

## 第七章 自回归

表7.5中国1980-2016年固定资产投资Y与社会零售总额X数据（单位：亿元）

```
import pandas as pd
import numpy as np
import matplotlib
import matplotlib.pyplot as plt

data = pd.read_csv('chapter7.csv', index_col='year')
#data.index = pd.to_datetime(data.index)
ts = data['Y']
from statsmodels.tsa.arima_model import ARMA, ARIMA
from pandas.plotting import register_matplotlib_converters
register_matplotlib_converters()
```

查看数据

```
ts.describe()
```

```
count      37.000000
mean       72285.251351
std        92455.891506
min         2140.000000
25%         8101.400000
50%        33378.100000
75%        89210.000000
max        332316.300000
Name: Y, dtype: float64
```

```
from statsmodels.tsa.arima_model import ARMA
from itertools import product
ps = range(0, 3)
qs = range(0, 3)
parameters = product(ps, qs)
parameters_list = list(parameters)

best_aic = float('inf')
results = []
# 搜索最优参数 确定模型
```

```

for param in parameters_list:
    try:
        model = ARMA(ts, order=(param[0], param[1])).fit()
    except ValueError:
        print("参数错误: ", param)
        continue
    aic = model.aic
    if aic < best_aic:
        best_model = model
        best_aic = model.aic
        best_param = param
    results.append([param, model.aic])
results_table = pd.DataFrame(results)
results_table.columns = ['parameters', 'aic']
print("最优模型", best_model.summary())

```

参数错误: (0, 2)

参数错误: (1, 1)

参数错误: (1, 2)

参数错误: (2, 1)

参数错误: (2, 2)

最优模型

ARMA Model Results

```

=====
Dep. Variable:          Y    No. Observations:          37
Model:                ARMA(1, 0)    Log Likelihood          -408.828
Method:                css-mle    S.D. of innovations      14205.997
Date:                Sat, 18 Dec 2021    AIC          823.657
Time:                00:12:01    BIC          828.490
Sample:                0    HQIC          825.361
=====

              coef      std err          z      P>|z|      [0.025      0.975]
-----
const          7.229e+04    2.51e+05      0.288      0.775    -4.19e+05    5.64e+05
ar.L1.Y          0.9970      0.006    166.320      0.000      0.985      1.009
=====
Roots
=====
              Real          Imaginary          Modulus          Frequency
-----
AR.1          1.0030          +0.0000j          1.0030          0.0000
=====

```

## 运行及预测

```

from pandas.plotting import register_matplotlib_converters
register_matplotlib_converters()

arma = ARMA(ts,order=(0, 1)).fit(dis = -1)
ts_predict_arma = arma.predict()
plt.plot(ts,label = 'ORG')
plt.plot(ts_predict_arma[:],label = 'PRE')
plt.title("ARMA(0,1)")
plt.legend()
plt.show()

```

