**Prediction Model Method**

Auto-regressive Integrated Moving Average model (ARIMA) was applied in the prediction process. It is specified by three main components, P, D and Q. P stands for auto-regression, representing the number of lag observations in the model; D stands for integration, representing the number of times that input raw data are differenced, making the model stationary; Q stands for moving average, representing the size of moving average window applied to lagged observations. For non-stationary data, to eliminate the non-stationarity, differencing step will be applied once or more. The model was established by Python. First, analyze the data and compute the difference between consecutive data to choose the P, which makes the data stationary. Then, loop the steps “p<data.size/10,q<data.size” to generate all combination of P and Q, and choose the parameter with the minimum Bayesian Information Criterion. Finally, generate prediction. The code is as follows:

#coding=gbk

import matplotlib.pyplot as plt

import pandas as pd

from statsmodels.tsa.arima\_model import ARIMA

from statsmodels.graphics.tsaplots import plot\_acf, plot\_pacf

filename = r'death+rate+data .csv'

data = pd.read\_csv(filename,usecols=['Year','Male'] ,index\_col = u'Year',dtype={'Male': float})

plt.rcParams['font.sans-serif'] = ['SimHei']

plt.rcParams['axes.unicode\_minus'] = False

init\_d= 1

D\_data = data.diff(init\_d).dropna()

D\_data.columns = [u'Male']

pmax = int(len(D\_data) / 10)

qmax = int(len(D\_data) / 10)

bic\_matrix = []

for p in range(pmax +1):

temp= []

for q in range(qmax+1):

try:

temp.append(ARIMA(data, (p, init\_d, q)).fit().bic)

except Exception:

temp.append(None)

bic\_matrix.append(temp)

bic\_matrix = pd.DataFrame(bic\_matrix)

p,q = bic\_matrix.stack().idxmin()

model = ARIMA(data, (p,init\_d,q)).fit()

model.summary2()

forecast, fcasterr, conf\_int = model.forecast(20)

for x in forecast:

print x