SYDE 631: Time Series Modelling
Project Report

Department of Systems Design Engineering
University of Waterloo
Course Instructor: Professor Keith W. Hipel

Xin Chen
20705271
Systems Design Engineering
Master Degree

Rainfall Forecast Analysis in Toronto

Abstract: In this article, we analyzed the amount of rainfall over the past several years and forecasted the amount of that in the future by creating some time series models. Meanwhile, we wrote a program with python to generate these model diagrams.

Key words: rainfall, time series, python, forecast

Introduction

Rainfall refers to the depth of the water layer that falls from the sky to the ground and accumulates on the water without evaporation, infiltration, and loss. It is usually measured in millimeters, which can visually indicate the amount of rainfall. Rainfall is an important basis for the calculation of regional water resources, so it is necessary to find an accurate and simple method to predict rainfall.

Rainfall is an important disaster data with chaotic characteristics. Different departments have different standards for the classification of rainfall levels.

Meteorological department:

Rainfall is the depth of the water layer that falls to the ground within a certain period of time, expressed in millimeters. The rainfall intensity per unit time is called rainfall intensity. Rainfall intensity is divided by rainfall grade.

Flood control department:

Rainfall is the depth of the water layer that falls at a certain point on the ground or a certain unit area in a certain period of time, measured in millimeters. Usually any 24-hour accumulated rainfall exceeding 50 mm will be classified as heavy rain.

Hydrology department:

Generally speaking, light rain, moderate rain, heavy rain, heavy rain, etc., are generally measured by daily rainfall. Among them, the light rain refers to daily rainfall below 10 mm; the rainy day rainfall is 10~24.9 mm; the heavy rain rainfall is 25~49.9 mm; the heavy rainfall is 50~99.9 mm; the heavy rain rainfall is 100~250 mm; Heavy rain rainfall is above 250 mm.

Data source: Download the weather data about the amount of rainfall from http://www.meteomanz.com, from Jan, 2004 to Oct, 2018 in Toronto.

AR: In statistics and signal processing, an autoregressive (AR) model is a representation of a type of random process; as such, it is used to describe certain time-varying processes in nature, economics, etc. The autoregressive model specifies that the output variable depends linearly on its own previous values and on a stochastic term (an imperfectly predictable term); thus the model is in the form of a stochastic difference equation.

MA: In time series analysis, the moving-average model (MA model), also known as moving-average process, is a common approach for modeling univariate time series. The moving-average model specifies that the output variable depends linearly on the current and various past values of a stochastic (imperfectly predictable) term.

ARIMA: In statistics and econometrics, and in particular in time series analysis, an autoregressive integrated moving average (ARIMA) model is a generalization of an autoregressive moving average (ARMA) model. Both of these models are fitted to time series data either to better understand the data or to predict future points in the series (forecasting). ARIMA models are applied in some cases where data show evidence of non-stationarity, where an initial differencing step (corresponding to the "integrated" part of the model) can be applied one or more times to eliminate the non-stationarity.

The AR part of ARIMA indicates that the evolving variable of interest is regressed on its own lagged (i.e., prior) values. The MA part indicates that the regression error is actually a linear combination of error terms whose values occurred contemporaneously and at various times in the past. The I (for "integrated") indicates that the data values have been replaced with the difference between their values and the previous values (and this differencing process may have been performed more than once). The purpose of each of these features is to make the model fit the data as well as possible.

Non-seasonal ARIMA models are generally denoted ARIMA(p,d,q) where parameters p, d, and q are non-negative integers, p is the order (number of time lags) of the autoregressive model, d is the degree of differencing (the number of times the data have had past values subtracted), and q is the order of the moving-average model. Seasonal ARIMA models are usually denoted ARIMA(p,d,q)$(P,D,Q)_m$, where m refers to the number of

periods in each season, and the uppercase P,D,Q refer to the autoregressive, differencing, and moving average terms for the seasonal part of the ARIMA model.

Background

Precipitation is an important indicator of the degree of drought. It directly reflects changes in nature. The amount of precipitation directly affects agricultural production. If scientific and accurate predictions can be made on precipitation, relevant departments such as agriculture and water conservancy can take timely measures to prevent floods and droughts and reduce unnecessary losses. Therefore, precipitation prediction becomes an important research topic in current predictions. Many scholars at home and abroad have done extensive research on rainfall prediction. The traditional methods are probabilistic statistics, which represent Markov Markov chain model, grayscale GM (1,1) model and exponential smoothing. These methods are quantitative prediction models, which can perform multi-step prediction, but can only represent exponentially increasing rainfall, in practice, rainfall is not a purely exponential growth law, but a natural phenomenon with violent fluctuations, so its prediction results are not accurate.

In the past 10 years, many scholars have used the time series method to conduct in-depth research on rainfall, mainly including autoregressive moving average model (ARMA) and autoregressive (AR) models, which regard rainfall as a chronological data. The prediction results are significantly improved compared with the traditional methods. With the advent of the autoregressive integrated moving average model, ARIMA is a time series prediction model that combines the advantages of time series analysis and regression analysis. It can describe the rainfall time correlation characteristics and has been widely used in the field of rainfall time series. However, the ARIMA model is actually a variant of time series, so it is a prediction method based on linear data. Because rainfall is a non-stationary stochastic process that changes with time, it has complex and dynamic characteristics. It is affected by various random factors, so these methods cannot fully reflect the nonlinear law of rainfall changes, and the prediction results are sometimes not ideal. Since rainfall is a

non-stationary time series, it should be preprocessed when using the ARIMA model to predict the non-stationary time series data of rainfall into stationary time series data so that it can be based on a linearly stable rainfall prediction model to describe. Wavelet transform is a time-frequency domain analysis model, which is very suitable for the smoothing of non-stationary time series data of rainfall. The rainfall becomes the stationary time series data, and then the traditional time series prediction model is used to predict the decomposed time series and provide a new idea for the prediction of some non-stationary time series.

Through the study of rainfall over the past few years, we can predict the rainfall in the next few years, which has greatly promoted the study of future weather phenomena.

Exploratory Data Analysis

A general time series (non-seasonal time series) contains a trend part and an irregular part (that is, a random part), and if it is a seasonal time series, in addition to the above two, there is a seasonal part.

Some time series models exhibit a certain cycle or periodicity, and such a time series is called a seasonal data. In statistics, it is generally believed that the demand (or sale) of a given product will exhibit seasonality as the underlying time series undergoes predictable periodic changes. Seasonality is one of the common statistical models which is used to improve the accuracy of demand forecasting.

For seasonal data, in a time series, if the similarity is exhibited after n time intervals, the sequence is said to have a periodic characteristic with a period of n.

We can decompose it into three parts:

1. trend section
2. seasonal part
3. irregular parts

A non-seasonal time series contains a trend section and an irregular section. Decomposing the time series is an attempt to split the time series into these

components, that is, the two parts of the trend and the irregularity which are needed to be estimated.

In order to estimate the trend portion of a non-seasonal time series so that it can be described by an additive model, the most common method is the smoothing method, such as calculating a simple moving average of a time series.

In our project, we can clearly know that the study of precipitation has great significance for the prediction and research of regional climate. Further research on the local geographical environment can also be carried out through the analysis of precipitation data, which will help the related research. In addition, in this project, the data about precipitation is seasonal because rainfall varies with the seasons. And in this set of data, we take a few years of data, so in the past few years, rainfall would change periodically with the seasonal changes of the year. Apart from that, this article collated the rainfall data from January 2004 to October 2018 in the Toronto central area and then builds and analyzes the model. The unit is mm.

Confirmatory Data Analysis

1. Import of the library used

The required library is mainly divided into three parts:

a) Data processing: pandas

b) Time series modeling: statsmodels

c) Visualization: matplotlib

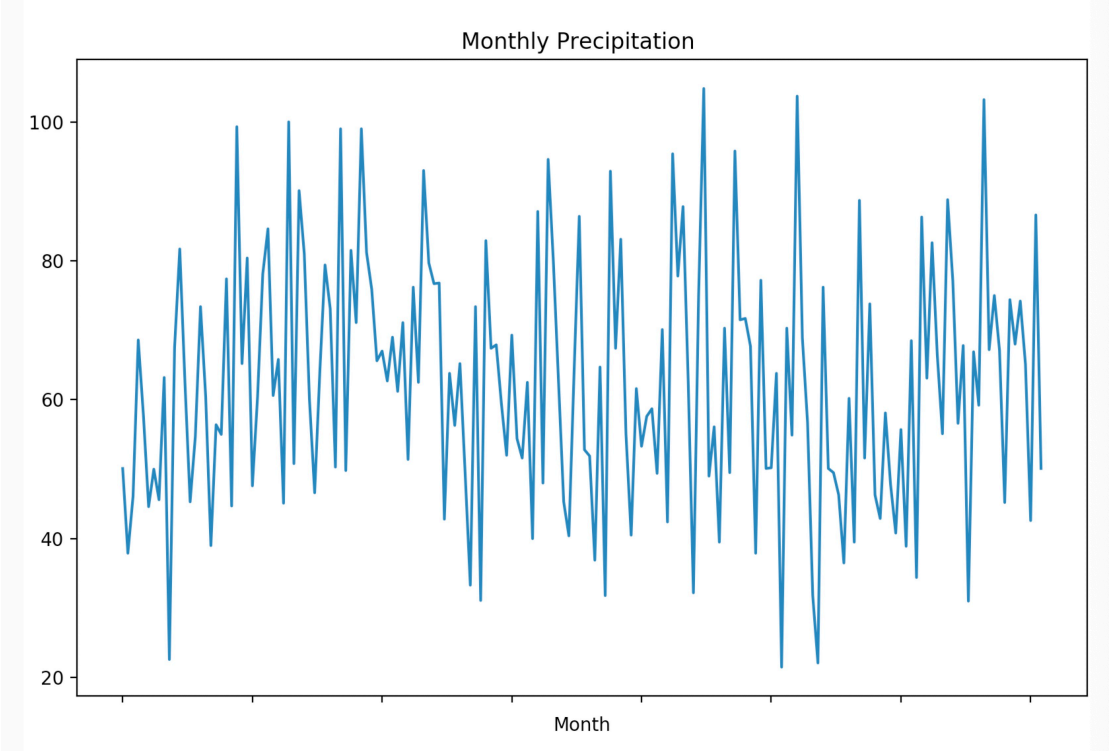
Load the required library：

```
import pandas as pd
import numpy as np
import itertools

# TSA from Statsmodels
import statsmodels.api as sm
import statsmodels.formula.api as smf
import statsmodels.tsa.api as smt
from statsmodels.graphics.api import qqplot
from statsmodels.tsa.stattools import adfuller
from statsmodels.tsa.seasonal import seasonal_decompose
#from sklearn.metrics import mean_squared_error
from statsmodels.tsa.stattools import acf, pacf
from statsmodels.tsa.arima_model import ARIMA
# from statsmodels.tsa.statespace.sarimax import SARIMAX

# Display and Plotting
import matplotlib.pylab as plt
import seaborn as sns
```

2. Use pandas to read in data and organize data

   Read in the precipitation data set in the Toronto central acrea you need and visualize the data set. The code is as follows:

```
precipitation = pd.read_csv("precipitation.csv", sep=',', index_col=0)
precipitation.plot(figsize=(12, 8))
plt.title("Monthly Precipitation ")
plt.show()
```



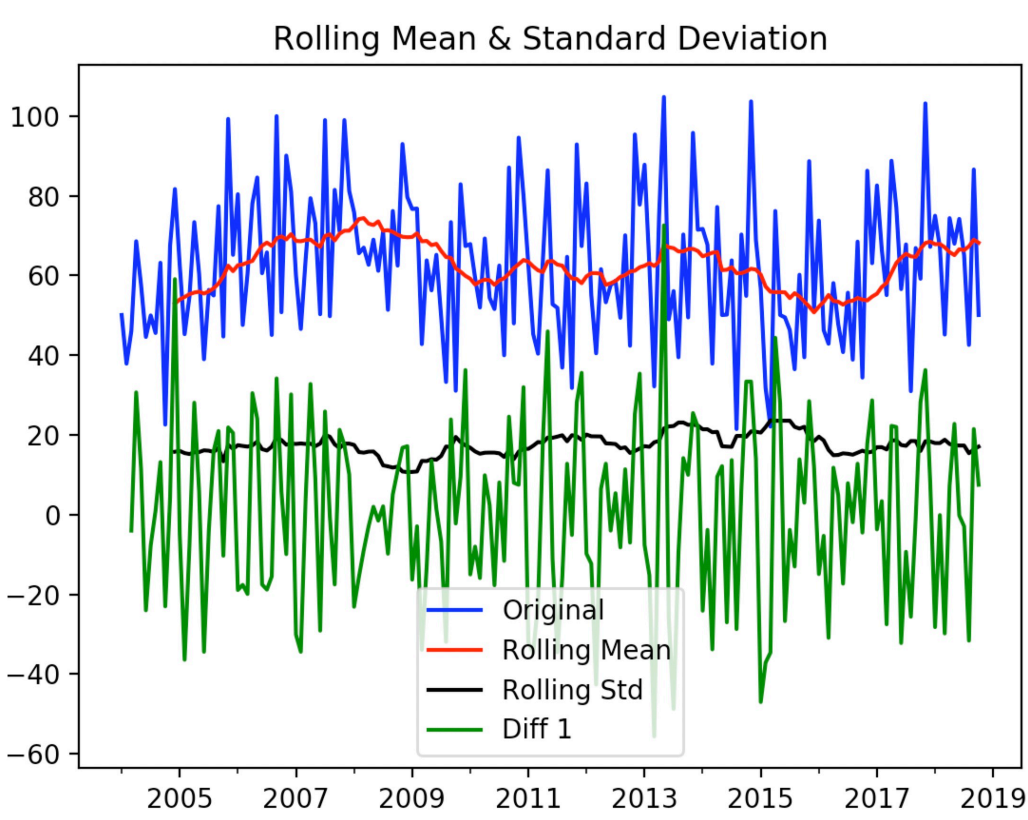
As can be seen from the above figure, the data shows a clear trend of fluctuations with a little upward trend at the beginning, and there are also seasonal signs. For the convenience of processing, we re-index the data, the code is as follows:

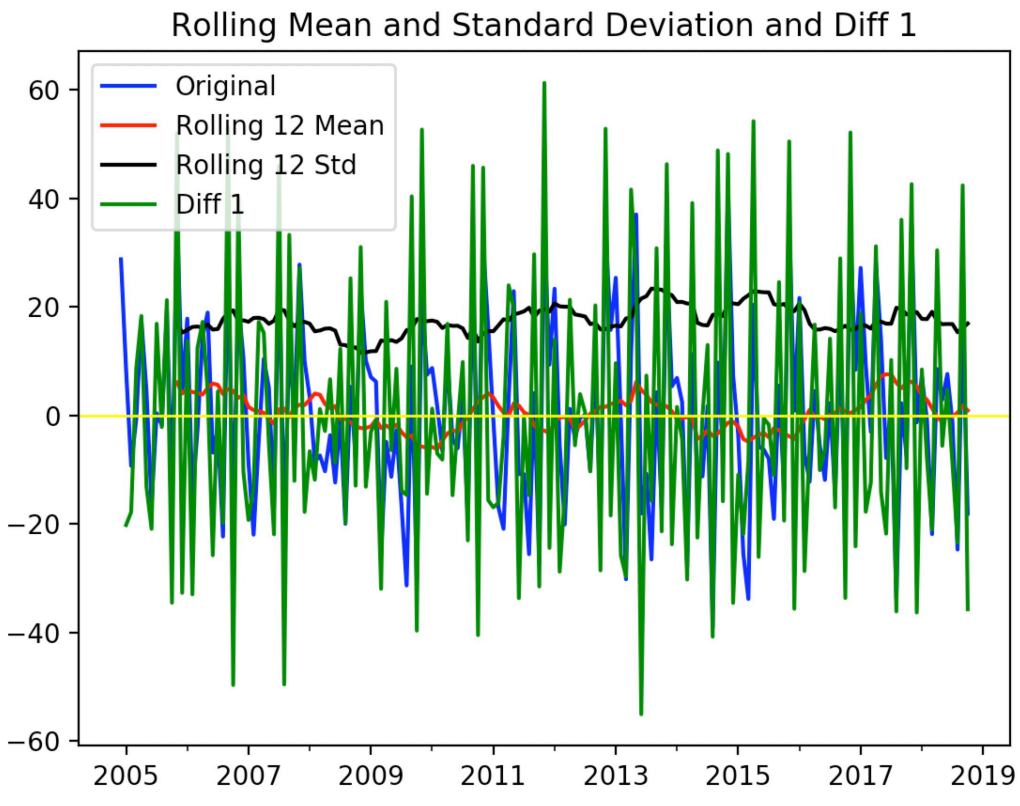
```
ts = pd.Series(np.array(precipitation['precipitation'].astype('float64')),
               index=pd.period_range('200401', '201810', freq='M'))
ts.head()
```

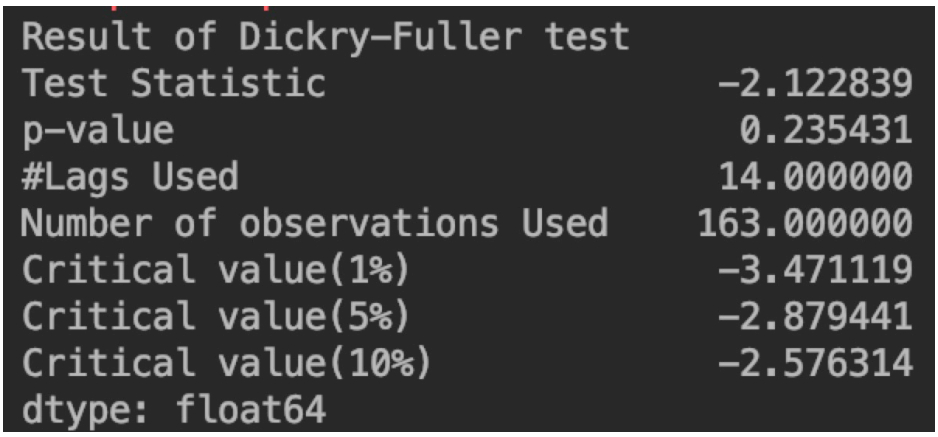
3. Stationarity test

After the data processing is completed, the data is checked for stability. In fact, the above data shows that the data is stationary, but in order to explain the integrity of the modeling process, the following is a routine test of stationarity. First, the data is scrolled to find the mean value, then the standard deviation of the data is given, and the data is subjected to first-order differential processing. The difference is made here for the convenience of the subsequent processing steps, and then use the ADF to test whether the unit root of the autoregressive model satisfies the requirements for stationarity under the AIC and BIC standards. It is given here in a visual way. The code is as follows:

```
def test_stationarity(timeseries):
    rolmean = timeseries.rolling(12).mean()

    rolstd = timeseries.rolling(12).std()
    ts_diff = timeseries - timeseries.shift()

    orig = timeseries.plot(color='blue', label='Original')
    mean = rolmean.plot(color='red', label='Rolling Mean')
    std = rolstd.plot(color='black', label='Rolling Std')
    diff = ts_diff.plot(color='green', label='Diff 1')

    plt.legend(loc='best')
    plt.title('Rolling Mean & Standard Deviation')
    plt.show(block=False)


    print('Result of Dickry-Fuller test')
    dftest = adfuller(timeseries, autolag='AIC')
    dfoutput = pd.Series(dftest[0:4], index=['Test Statistic', 'p-value', '#Lags Used', 'Number of observations Used'])
    for key, value in dftest[4].items():
        dfoutput['Critical value(%s)' % key] = value
    print(dfoutput)


test_stationarity(ts)
```


Rolling Mean & Standard Deviation
100
80
60
40
20
0
−20
−40
−60
2005
2007
2009
2011
2013
2015
2017
2019
Original
Rolling Mean
Rolling Std
Diff 1


Rolling Mean and Standard Deviation and Diff 1
60
40
20
0
−20
−40
−60
2005
2007
2009
2011
2013
2015
2017
2019
Original
Rolling 12 Mean
Rolling 12 Std
Diff 1



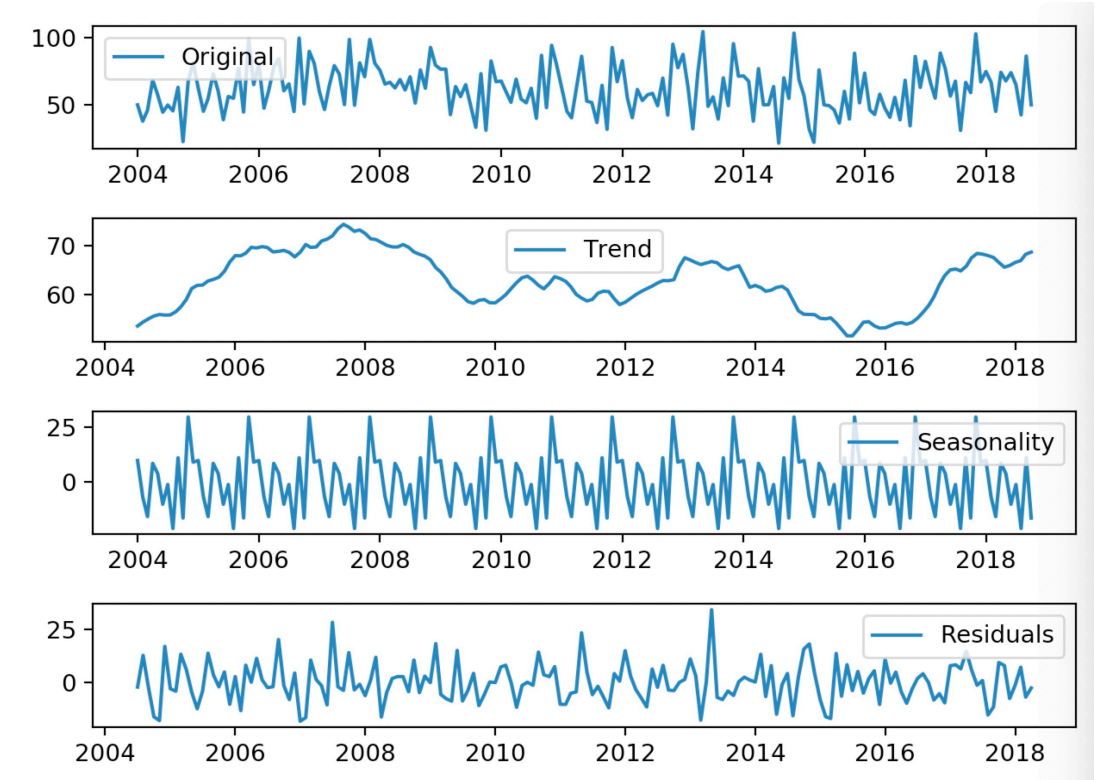
The blue line is the visualization of the original data, and the red line is the visualization of the rolling average. It can be seen that the data shows a clear fluctuation trend, but there is some clear rising trend at the beginning. The black line is the standard deviation visualization, the yellow line is the 0 position line, and the green line is the visualization after the first order difference data. It can be seen from the comparison of different colors lines that after the first-order difference, the data basically meets the requirements of stationarity.

.

4. AR, MA, ARIMA model recognition

This article mainly wants to use ARIMA related models for modeling, but as we mentioned above, we can see that the data has a seasonal trend, so we do a seasonal test, and see if the data really has a seasonal trend from the image. The code is as follows:

```
from statsmodels.tsa.seasonal import seasonal_decompose
ts.interpolate(inplace = True)
ts.index=ts.index.to_timestamp()
decomposition = seasonal_decompose(ts)

trend = decomposition.trend
seasonal = decomposition.seasonal
residual = decomposition.resid
plt.subplot(411)
plt.plot(ts, label='Original')
plt.legend(loc='best')
plt.subplot(412)
plt.plot(trend, label='Trend')
plt.legend(loc='best')
plt.subplot(413)
plt.plot(seasonal, label='Seasonality')
plt.legend(loc='best')
plt.subplot(414)
plt.plot(residual, label='Residuals')
plt.legend(loc='best')
plt.tight_layout()
plt.show()
```



In the above figure, the first picture is the image of the original data, the second image is the trend image of the data, and the obvious fluctuation trend can be seen. The third image is the seasonal trend. The conventional seasonal trend is in front of us. The last image is the part of the residual. It can be seen that the residual is also a sign of white noise. Below, we will use the AIC, BIC, HQIC standards to see what kind of

parameter input is required if using AR, MA, or ARIMA.

Akaike information criterion, referred to as AIC, is a standard for measuring the goodness of statistical model fitting. It was founded and developed by Japanese statistician Akio Hiroshi. The Akaike information criterion is based on the concept of entropy, which can weigh the complexity of the estimated model and the superiority of the model fitting data. The preferred model should be the one with the lowest AIC value.

BIC which is Bayesian Information Criterions. The log likelihood of the model is the value that is maximized by the process that computes the maximum likelihood value for the Bi parameters. The first half of the formulas of AIC and BIC are the same, and the second half of BIC is the penalty. When n≥8, kln(n) ≥ 2k, so BIC punishes the model parameters more than AIC in large data volume. This leads to BIC preferring to choose a simple model with fewer parameters.

They are all based on the likelihood function L and the number k of parameters, where n represents the sample size. Since there is a corresponding function in Python.

The formulas of them are:

$$AIC = -2\ln L + 2k$$
$$BIC = -2\ln L + \ln n \cdot k$$
$$HQIC = -2\ln L + \ln(\ln n) \cdot k$$

The code is as follows:

```
import itertools

# Define the p, d and q parameters to take any value between 0 and 2
p = d = q = range(0, 3)

# Generate all different combinations of p, q and q triplets
pdq = list(itertools.product(p, d, q))
print('p,d,q', pdq, '\n')
for param in pdq:
    try:
        model = ARIMA(ts, order=param)
        result_ARIMA = model.fit(disp=-1)
        print('ARIMA{}-- AIC:{} -- BIC:{} --HQIC:{}'.format(
            param, result_ARIMA.aic, result_ARIMA.bic, result_ARIMA.hqic))
    except:
        continue
```

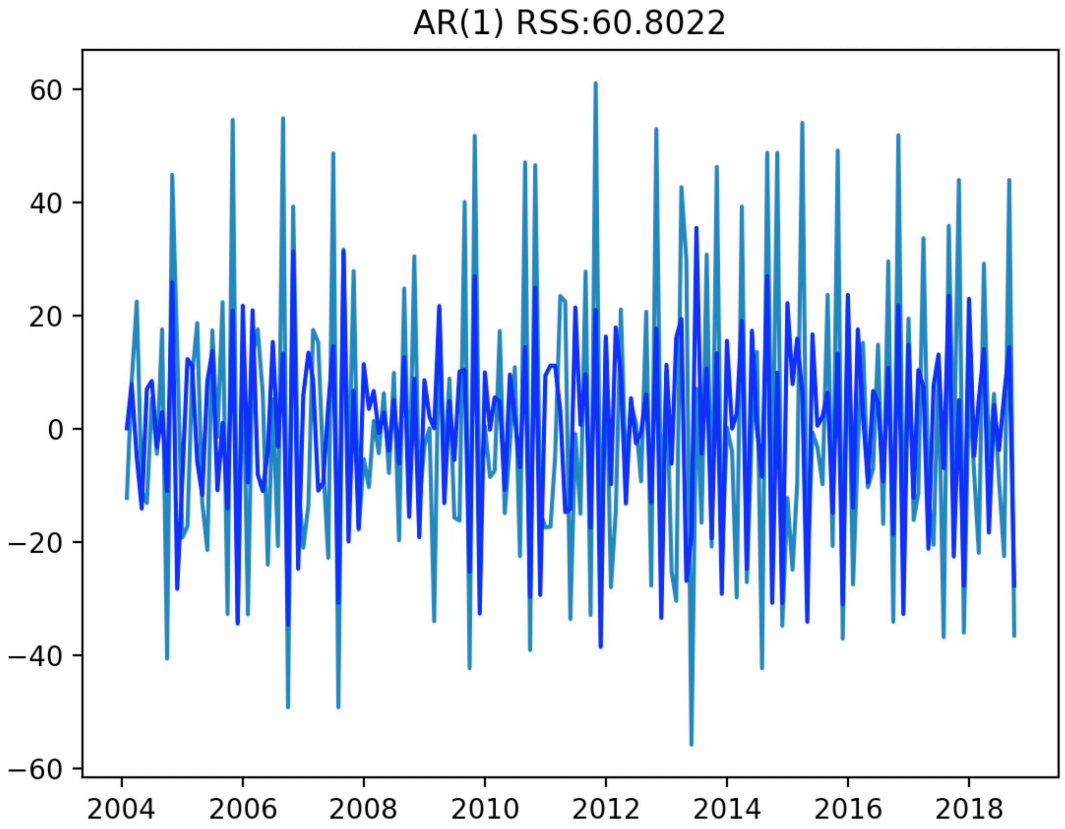
```
ARIMA(0, 0, 0)-- AIC:1659.7697364548626 -- BIC:1666.1220359200101 --HQIC:1662.345982394946
ARIMA(0, 0, 1)-- AIC:1532.5883314197572 -- BIC:1542.1167806174788 --HQIC:1536.4527003298824
ARIMA(0, 0, 2)-- AIC:1532.3364821381429 -- BIC:1545.0410810684382 --HQIC:1537.4889740183098
ARIMA(0, 1, 0)-- AIC:1858.63464860544 -- BIC:1864.9756165955162 --HQIC:1861.2065138062405
ARIMA(0, 1, 1)-- AIC:1658.5732772442384 -- BIC:1668.084729229353 --HQIC:1662.4310750454395
ARIMA(0, 2, 0)-- AIC:2070.6421228947183 -- BIC:2076.9716948425653 --HQIC:2073.209577550495
ARIMA(0, 2, 1)-- AIC:1856.2461753394819 -- BIC:1865.7405332612525 --HQIC:1860.0973573231465
ARIMA(1, 0, 0)-- AIC:1571.6200715011933 -- BIC:1581.1485206989148 --HQIC:1575.4844404113185
ARIMA(1, 0, 1)-- AIC:1531.600726068455 -- BIC:1544.3053249987504 --HQIC:1536.753217948622
ARIMA(1, 0, 2)-- AIC:1537.4145034950816 -- BIC:1553.2952521579507 --HQIC:1543.8551183452903
ARIMA(1, 1, 0)-- AIC:1689.969208462542 -- BIC:1699.4806604476564 --HQIC:1693.827006263743
ARIMA(1, 1, 1)-- AIC:1571.900681551906 -- BIC:1584.5826175320585 --HQIC:1577.0444119535073
ARIMA(1, 2, 0)-- AIC:1848.8947008915382 -- BIC:1858.3890588133088 --HQIC:1852.7458828752028
ARIMA(2, 0, 0)-- AIC:1570.0446727581088 -- BIC:1582.7492716884042 --HQIC:1575.1971646382758
ARIMA(2, 0, 1)-- AIC:1528.8922961338908 -- BIC:1544.77304479676 --HQIC:1535.3329109840995
ARIMA(2, 0, 2)-- AIC:1521.0141493328133 -- BIC:1540.0710477282562 --HQIC:1528.7428871530637
ARIMA(2, 1, 0)-- AIC:1659.8303420004818 -- BIC:1672.5122779806343 --HQIC:1664.974072402083
ARIMA(2, 1, 1)-- AIC:1570.5919617655136 -- BIC:1586.4443817407043 --HQIC:1577.0216247675153
ARIMA(2, 2, 0)-- AIC:1771.4552356981237 -- BIC:1784.1143795938178 --HQIC:1776.5901450096767
ARIMA(2, 2, 1)-- AIC:1660.460481451369 -- BIC:1676.2844113209867 --HQIC:1666.8791180908102
```

From the above AIC, BIC, HQIC output results, we can generally try several models like AR (1), MA (1), ARIMA (1, 0, 1) and so on. When we decide to use the AR or MA model, we can use ACF and PACF to judge the value of the parameter.

5. Model fitting

Firstly try the model AR (1) with the following code:

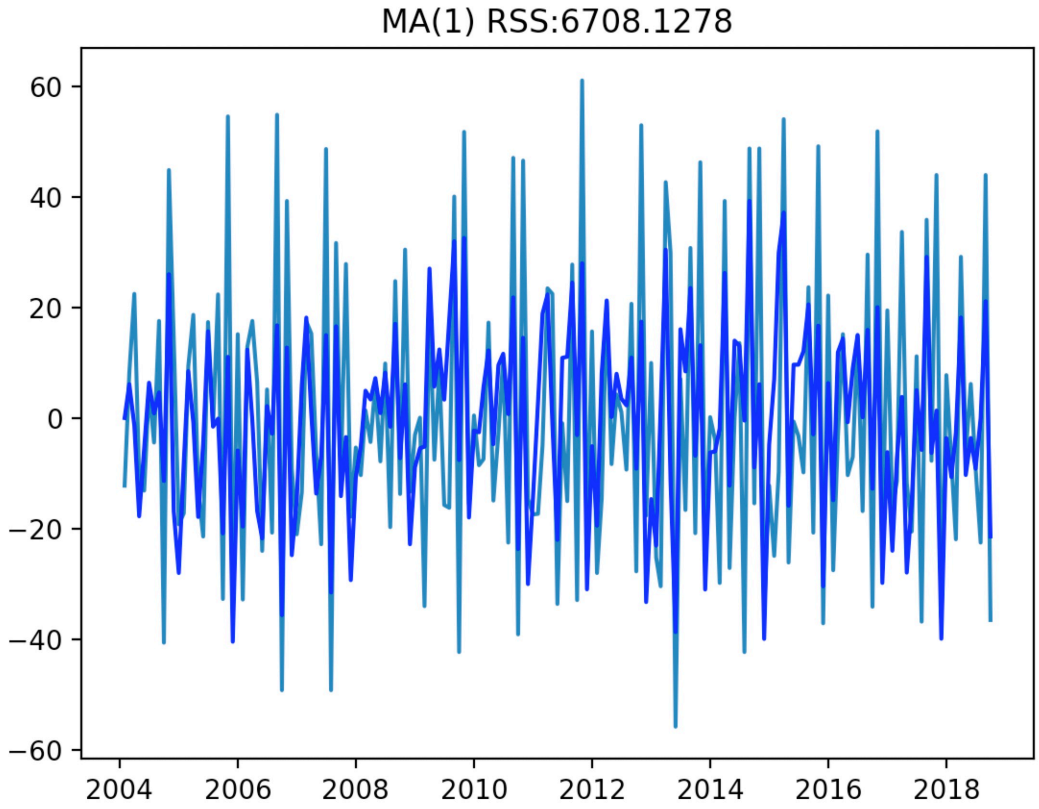
```
#AR model
model=ARIMA(ts_diff,order=(1,0,0))
result_AR=model.fit(disp=-1)
plt.plot(ts_diff)
plt.plot(result_AR.fittedvalues,color='blue')
plt.title('AR(1) RSS:%.4f'%sum(result_AR.fittedvalues-ts_diff)**2)
plt.show()
```



As can be seen from the output of the AR (1) model above, the RSS value is 60.8022.

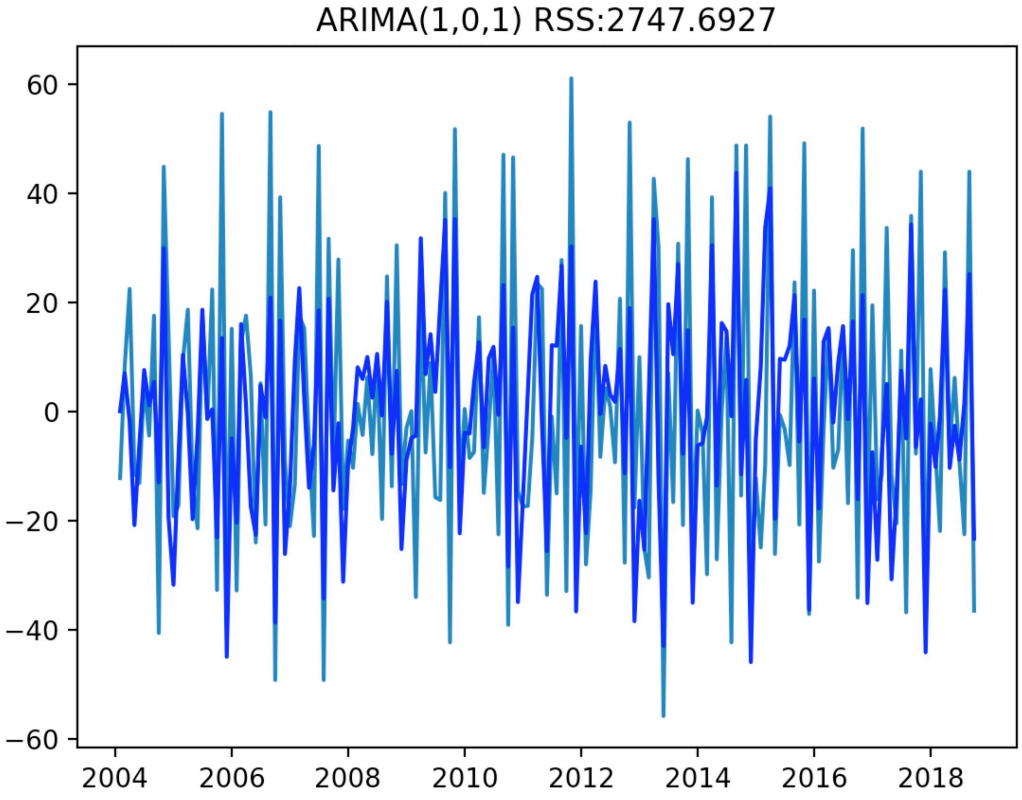
And then try the MA (1) model below, the code is as follows:

```
#MA model
model=ARIMA(ts_diff,order=(0,0,1))
result_MA=model.fit(disp=-1)
plt.plot(ts_diff)
plt.plot(result_MA.fittedvalues,color='blue')
plt.title('MA(1) RSS:%.4f'%sum(result_MA.fittedvalues-ts_diff)**2)
plt.show()
```



As can be seen from the output of the MA (1) model above, the RSS value is 6708.1278, which is larger significantly than that of AR (1). Finally, we will try ARIMA (1,0,1) to see the fitting effect of the data, the code is as follows:

```
model=ARIMA(ts_diff,order=(1,0,1))
result_ARIMA_diff=model.fit(disp=-1)
plt.plot(ts_diff)
plt.plot(result_ARIMA_diff.fittedvalues,color='blue')
plt.title('ARIMA(1,0,1) RSS:%.4f'%sum(result_ARIMA_diff.fittedvalues-ts_diff)**2)
plt.show()
```

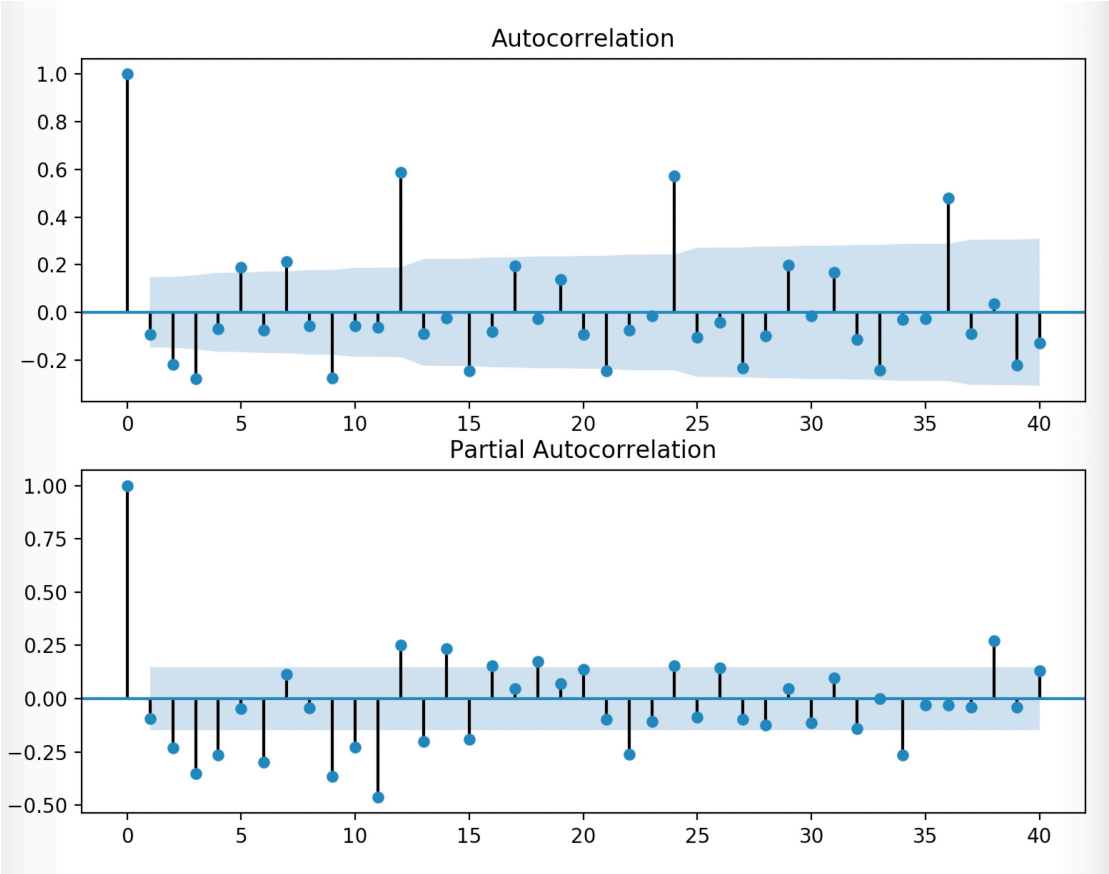


Comparing output results RSS from AR(1), MA (1) model, ARIMA (1,0,1), RSS value of AR (1) is the smallest, and ARIMA (1, 0, 1) is also much smaller than MA (1).

6. Model diagnosis

Let's take a look at the diagnostic results of the model AR (1), the code is as follows:

```
fig = plt.figure(figsize=(12,8))
ax1 = fig.add_subplot(211)
fig = sm.graphics.tsa.plot_acf(result_ARIMA_diff.resid.values.squeeze(), lags=40, ax=ax1)
ax2 = fig.add_subplot(212)
fig = sm.graphics.tsa.plot_pacf(result_ARIMA_diff.resid, lags=40, ax=ax2)
plt.show()
```



From the above ACF and PACF, there is no correlation between the residuals at the lower order, and there are still some problems at the high order.

Let's take a look at the D_W test. The code is as follows:

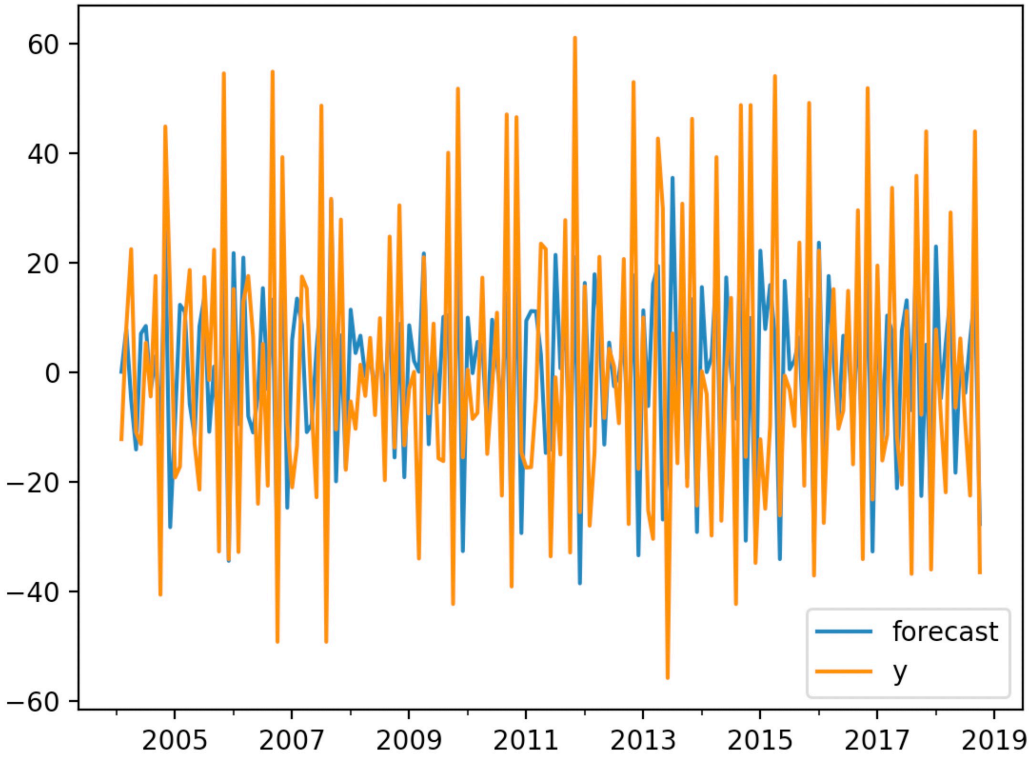
```
import statsmodels.api as sm
print(sm.stats.durbin_watson(result_ARIMA_diff.resid.values))
```

The result is: 2.171238395453154.

7. Prediction

After the diagnosis of the model is completed, because the diagnosis results are still satisfactory, we will predict the data below. Here, the output prediction results are compared with the real results. The code is as follows:

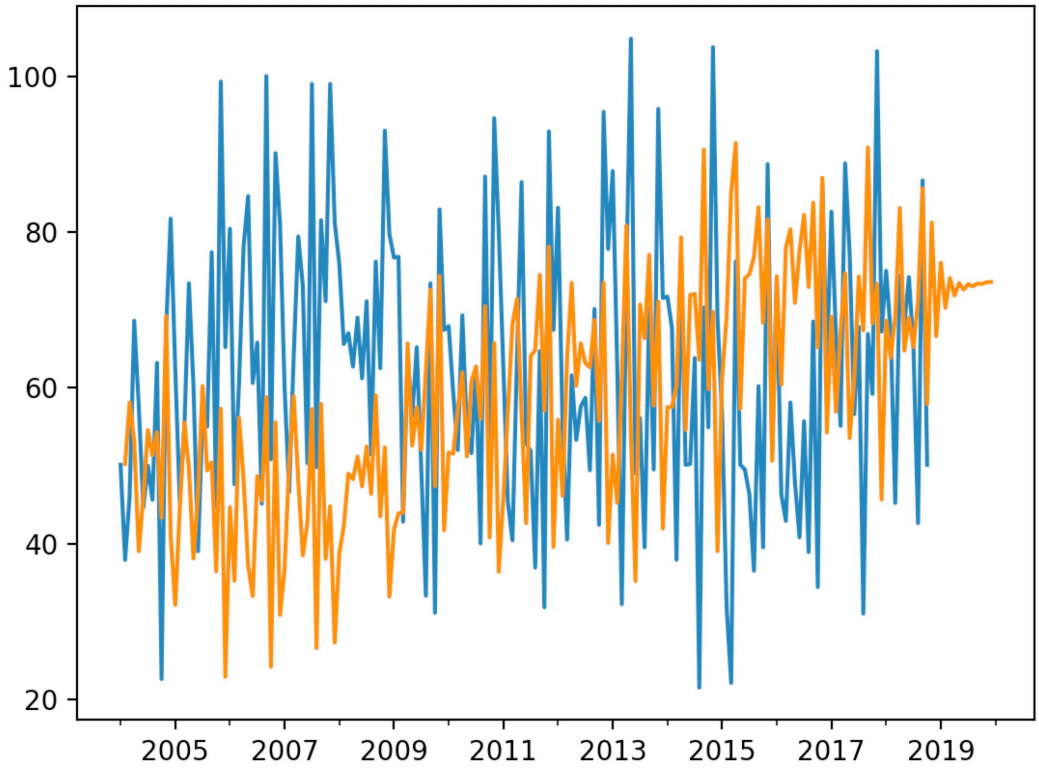
```
model_results=ARIMA(ts_diff,order=(1,0,0))
result_ARIMA=model_results.fit(disp=-1)
result_ARIMA.plot_predict()
result_ARIMA.forecast()
plt.show()
```



The figure is a visualization of the data and the prediction of the data, where the orange image is a visualization of the first-order difference of the raw data and the blue image is a visualization of the results predicted by the orange data. As can be seen from the figure, the prediction results basically capture the characteristics of the data.

The following code is used to output our predictions for the raw data:

```
predict_dta =result_ARIMA_diff.predict('2018-11', '2019-12', dynamic=True)
AA = predictions_ARIMA_diff.append(predict_dta)
predictions_ARIMA = pd.Series(ts.ix[0], index=pd.period_range('200402','201912',freq='M'))
AA.index =pd.period_range('200402','201912',freq='M')
predictions_ARIMA2 = predictions_ARIMA.add(AA.cumsum(),fill_value=0)
```



Here is the forecast of rainfall from November 2018 to December 2019, with visual results. As can be seen, it is foreseeable that rainfall will increase in the coming period.

Conclusions

This paper mainly introduces the establishment of the time series ARIMA correlation model and the diagnosis and prediction process. The article first uses pandas to read in the data and organize the data; then use the test_stationarity function to perform the stationarity test of the original data and the first-order differential data to meet the requirements of the stability of the latter ARIMA model; then the data is differentiated to make the data meet the smoothness. After that, using the grid search method to determine the optimal parameters, and then use the relevant time series function in the statsmodels library to identify the time series model AR, MA, ARIMA, and fit the model. For ARIMA (1,0,1), the model performs the diagnosis of the model, mainly including the test of autocorrelation and independence; based on this, we then use the model to predict the first-

order differential data. Finally, we predict the rainfall from November 2018 to December 2019, and thus complete the whole process of the entire time series model.

The study of rainfall has an important impact on future weather research. In addition, it is also helpful for natural disasters such as floods and mudslides.

Appendices

Here is the data set:

month	production
4-Jan	50.1
4-Feb	37.9
4-Mar	46.1
4-Apr	68.6
4-May	57.7
4-Jun	44.6
4-Jul	50
4-Aug	45.6
4-Sep	63.2
4-Oct	22.6
4-Nov	67.5
4-Dec	81.7
5-Jan	62.5
5-Feb	45.3
5-Mar	54.7
5-Apr	73.4
5-May	60.4
5-Jun	39
5-Jul	56.4
5-Aug	55
5-Sep	77.4
5-Oct	44.7
5-Nov	99.3
5-Dec	65.2
6-Jan	80.4
6-Feb	47.6
6-Mar	60.5
6-Apr	78.1
6-May	84.6
6-Jun	60.6
6-Jul	65.8
6-Aug	45.1
6-Sep	100
6-Oct	50.8
6-Nov	90.1
6-Dec	81
7-Jan	60
7-Feb	46.6
7-Mar	64.1

7-Apr	79.4
7-May	73.1
7-Jun	50.3
7-Jul	99
7-Aug	49.8
7-Sep	81.5
7-Oct	71.1
7-Nov	99
7-Dec	81.2
8-Jan	75.9
8-Feb	65.6
8-Mar	67
8-Apr	62.7
8-May	69
8-Jun	61.2
8-Jul	71.1
8-Aug	51.4
8-Sep	76.2
8-Oct	62.5
8-Nov	93
8-Dec	79.7
9-Jan	76.7
9-Feb	76.8
9-Mar	42.8
9-Apr	63.8
9-May	56.3
9-Jun	65.2
9-Jul	49.5
9-Aug	33.3
9-Sep	73.4
9-Oct	31.1
9-Nov	82.9
9-Dec	67.4
10-Jan	67.9
10-Feb	59.4
10-Mar	52
10-Apr	69.3
10-May	54.4
10-Jun	51.6
10-Jul	62.5
10-Aug	40
10-Sep	87.1
10-Oct	48
10-Nov	94.6
10-Dec	80

11-Jan	62.6
11-Feb	45.3
11-Mar	40.4
11-Apr	63.9
11-May	86.4
11-Jun	52.8
11-Jul	51.9
11-Aug	36.9
11-Sep	64.7
11-Oct	31.8
11-Nov	92.9
11-Dec	67.4
12-Jan	83.1
12-Feb	55.1
12-Mar	40.5
12-Apr	61.6
12-May	53.3
12-Jun	57.6
12-Jul	58.7
12-Aug	49.4
12-Sep	70.1
12-Oct	42.4
12-Nov	95.4
12-Dec	77.8
13-Jan	87.8
13-Feb	62.6
13-Mar	32.2
13-Apr	74.9
13-May	104.8
13-Jun	49
13-Jul	56.1
13-Aug	39.5
13-Sep	70.3
13-Oct	49.5
13-Nov	95.8
13-Dec	71.5
14-Jan	71.7
14-Feb	67.7
14-Mar	37.9
14-Apr	77.2
14-May	50.1
14-Jun	50.2
14-Jul	63.8
14-Aug	21.5
14-Sep	70.3

14-Oct	54.9
14-Nov	103.7
14-Dec	68.9
15-Jan	56.7
15-Feb	31.8
15-Mar	22.1
15-Apr	76.2
15-May	50.1
15-Jun	49.5
15-Jul	46.3
15-Aug	36.5
15-Sep	60.2
15-Oct	39.5
15-Nov	88.7
15-Dec	51.6
16-Jan	73.8
16-Feb	46.3
16-Mar	42.9
16-Apr	58.1
16-May	47.8
16-Jun	40.8
16-Jul	55.7
16-Aug	38.9
16-Sep	68.5
16-Oct	34.4
16-Nov	86.3
16-Dec	63.1
17-Jan	82.6
17-Feb	66.5
17-Mar	55.1
17-Apr	88.8
17-May	77.1
17-Jun	56.6
17-Jul	67.8
17-Aug	31
17-Sep	66.9
17-Oct	59.2
17-Nov	103.2
17-Dec	67.2
18-Jan	75
18-Feb	67.1
18-Mar	45.2
18-Apr	74.4
18-May	68
18-Jun	74.2

18-Jul	65.1
18-Aug	42.6
18-Sep	86.6
18-Oct	50.1

References

ALI F, MANSOOR A B. Computer vision based automatic scoring of shooting targets[C] // Multi-topic Conference, 2008(IN-MIC 2008), IEEE International. IEEE, 2008:515-519.

HUANG G B,ZHOU H M,DING X J, et al. Extreme Learning Machine for Regression and Multiclass Classification[J]. IEEE Transactions on Systems, Man, and Cybernetics, 2012, 42(2): 513G529.

LEEWC, CHEN C H. A fast template matching method for rotation invariance using two-stage process [C] // Proceeding of IEEE Conference on Intelligent Information Hiding and Multi-media Signal Processing. Kyoto, Japan, 2009:9-12.

NIU C, LIX H, YIS H, et al. Forecasting Model of Geomagnetic Variation Field Based on Modified Ensemble Empirical Mode Decomposition-Sample Entropy-east Square support Vector Machine[J]. Geomatics and Information Science of Wuhan University, 2014, 39(5): 626-630.

PAN W C.A New Fruit Fly Optimization Algorithm:Taking the Financial Distress Model as An Example[J]. Knowledge Based Systems, 2012, 26(2): 69-74.

XIN N F, QIAN Q C,PENG H D,et al. Design of automatic target-scoring system of shooting game based on computer vision [C] // Proceedings of the IEEE International Conference on Automation and Logistics. Shenyang, China, 2009.

XINGZX, GUO H,FU Q. Analysis of Influencing Factors of Rainfall in Irrigation Area and Combining Rainfall Forecasting [J]. Transactions of the Chinese Society for Agricultural Machinery, 2015, 46(8): 97-103.

ZHAO L L, GENG G H, LI K, et al. Images matching algorithm based on SURF and fast approximate nearest neighbor search [J]. Application Research of Computers, 2013, 30(3): 921-923.