bic准则，选出最优的模型为：',best\_model.summary2())

SelectModel(data)



根据BIC准则推荐的模型为MA(1),并且模型MA(1)模型的参数通过显著性检验，因此在该问题中，我们应该使用MA(1)模型来拟合序列的发展值合适。接下来对模型进行检验：

Python代码：

# 模型的检验

# 原假设是，序列剩下的没被用来预测的部分的信息为白噪声

# 若不拒绝原假设，则通过检验，模型显著

from scipy import stats

resid = arma\_mod.resid

stats.normaltest(resid)



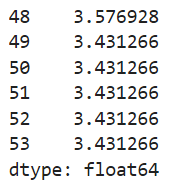
代码运行结果如上图所示，由于p值大于0.05不拒绝原假设，故模型显著。

（5）利用拟合模型预测该城市未来5年的人口净流入情况。

Python代码：

predict\_number = arma\_mod.predict(len(data),len(data)+5)

print(predict\_number)



代码运行结果如上所示，未来5期的数据如上图所示。

**[习题4.4]** 某城市过去45年中每年的人口死亡率（单位：千分之一），如下表所示（行数据）。

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 3.665 | 4.247 | 4.674 | 3.669 | 4.752 | 4.785 | 5.929 | 4.468 | 5.102 | 4.831 | 6.889 | 5.337 |
| 5.086 | 5.603 | 4.153 | 4.945 | 5.726 | 4.965 | 1.820 | 3.723 | 5.663 | 4.739 | 4.845 | 4.535 |
| 4.774 | 5.962 | 6.614 | 5.255 | 5.355 | 6.144 | 5.590 | 4.388 | 3.447 | 4.615 | 6.032 | 5.740 |
| 4.391 | 3.128 | 3.436 | 4.964 | 6.332 | 7.665 | 5.277 | 4.904 | 4.830 |  |  |  |

（1）绘制该序列时序图。

Python代码：

import numpy as np

import pandas as pd

import pylab as plt

dta = np.array([3.665,4.247,4.674,3.669,4.752,4.785,5.929,4.468,5.102,4.831,6.889,5.337,

5.086,5.603,4.153,4.945,5.726,4.965,1.820,3.723,5.663,4.739,4.845,4.535,

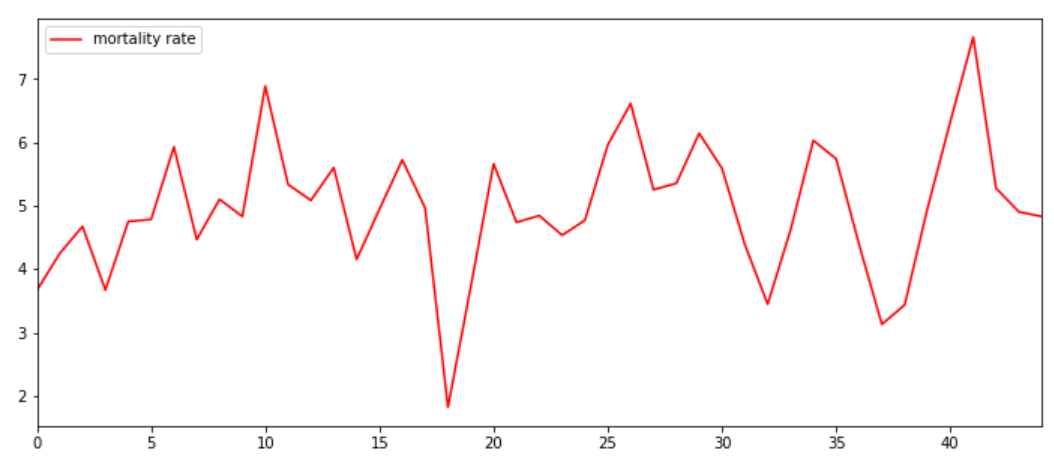
4.774,5.962,6.614,5.255,5.355,6.144,5.590,4.388,3.447,4.615,6.032,5.740,

4.391,3.128,3.436,4.964,6.332,7.665,5.277,4.904,4.830])

data = pd.Series(dta,name='mortality rate')

data.plot(figsize=[12,5],colors='r')

plt.legend()



程序运行结果如上图所示。

（2）判断该序列的平稳性和纯随机性。

Python代码：

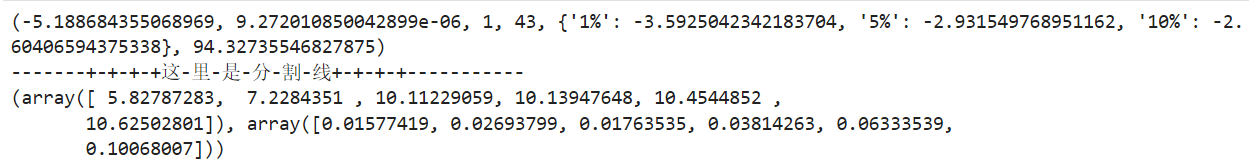
from statsmodels.stats.diagnostic import unitroot\_adf # 平稳性检验

from statsmodels.stats.diagnostic import acorr\_ljungbox #白噪声检验

print(unitroot\_adf(data))

print("-------+-+-+-+这-里-是-分-割-线+-+-+-+-----------")

print(acorr\_ljungbox(data,lags=1))



分析代码运行结果，该时间序列在进行一阶差分计算后，p值为远远小于给定显著性水平，认为此时平稳。在白噪声检验中，之后阶数为前四阶对于的p值为小于给定的显著性水平，认为此时为非白噪声数据。综述可得，该序列为平稳非白噪声数据，可以使用ARMA模型来拟合预测。

（3）考察该序列的自相关系数和偏自相关系数的性质。

Python代码：

import statsmodels.api as sm

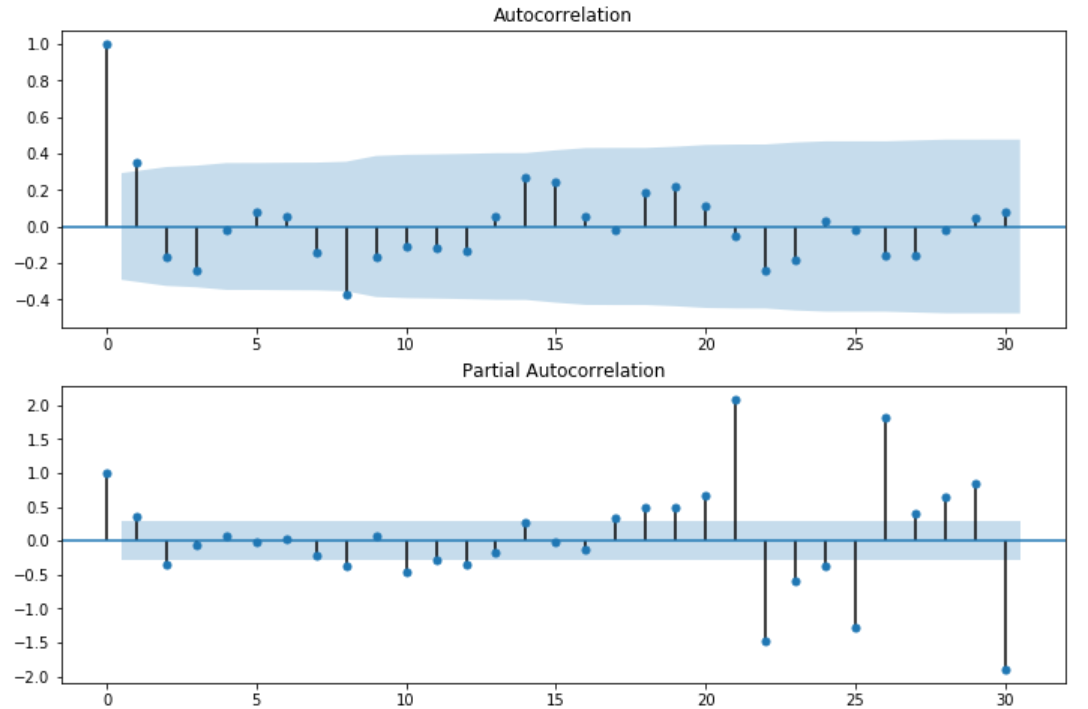
fig = plt.figure(figsize=(12,8))

ax1 = fig.add\_subplot(211)

ax2 = fig.add\_subplot(212)

fig = sm.graphics.tsa.plot\_acf(data.values.squeeze(),lags=30,ax=ax1)

fig = sm.graphics.tsa.plot\_pacf(data,lags=30,ax=ax2)



（4）尝试多个模型拟合该序列的发展，并考察该序列的拟合模型的优化问题。

分析第（3）们自相关和偏自相关图，自相关图截尾，偏自相关图拖尾，因此，我们先假设使用AR(1)模型来拟合该序列的发展。

Python代码：

# 根据bic准则编写函数来推荐最优的模型：

def SelectModel(data):

best\_model = sm.tsa.ARMA(data,(0,0)).fit()

for i in range(2):

for j in range(2):

arma\_mod = sm.tsa.ARMA(data,(i,j)).fit()

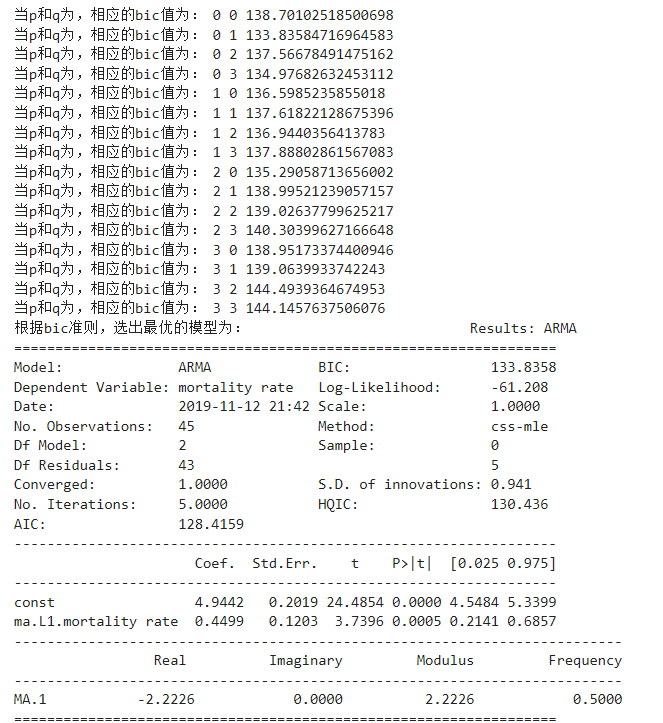
print('当p和q为，相应的bic值为：',i,j,arma\_mod.bic)

if best\_model.bic > arma\_mod.bic :

best\_model = arma\_mod

print('根据bic准则，选出最优的模型为：',best\_model.summary2())

SelectModel(data)



根据BIC准则推荐的模型为MA(1),并且模型MA(1)模型的参数通过显著性检验，因此在该问题中，我们应该使用MA(1)模型来拟合序列的发展值合适。接下来对模型进行检验：

Python代码：

# 模型的检验

# 原假设是，序列剩下的没被用来预测的部分的信息为白噪声

# 若不拒绝原假设，则通过检验，模型显著

from scipy import stats

resid = arma\_mod.resid

stats.normaltest(resid)



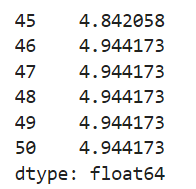
代码运行结果如上图所示，由于p值大于0.05不拒绝原假设，故模型显著。

（5）利用最优的拟合模型预测该城市未来5年的人口死亡率情况。

Python代码：

predict\_number = arma\_mod.predict(len(data),len(data)+5)

print(predict\_number)

****

代码运行结果如上所示，未来5期的数据如上图所示。

**[习题4.6]** 某城市过去63年中每年降雪量数据如表4-28所示（行数据）。

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 126.4 | 82.4 | 78.1 | 51.1 | 90.9 | 76.2 | 104.5 | 87.4 |
| 110.5 | 25 | 69.3 | 53.5 | 39.8 | 63.6 | 46.7 | 72.9 |
| 79.6 | 83.6 | 80.7 | 60.3 | 79 | 74.4 | 49.6 | 54.7 |
| 71.8 | 49.1 | 103.9 | 51.6 | 82.4 | 83.6 | 77.8 | 79.3 |
| 89.6 | 85.5 | 58 | 120.7 | 110.5 | 65.4 | 39.9 | 40.1 |
| 88.7 | 71.4 | 83 | 55.9 | 89.9 | 84.8 | 105.2 | 113.7 |
| 124.7 | 114.5 | 115.6 | 102.4 | 101.4 | 89.8 | 71.5 | 70.9 |
| 98.3 | 55.5 | 66.1 | 78.4 | 120.5 | 97 | 110 |  |

1. 判断该序列的平稳性与纯随机性。

import numpy as np

import pandas as pd

import pylab as plt

dta = [126.4,82.4,78.1,51.1,90.9,76.2,104.5,87.4,

110.5,25,69.3,53.5,39.8,63.6,46.7,72.9,

79.6,83.6,80.7,60.3,79,74.4,49.6,54.7,

71.8,49.1,103.9,51.6,82.4,83.6,77.8,79.3,

89.6,85.5,58,120.7,110.5,65.4,39.9,40.1,

88.7,71.4,83,55.9,89.9,84.8,105.2,113.7,

124.7,114.5,115.6,102.4,101.4,89.8,71.5,70.9,

98.3,55.5,66.1,78.4,120.5,97,110]

data = pd.Series(dta,name='snowfall')

data.plot(figsize=[12,5],colors='g')

plt.legend()

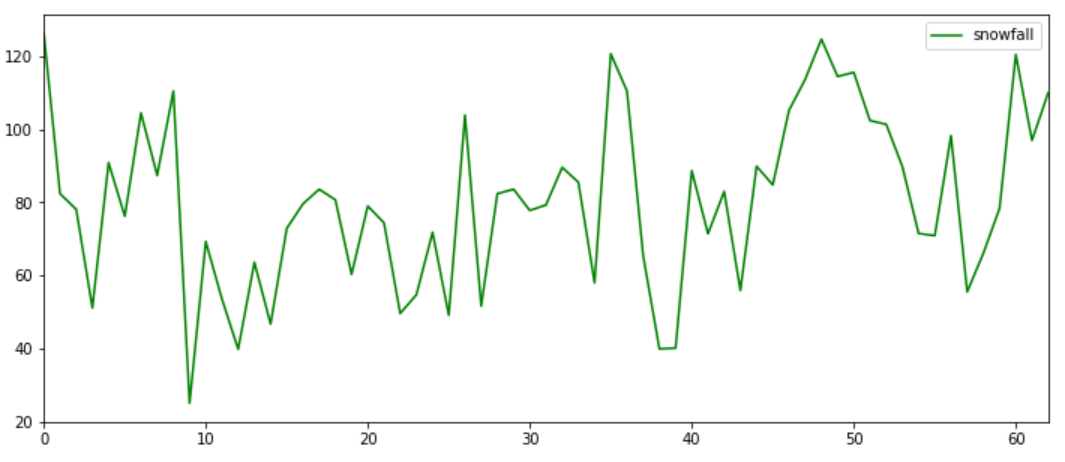
from statsmodels.stats.diagnostic import unitroot\_adf # 平稳性检验

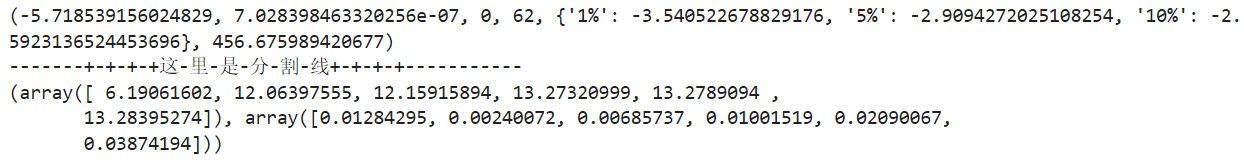
from statsmodels.stats.diagnostic import acorr\_ljungbox #白噪声检验

print(unitroot\_adf(data))

print("-------+-+-+-+这-里-是-分-割-线+-+-+-+-----------")

print(acorr\_ljungbox(data,lags=6))





分析程序运行结果可知，该数据为平稳非白噪声数据。故适合使用ARMA模型来拟合数据。

1. 如果序列平稳且非白噪声，选择适当的模型拟合该序列的发展。

Python代码：

import statsmodels.api as sm

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

ax1 = fig.add\_subplot(211)

ax2 = fig.add\_subplot(212)

fig = sm.graphics.tsa.plot\_acf(data.values.squeeze(),lags=30,ax=ax1)

fig = sm.graphics.tsa.plot\_pacf(data,lags=30,ax=ax2)

# 根据bic准则编写函数来推荐最优的模型：

def SelectModel(data):

best\_model = sm.tsa.ARMA(data,(0,0)).fit()

for i in range(3):

for j in range(3):

arma\_mod = sm.tsa.ARMA(data,(i,j)).fit()

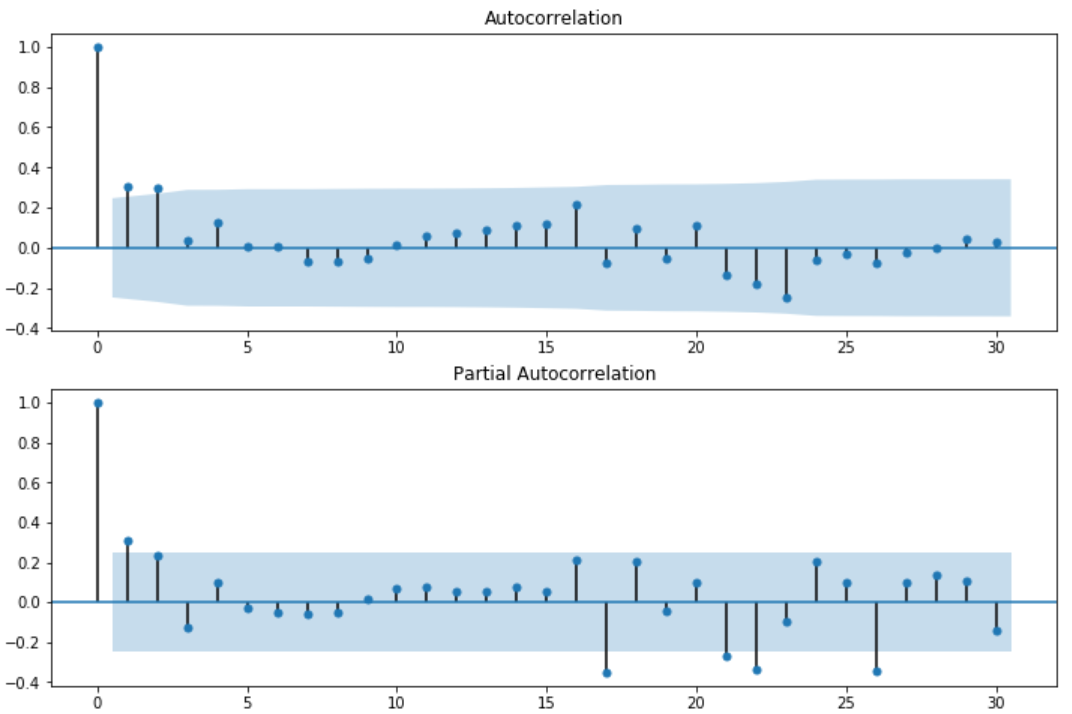
print('当p和q为，相应的bic值为：',i,j,arma\_mod.bic)

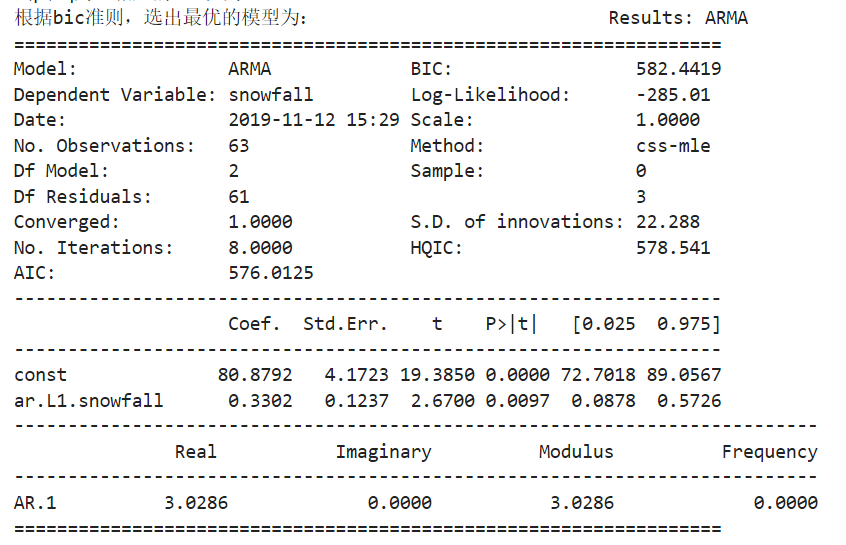
if best\_model.bic > arma\_mod.bic :

best\_model = arma\_mod

print('根据bic准则，选出最优的模型为：',best\_model.summary2())

SelectModel(data)

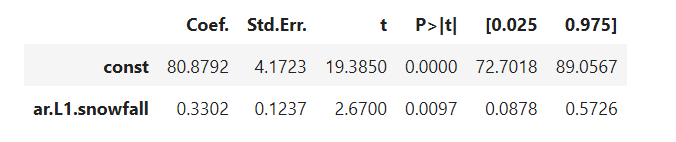




Python代码：

arma\_mod = sm.tsa.ARMA(data,(1,0)).fit()

arma\_mod.summary2()



因为AR(1)模型的各项参数都通过了显著性检验，因此该问题适合使用AR（1）来拟合序列的发展。

1. 利用拟合模拟预测该城市未来五年的降雪量。

**[习题4.8]** 现有201个连续的生产记录，如下所示：

81.9,89.4,79.0,81.4,84.8,85.9,88.0,80.3,82.6,83.5,80.2,85.2,87.2,83.5,84.3,82.9,84.7,82.9

81.5,83.4,87.7,81.8,79.6,85.8,77.9,89.7,85.4,86.3,80.7,83.8,90.5,84.5,82.4,86.7,83.0,81.8

89.3,79.3,82.7,88.0,79.6,87.8,83.6,79.5,83.3,88.4,86.6,84.6,79.7,86.0,84.2,83.0,84.8,83.6 81.8,85.9,88.2,83.5,87.2,83.7,87.3,83.0,90.5,80.7,83.1,86.5,90.0,77.5,84.7,84.6,87.2,80.5

86.1,82.6,85.4,84.7,82.8,81.9,83.6,86.8,84.0,84.2,82.8,83.0,82.0,84.7,84.4,88.9,82.4,83.0

85.0,82.2,81.6,86.2,85.4,82.1,81.4,85.0,85.8,84.2,83.5,86.5,85.0,80.4,85.7,86.7,86.7,82.3

86.4,82.5,82.0,79.5,86.7,80.5,91.7,81.6,83.9,85.6,84.8,78.4,89.9,85.0,86.2,83.0,85.4,84.4

84.5,86.2,85.6,83.2,85.7,83.5,80.1,82.2,88.6,82.0,85.0,85.2,85.3,84.3,82.3,89.7,84.8,83.1

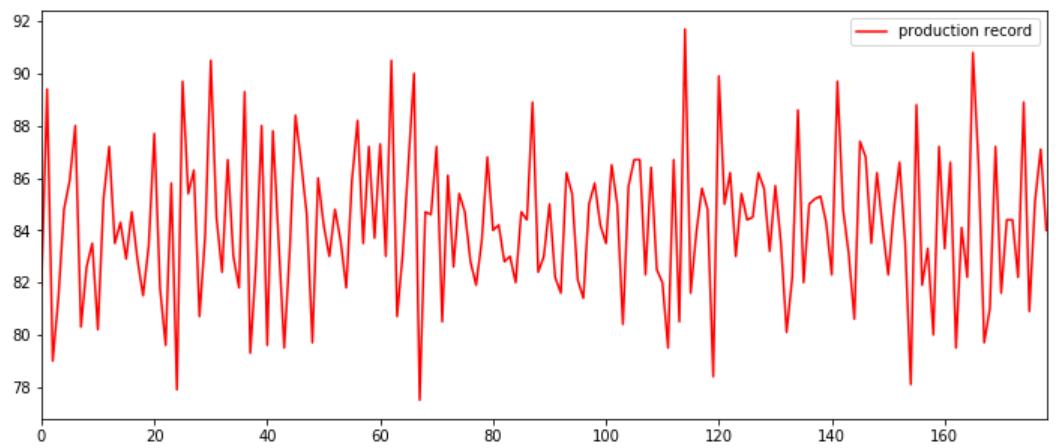
80.6,87.4,86.8,83.5,86.2,84.1,82.3,84.8,86.6,83.5,78.1,88.8,81.9,83.3,80.0,87.2,83.3,86.6

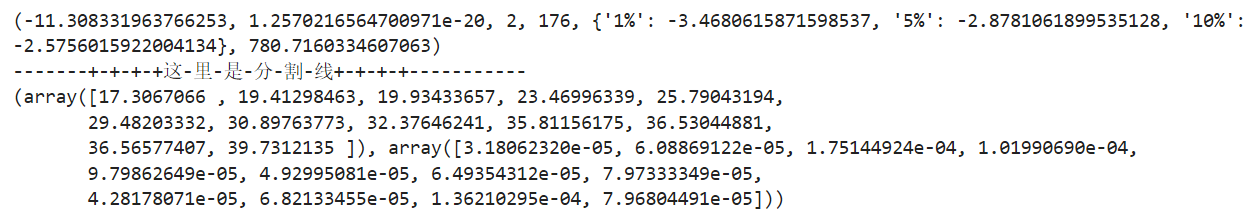
79.5,84.1,82.2,90.8,86.5,79.7,81.0,87.2,81.6,84.4,84.4,82.2,88.9,80.9,85.1,87.1,84.0

（1）判断该序列的平稳性与纯随机性。

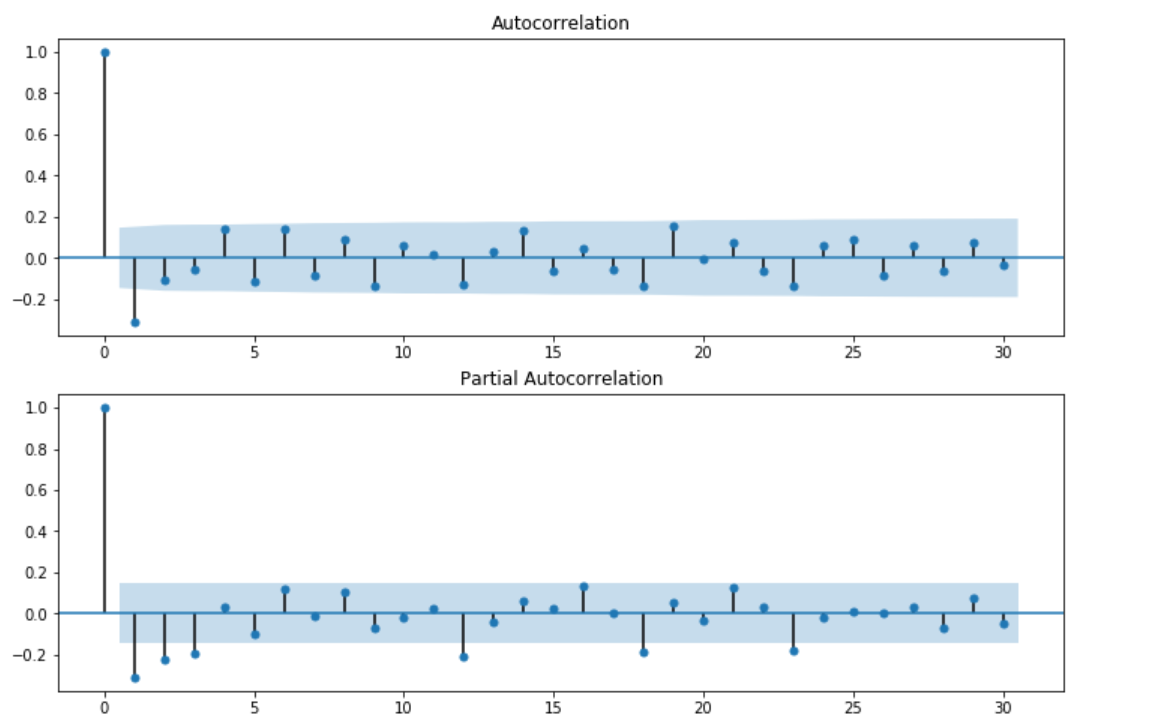
由于代码都是重复的，因此在之后的题目当中，重复的代码我将会省去。

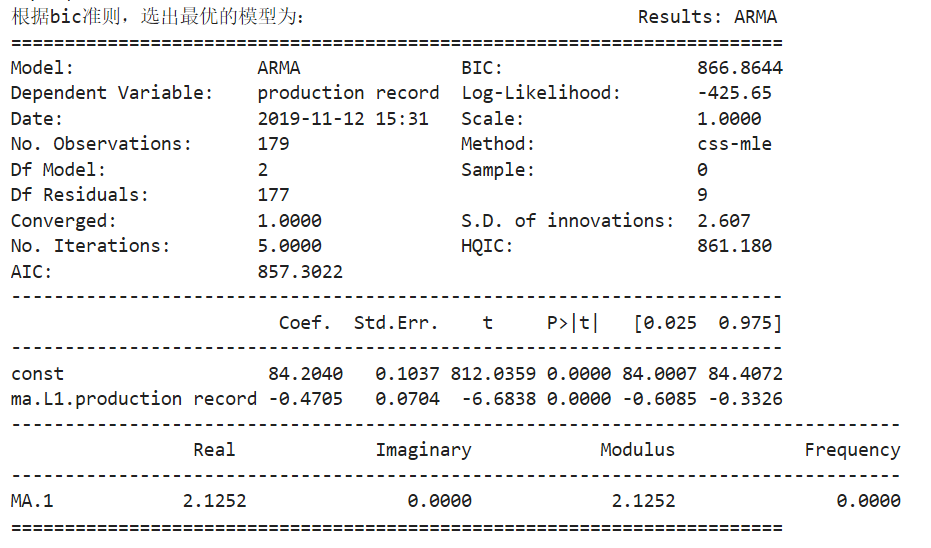
程序运行结果如下所示，根据ADF和白噪声检验的结果，我们认为该序列为平稳非白噪声数据。





（2）如果序列平稳且非白噪声，选择适当的模型拟合该序列的发展。



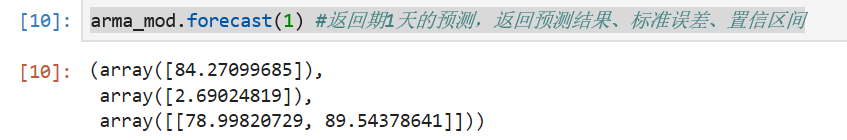


根据程序运行结果，BIC准则推荐使用AR（1）模型，并且AR（1）模型的各项参数通过显著性检验，因此我们使用AR(1)模型来拟合该时间序列。

（3）利用拟合模型预测该序列下一时刻95%的置信区间。

Python代码：

arma\_mod.forecast(1) #返回期1天的预测，返回预测结果、标准误差、置信区间



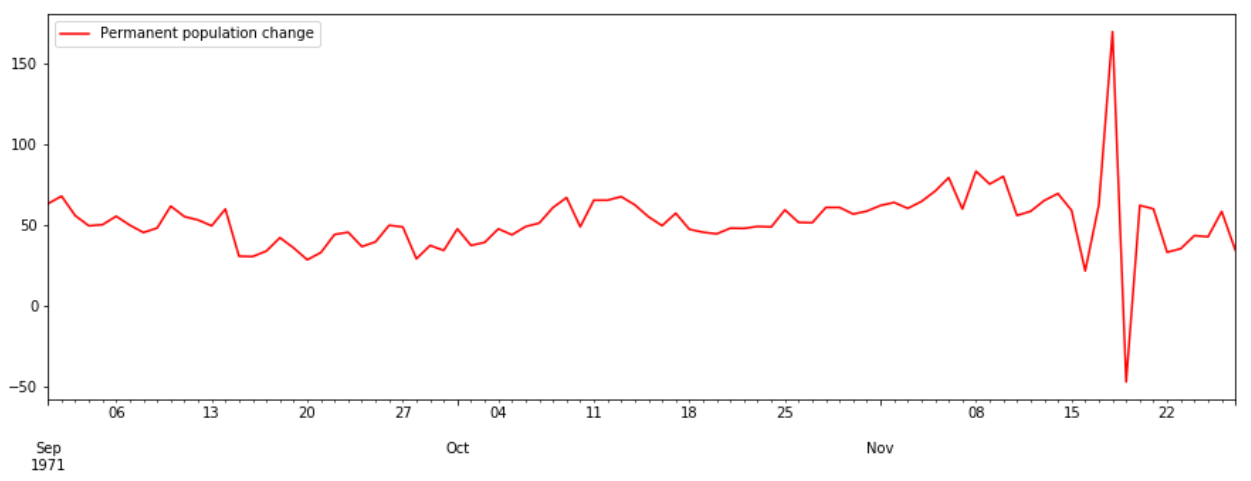
分析程序运行结果可得，下一时期95%的置信区间为[79，90]。

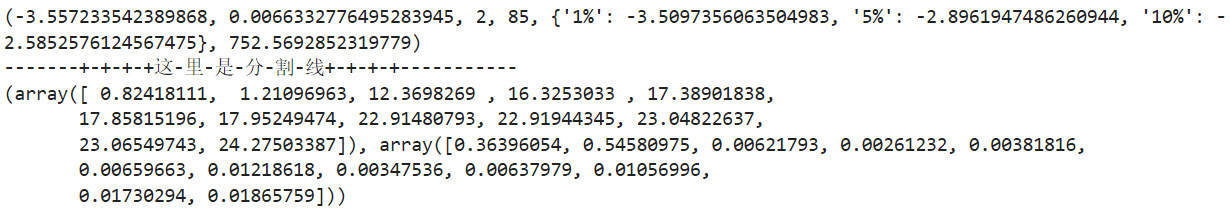
**[习题4.9]** 1971年9月至1993年6月澳大利亚季度常住人口变动情况如表4-31所示（行数据）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 63.2 | 67.9 | 55.8 | 49.5 | 50.2 | 55.4 |
| 49.9 | 45.3 | 48.1 | 61.7 | 55.2 | 53.1 |
| 49.5 | 59.9 | 30.6 | 30.4 | 33.8 | 42.1 |
| 35.8 | 28.4 | 32.9 | 44.1 | 45.5 | 36.6 |
| 39.5 | 49.8 | 48.8 | 29 | 37.3 | 34.2 |
| 47.6 | 37.3 | 39.2 | 47.6 | 43.9 | 49 |
| 51.2 | 60.8 | 67 | 48.9 | 65.4 | 65.4 |
| 67.6 | 62.5 | 55.1 | 49.6 | 57.3 | 47.3 |
| 45.5 | 44.5 | 48 | 47.9 | 49.1 | 48.8 |
| 59.4 | 51.6 | 51.4 | 60.9 | 60.9 | 56.8 |
| 58.6 | 62.1 | 64 | 60.3 | 64.6 | 71 |
| 79.4 | 59.9 | 83.4 | 75.4 | 80.2 | 55.9 |
| 58.5 | 65.2 | 69.5 | 59.1 | 21.5 | 62.5 |
| 170 | -47.4 | 62.2 | 60 | 33.1 | 35.3 |
| 43.4 | 42.7 | 58.4 | 34.4 |  |  |

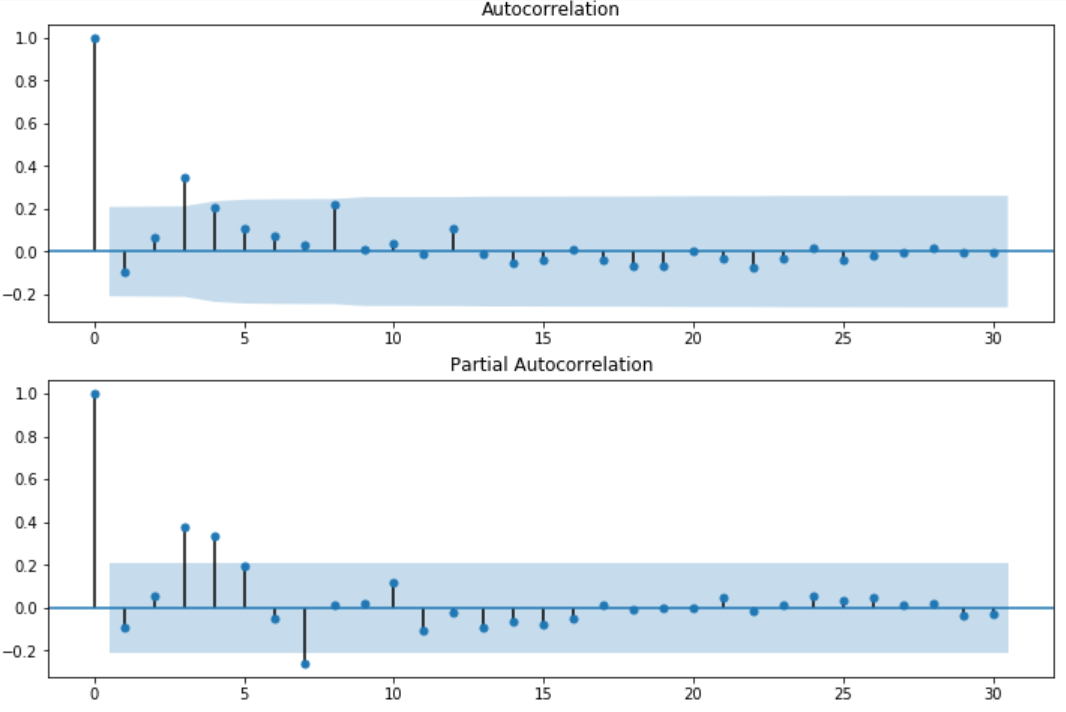
（1）判断该序列的平稳性与纯随机性。

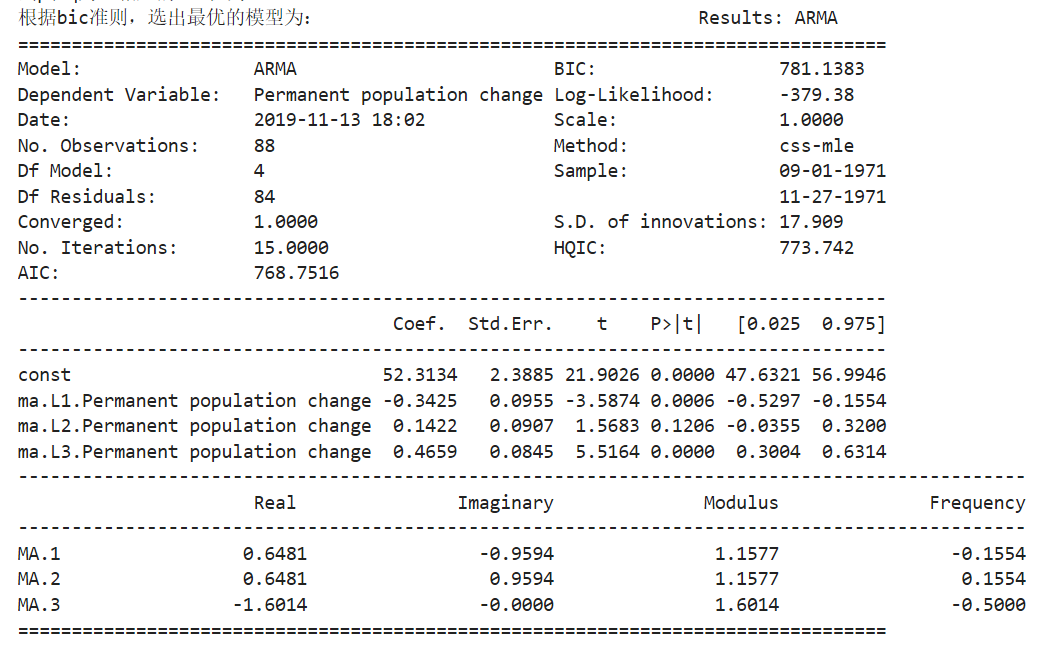
程序运行结果如下所示，根据ADF和白噪声检验的结果，我们认为该序列为平稳非白噪声数据。





（2）如果序列平稳且非白噪声，选择适当的模型拟合该序列的发展。





（3）绘制该序列的拟合图以及未来5年的预测图。

Python代码：

predict\_number = arma\_mod.predict(0,len(data)+5)

plt.plot(data,color='r',label='Permanent population change')

plt.plot(predict\_number,color='g',label = 'predict')

plt.rcParams['figure.figsize'] = (25.0, 4.0)

plt.legend(loc='best')

