MAT-5491-2 Topics in Complex System:

Time Series Final Project

Liu Xingyu

April 2020

Yeshiva University

**Content**

**1 Introduction and motivation** …………………………………………………….2

**2 Data source identification and description** ……………………………...2

**3 Methodology** ……………………………………………………………………………..3

**4 Data preprocessing** …………………………………………………………………..3

**5 Initial Hypothesis** ……………………………………………………………………...4

**6** **Descriptive analysis** ………………………………………………………………….4

6.1 Time series plot ……………………………………………………………………….5

6.2 Autocorrelation checking …………………………………………….7

6.3 Partial Autocorrelation checking ………………………………………………9

6.4 Scatter plot ……………………………………………………………………………..12

6.5 Normality checking …………………………………………………………………14

6.6 Exponential checking ………………………………………………………………16

**7 Modelling: ARIMA model** ………………………………………………………….20

7.1 Existence of stationarity on the whole data**………………………….. …**20

7.2 Fitting ARIMA(p,q) models with arima()………………………………………..22

7.3 Forecasting the ARIMA model……………………………………...39

**8 Modelling: ARCH model and effect** …………………………………………...34

8.1 McLeod-Li Test Statistics for partial data…………….....…...………..……34

8.2 Order Specification …………………………………………………………………..35

8.3 Model Fitting, Estimation and Diagnostic Checking…………………….36

8.3.1 GARCH (2,1) ………………………………………………………………………….36

8.3.2 ARCH (1) ………………………………………………………………………………37

**9** **Final Best Model Selection for ARMA + ARCH** …………………………..39

9.1 ARCH (1) + ARIMA(2,1,2) ………………………………………………………….39

**10 Forecasting** …………………………………………………………………………….. 43

**11 Conclusion** ……………………………………………………………………………… 45

**1 Introduction and Motivation**

Typhoon Gordon:

Originated location: Northern Mariana islands

Begin: 1989.7.9

Extreme: 1989.7.15

Made landfall in south China: 1989.7.18

End: 1989.7.19

Dissipated location: south China

The data I analyzed in this paper is the hourly sea surface height in Shanwei from the 23rd of July 1984 to the 24th of August 1989.

The objective of this paper is to analyze the hourly sea surface height before the happening of Typhoon Gordon, in the happening of Typhoon Gordon and after the happening of Typhoon Gordon by using the time series analysis methods and then choose the best model among different models, giving forecasts about the storm, checking whether the forecast is approximate to the real data set.

**2 Data source identification and description**

I became Marian Gidea's research assistant in December 2019, the research is mainly about using sea levels to judge whether there is an advant about the hurricanes and typhoons. Professor Christian and Dana in Yeshiva University gave the data here: the data is the hourly sea surface height in different geslas, which sit in different cities that are relevant to the hurricanes and typhoons. In this project, I picked the event of Typhoon Gordon. Typhoon Gordon was originated on July 9th, 1989 in Northern Mariana islands, got its extremity on July 15th, 1989. After striking the northern Philippines, Gordon moved through the South China Sea and slowly weakened. On July 18th, the storm made landfall in southern China and ended on July 19th. There were 6 sets of data in different cities in south China: Haikou, Hongkong, Kaohsiung, Shanwei, Xiamen, Zhapo, the data is from July 23rd, 1984 to August 24th, 1989, which perfectly covers the whole storm. Also, the data are all with suffix uhslc, which means the data are all came from University of Hawaii Sea Level Center. In this paper, I choose the city Shanwei to study.

**3 Methodology**

First, I will check whether the data is seasonal or nonseasonal, then, I considered three models, ARMA, GARCH and ARMA plus GARCH. Then, I will choose for the best model.

**4 Data preprocessing**

In this report, I used the following **R** packages:

**library(tseries)**

**library(forecast)**

**library(fGarch)**

**library(FitAR)**

**library(tswge)**

**library(rugarch)**

First, I read the sea level data into **R**:

**shanwei <- read\_excel(“~Desktop/2020 research - Typhoon gordon/gesla\_shanwei-641a-china-uhslc\_19840723-19890823.xls")**

**shanweisea = shanwei $ssh**

**5 Initial Hypothesis**

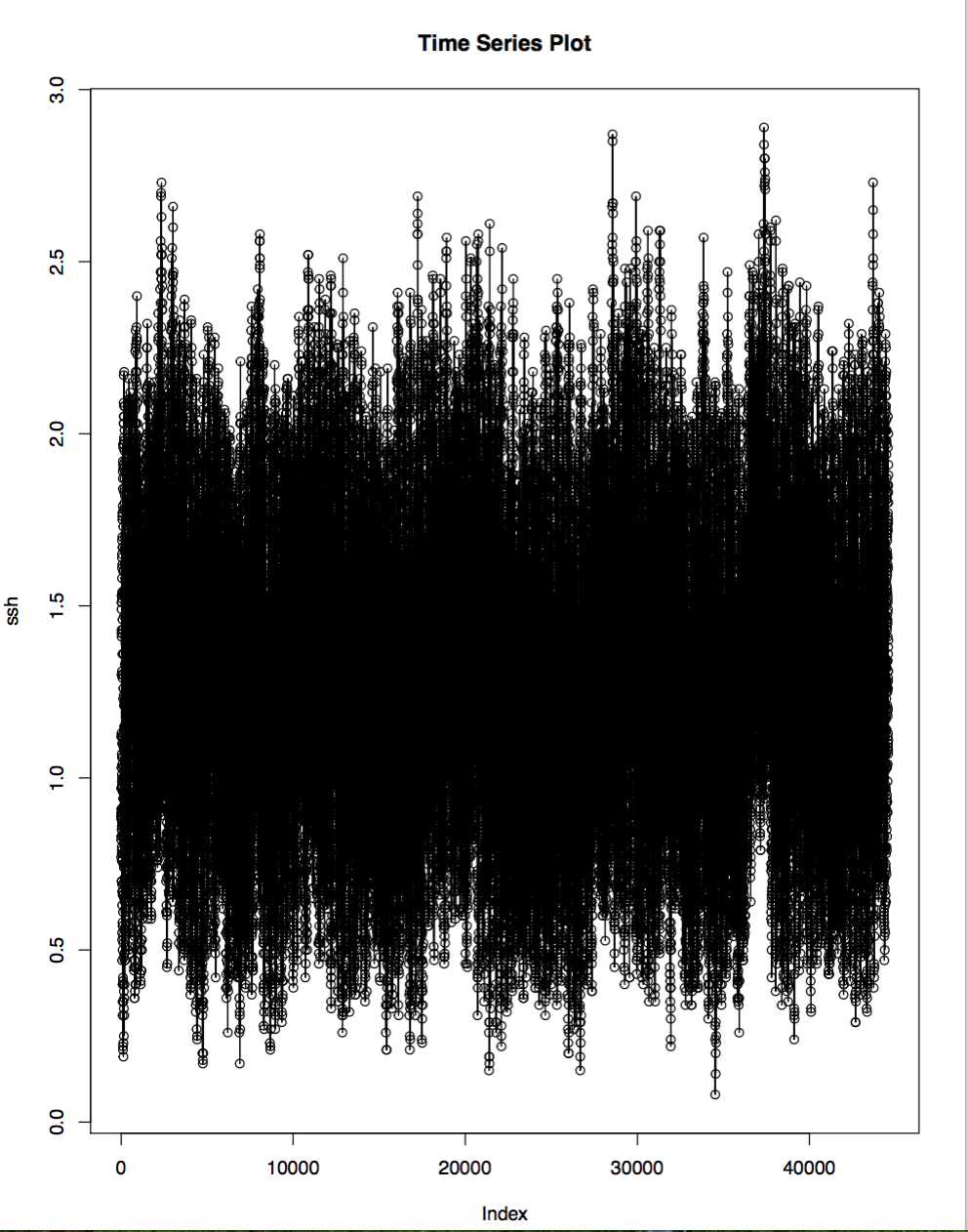
I and Marian expected to see some unusual phenomenon near and before the time of Typhoon. Suppose there should have been somethings seem like "outliers" in the model I fitted before the happening of the Typhoon, but then it means the model failed. So I assumed there should be some abnormal data pattern in the happening of Typhoon, so we could judge the regularity of those abnormal data patterns, make some models, making forecasts from this way.

Since the tide exists, I expect to see the data shows some seasonality or periodicity, In the following I will go to check it and make forward to the next step.

**6 Descriptive analysis**

**6.1 Time series plot**

**plot(shanweisea, type = 'o', ylab= 'ssh', main='Time Series Plot')**

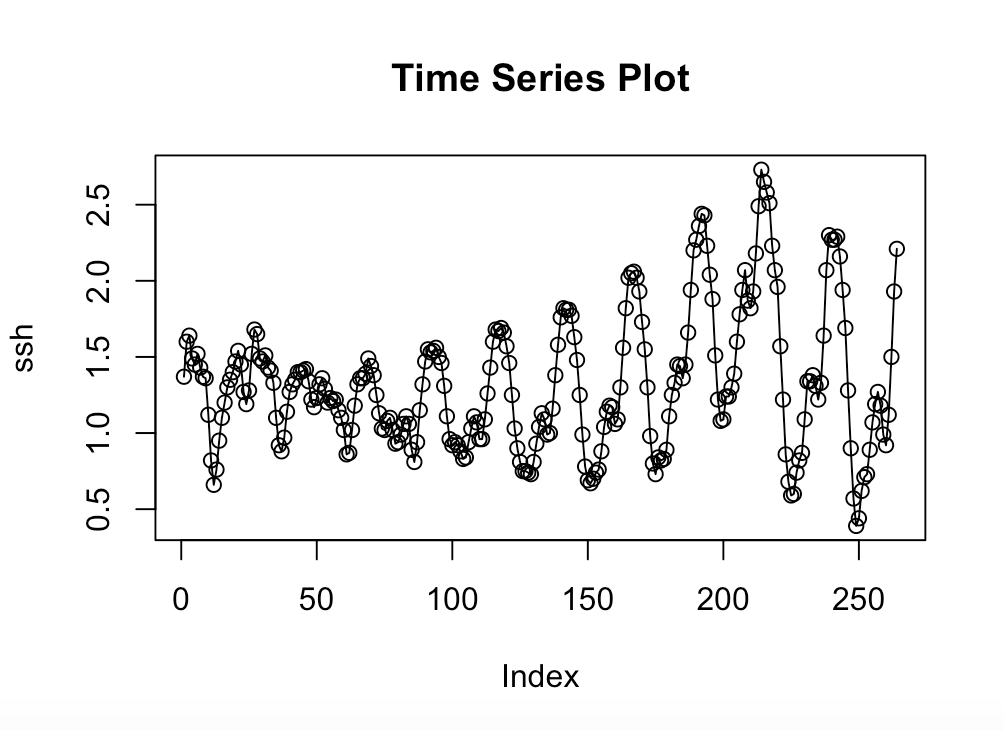
****

Also I picked a piece of data which covers the happening of Typhoon and plot its times series plot either:

**start\_data=43489**

**end\_data=43752**

**plot(shanweisea [start\_data:end\_data], type = 'o', ylab= 'ssh', main='Time Series Plot')**

****

**The interesting thing here is, the trough which is around 200 is apparently higher than other troughs, this trough is around 1989.7.17 just before typhoon Gordon made the landfall in south China.**

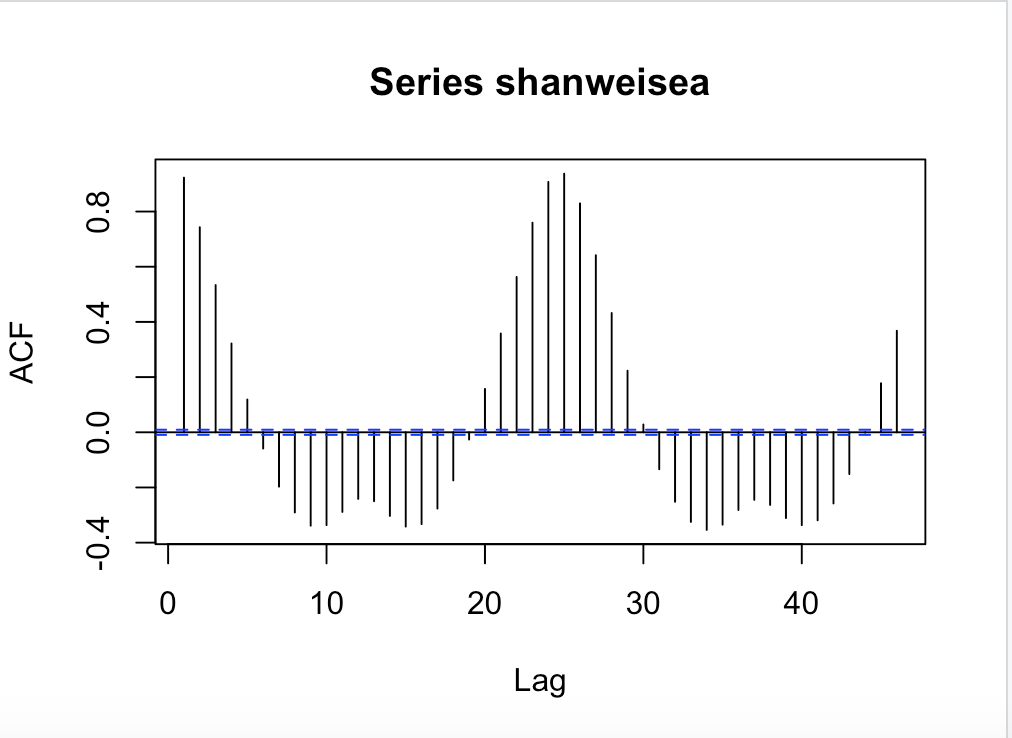
**6.2 Autocorrelation Function**

**ACF** is a complete auto-correlation function which gives us values of auto-correlation of any series with its lagged values. It describes how well the present value of the series is related with its past values. A time series can have components like trend, seasonality, cyclic and residual. ACF considers all these components while finding correlations.

**acf(shanweisea, lag.max = NULL,**

**type = c("correlation", "covariance", "partial"),**

**plot = TRUE, na.action = na.fail, demean = TRUE)**



**Try the ACF on partial data which perfectly covers the event of Typhoon Gordon :**

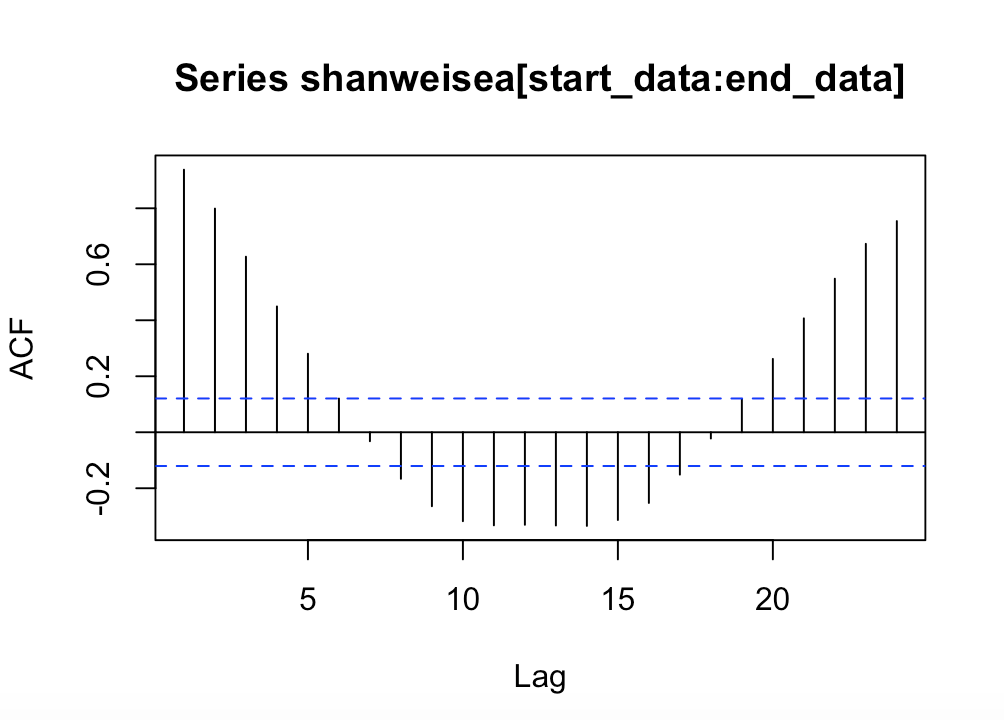
**start\_data=43489**

**end\_data=43752**

**acf(shanweisea[start\_data:end\_data], lag.max = NULL,**

**type = c("correlation", "covariance", "partial"),**

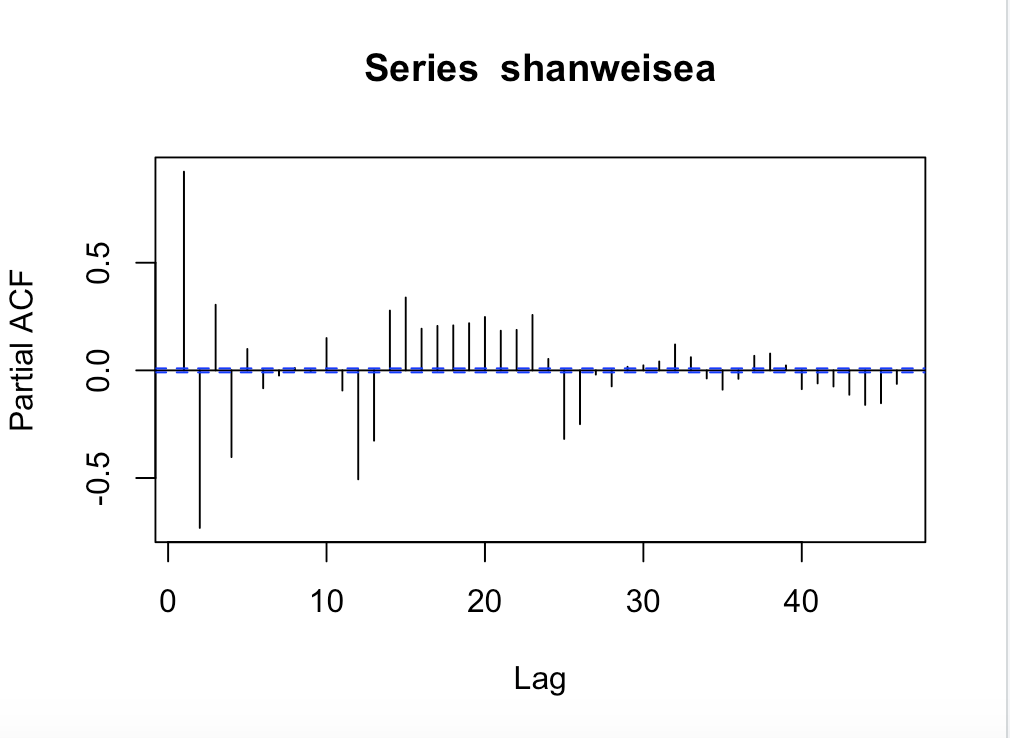
**plot = TRUE, na.action = na.fail, demean = TRUE)**

****

**6.3 Partial Autocorrelation checking**

**PACF** is a partial auto-correlation function. It finds correlation of the residuals with the next lag value. So if there is any hidden information in the residual which can be modeled by the next lag, we might get a good correlation and we will keep that next lag as a feature while modeling.

**pacf(shanweisea, lag.max = NULL, plot = TRUE, na.action = na.fail)**

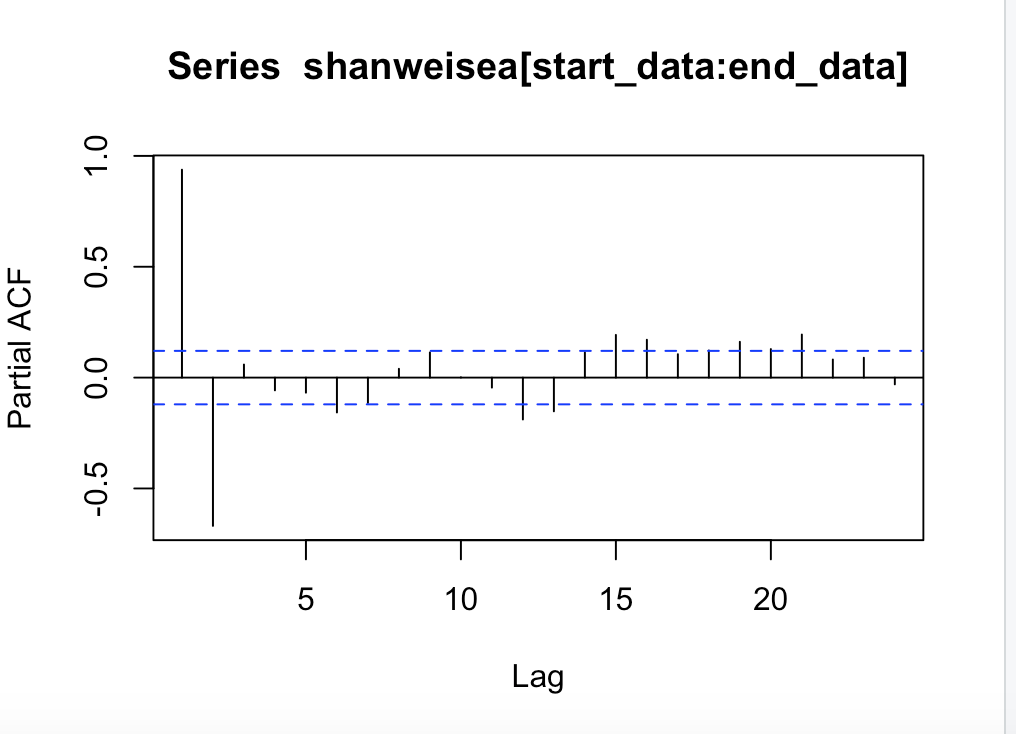


**Try the PACF on partial data which perfectly covers the event of Typhoon Gordon :**

**start\_data=43489**

**end\_data=43752**

**pacf(shanweisea[start\_data:end\_data], lag.max = NULL, plot = TRUE, na.action = na.fail)**



**Defintion**: A **seasonal** pattern exists when a series is influenced by **seasonal** factors . A **cyclic** pattern exists when data exhibit rises and falls that are not **of** fixed period.

Based on plot above, here are four main characteristics could be noticed based on the whole data:

1. Trend: No, there is no obvious trend, it states the stationary.

2. Repeat pattern: There is repeated pattern and seasonality component.

3. Changing Variance: No changing variance.

4. Behavior: The above gives a plot of the ACF and PACF. The data have autocorrelations in some way.

Also, Based on plot above, here are four main characteristics could be noticed based on the partial data:

1. Trend: Yes, there is a positive trend

2. Repeat pattern: There is repeated pattern and seasonality component.

3. Changing Variance: Yes, there is a changing variance.

4. Behavior: The above gives a plot of the ACF and PACF. The data have autocorrelations in some way.

**From the table below:**

|  |  |  |
| --- | --- | --- |
|  | **ACF** | **PACF** |
| **AR** | Geometric | p significant lags(order) |
| **MA** | q significant lags(order) | Geometric |
| **ARMA** | Geometric | Geometric |

**Temporarily conclusion till now**:

In the whole data in Shanwei, the ACF displays geometric, while the PACF also displays geometric.

In the data I selected from the beginning date to end date in Shanwei, the ACF displays geometric, while the PACF displays **2** significant lags.

According to the table, the whole data obeys the ARMA pattern, while the data I selected which covers the Typhoon is uncertain since it is non-stationary,

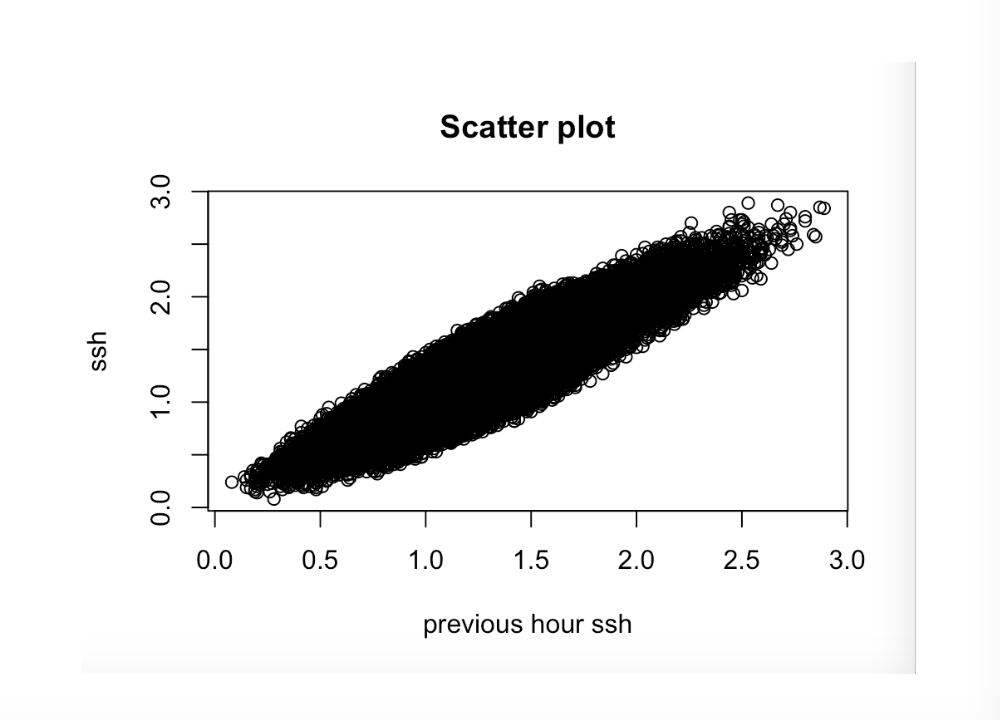
**6.4 Scatter Plot**

In order to know whether or not consecutive hours are related in some way, we can use one hour's sea surface height to predict the next hour. The scatter plot generates below investigates the correlation between the pairs of consecutive hour of sea surface height.

**plot(y = shanweisea, x = zlag (shanweisea),**

**ylab ='ssh',**

**xlab = 'previous month ssh', main = “Scatter plot")**

****

**Lemma:** When the y variable tends to increase as the x variable increases, we say there is a **positive correlation** between the variables. When the y variable tends to decrease as the x variable increases, we say there is a **negative correlation** between the variables. When there is no clear relationship between the two variables, we say there is **no correlation** between the two variables.

The scatter plot indicates there might be a positive weak linear correlation between the sea surface height with its previous hour’s sea surface height.

Since the linear relationship is positive weak, the correlation coefficient r is closer to 0.

Then I check the partial data as usual:

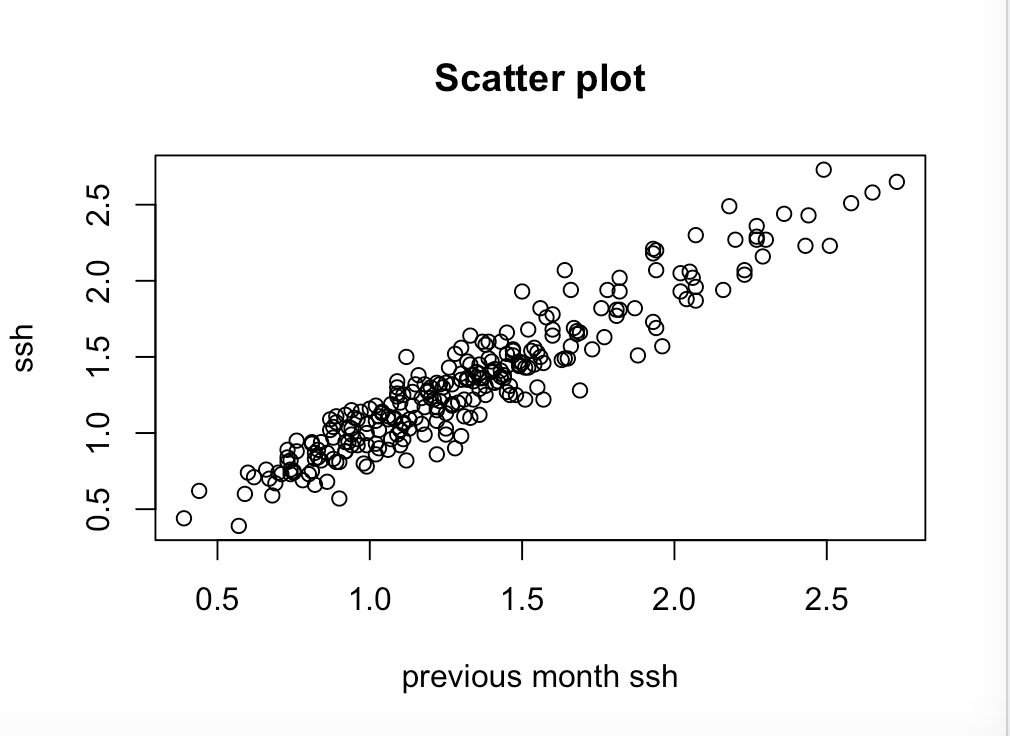
**start\_data=43489**

**end\_data=43752**

**plot(y = shanweisea[start\_data:end\_data], x = zlag (shanweisea[start\_data:end\_data]),**

**ylab ='ssh',**

**xlab = 'previous month ssh', main = "Scatter plot")**

****

The scatter plot of partial data indicates there might be a positive strong linear correlation between the sea surface height with its previous hour’s sea surface height.

Since the linear relationship is strong positive, the correlation coefficient r is closer to 1.

**6.5 Normality Checking**

The Q-Q plot, or quantile-quantile plot, is a graphical tool to help us assess if a set of data plausibly came from some theoretical distribution such as a Normal or exponential distribution. A Q-Q plot is a scatter plot created by plotting two sets of quantiles against one another.

**Definition**: **Quantiles** are points in your data below which a certain proportion of your data fall. For example, imagine the classic bell-curve standard Normal distribution with a mean of 0. The 0.5 quantile, or 50th percentile, is 0.

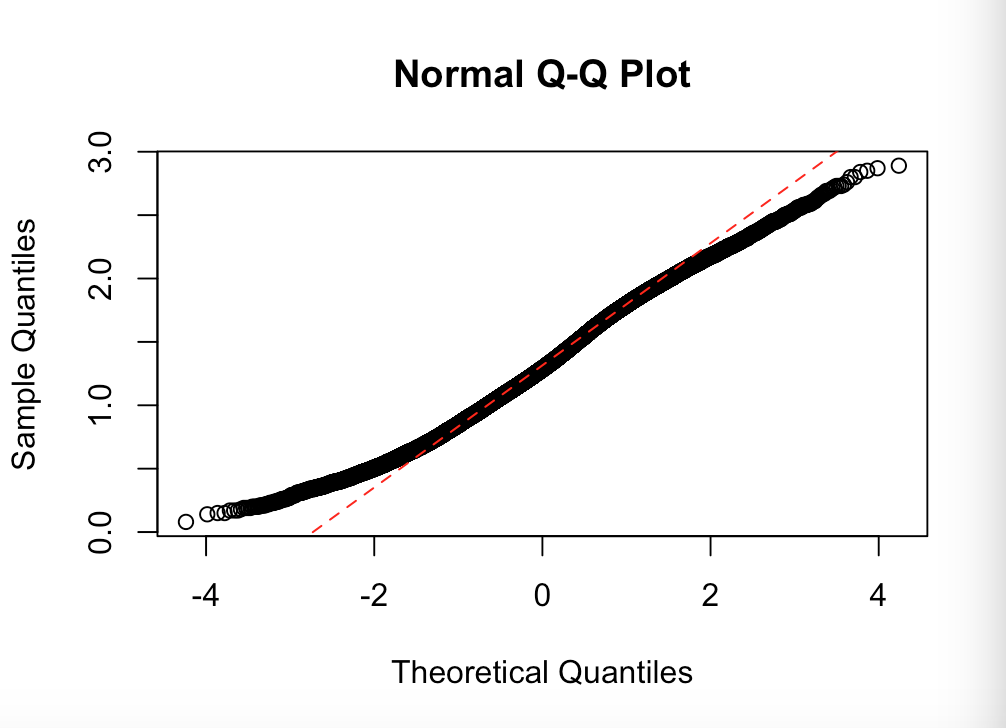
**Lemma**: If both sets of quantiles came from the same distribution, we should see the points forming a line that’s roughly straight.

**qqnorm(shanweisea)**

**qqline(shanweisea, col = 2, lwd = 1, lty = 2)**

(**qqnorm** creates a Normal Q-Q plot. You give it a vector of data and R plots the data in sorted order versus quantiles from a standard Normal distribution. **qqline** adds a line to a “theoretical”, by default normal, quantile-quantile plot which passes through the probs quantiles, by default the first and third quartiles.)

The Normal Q-Q Plot of the whole data is as follows:



Check the partial data as usual:

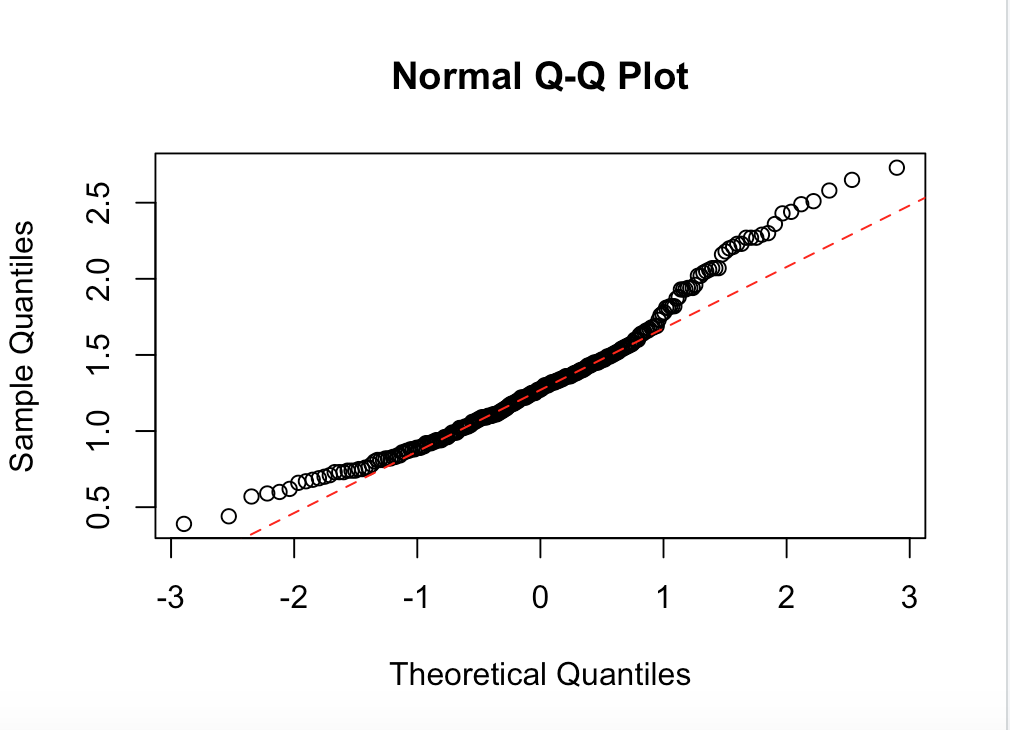
**start\_data=43489**

**end\_data=43752**

**qqnorm(shanweisea[start\_data:end\_data])**

**qqline(shanweisea[start\_data:end\_data], col = 2, lwd = 1, lty = 2)**

The Q-Q plot of the partial data is as follows:



Notice the points fall along a line in the middle of the graph, but curve off in the extremities. Normal Q-Q plots that exhibit this behavior usually mean the data have more extreme values than would be expected if they truly came from a Normal distribution.

Place Shapiro Test on partial data: (Since the Shapiro Test requires the sample size is between 3 to 5000, so I could not place this test on the whole data)

**start\_data=43489**

**end\_data=43752**

**shapiro.test(shanweisea[start\_data:end\_data])**

The output is:

Shapiro-Wilk normality test

data: shanweisea[start\_data:end\_data]

W = 0.95958, p-value = 9.718e-07

**Conclusion until now:** The Q-Q plot reveals it did not meet the requirements of normality, and Shapiro test also confirmed it with p-value less than 0.01, which rejects the null hypothesis that the data is normally distributed.

**6.6 Exponential Checking**

**Shanweisea <- rexp(shanweisea) #random sample from exponential distribution#**

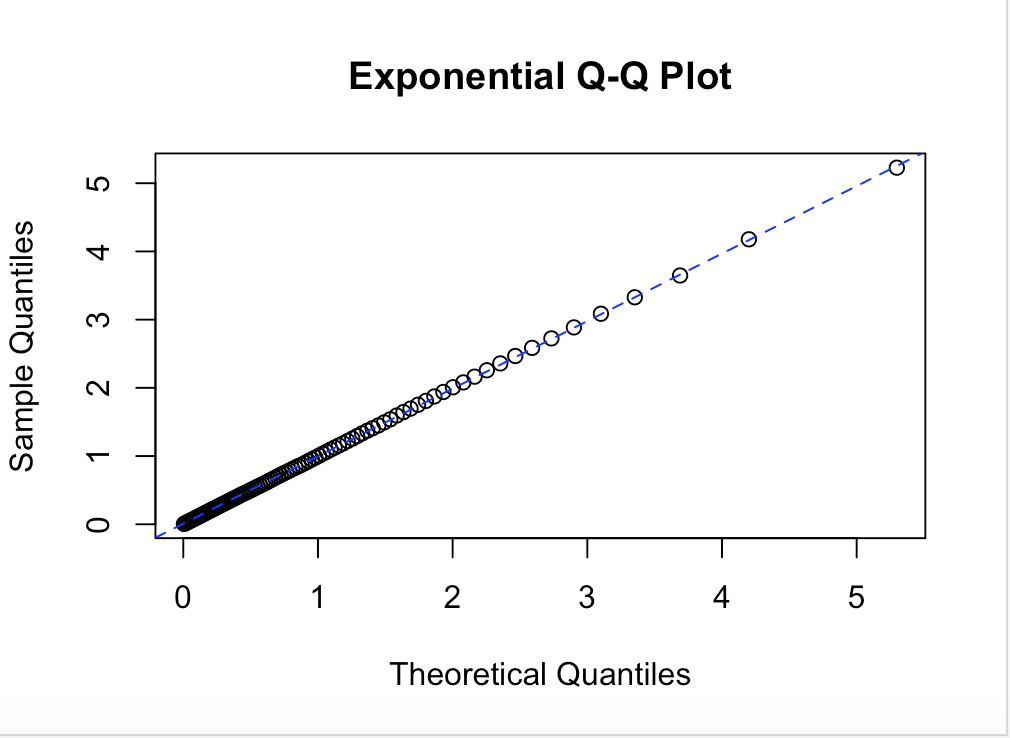
**p <- ppoints(100) #100 equally spaced points on (0,1), excluding endpoints#**

**q <- quantile(Shanweisea,p=p) #percentiles of the sample distribution#**

**plot(qexp(p) ,q, main="Exponential Q-Q Plot",**

**xlab="Theoretical Quantiles",ylab="Sample Quantiles")**

**qqline(q, distribution=qexp,col="blue", lty=2)**



Check the partial data:

**start\_data=43489**

**end\_data=43752**

**Shanweisea <- rexp(shanweisea[start\_data:end\_data])**

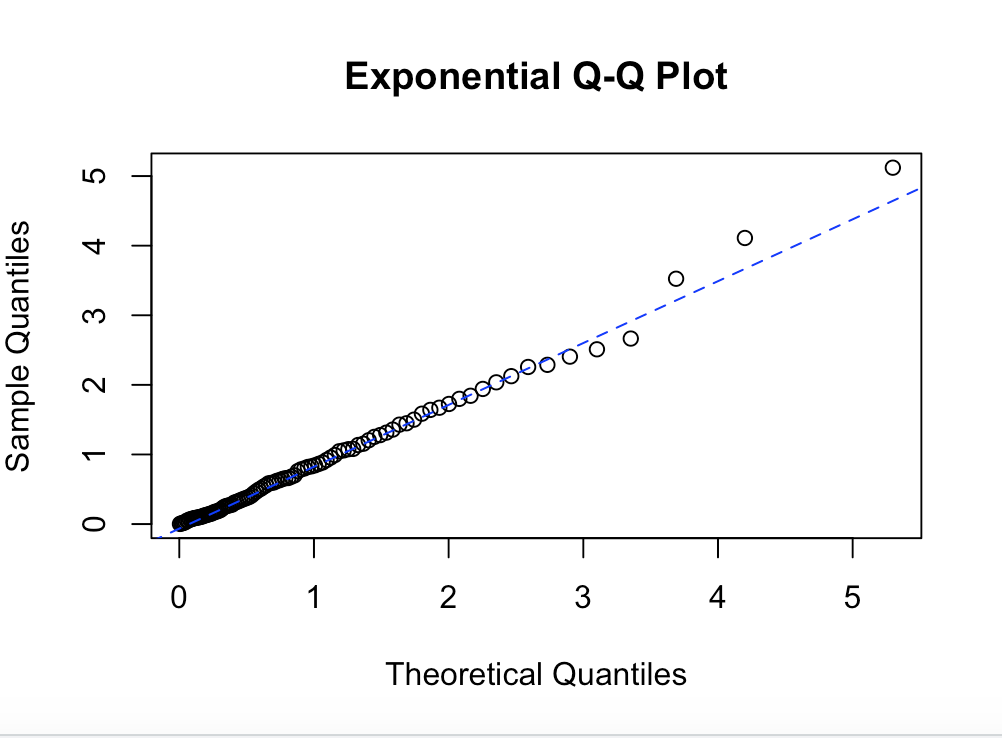
**p <- ppoints(100)**

**q <- quantile(Shanweisea,p=p)**

**plot(qexp(p) ,q, main="Exponential Q-Q Plot",**

**xlab="Theoretical Quantiles",ylab="Sample Quantiles")**

**qqline(q, distribution=qexp,col="blue", lty=2)**



Conclusion: From picture, the whole data seems approximate to exponential distribution but the partial data seems not.

**7 Modelling: ARIMA Model**

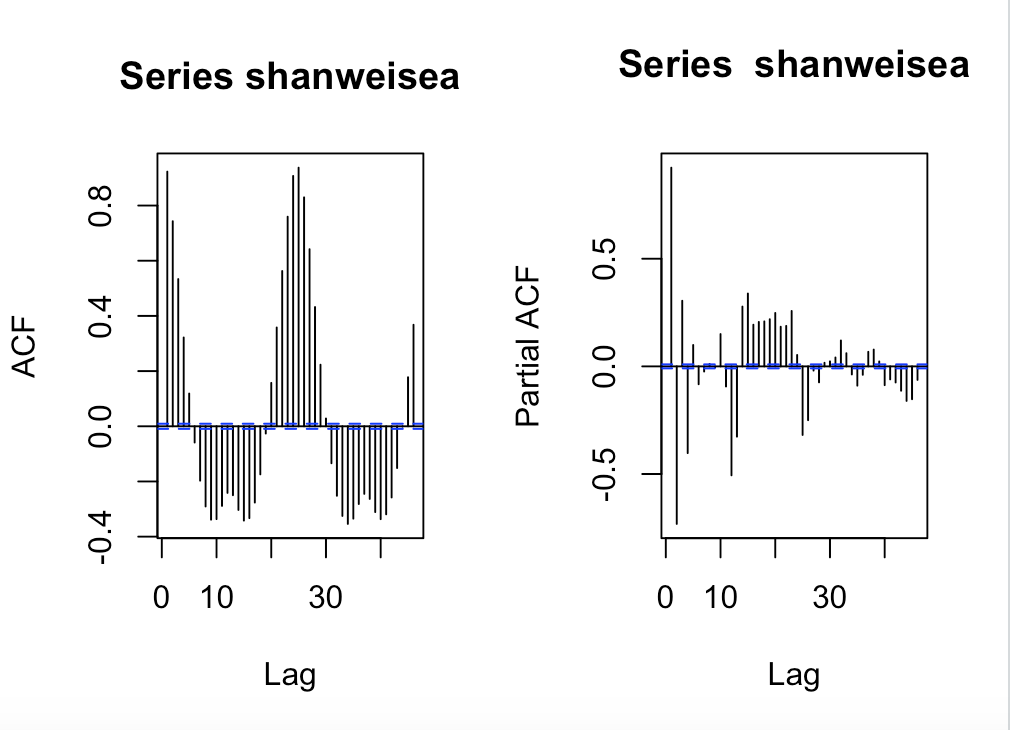
**7.1 Existence of stationarity on the whole data**

The ACF&PACF plot and ADF test to support evidence of stationary again.

**par(mfrow = c(1,2))**

**acf(shanweisea)**

**pacf(shanweisea)**



**par(mfrow = c(1,1))**

**adf.test(shanweisea)**

Augmented Dickey-Fuller Test

data: shanweisea

Dickey-Fuller = -14.462, Lag order = 35, p-value = 0.01

alternative hypothesis: stationary

Warning message:

In adf.test(shanweisea) : p-value smaller than printed p-value

Same procedure to test the partial data:

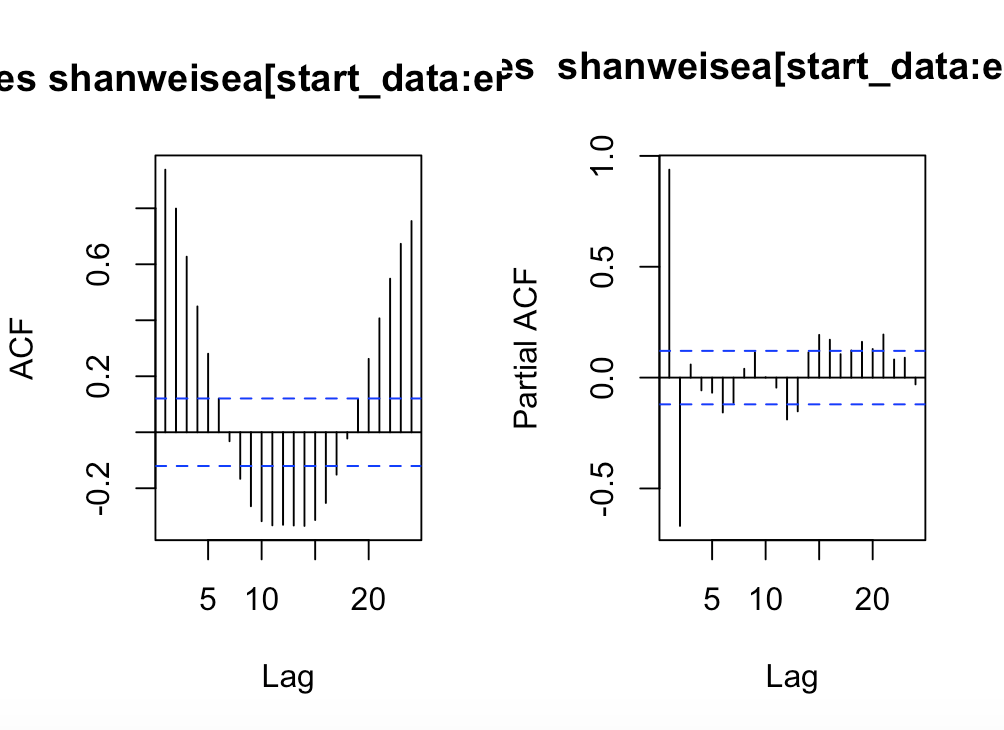
**start\_data=43489**

**end\_data=43752**

**par(mfrow = c(1,2))**

**acf(shanweisea[start\_data:end\_data])**

**pacf(shanweisea[start\_data:end\_data])**



**start\_data=43489**

**end\_data=43752**

**par(mfrow = c(1,1))**

**adf.test(shanweisea[start\_data:end\_data])**

Augmented Dickey-Fuller Test

data: shanweisea[start\_data:end\_data]

Dickey-Fuller = -6.5944, Lag order = 6, p-value = 0.01

With ADF test, p-value which is smaller than 0.01, indicates that we can reject the null hypothesis stating that the series is non-stationary.

Based on all the previous tests, I can make a conclusion that the data is stationary, not normally distributed, the whole data could be assumed as ARMA Model.

**Why not ARIMA for the whole data?**

ARIMA is a linear regression model which requires the data to be stationary, the first step to build an ARIMA model is to make the time series stationary since linear regression model works best if the predictors are not correlated and are independent of each other. If the data is non-stationary, the most common approach is to difference it., subtracting the previous value from the current value. If the time series is already stationary, then we do not need to subtract it.

We can write an ARMA(p,q) as a mixture of AR(p) and MA(q) models, such that:

xt  = wt + Φ1xt-1 + Φ2xt-2 + … + Φpxt-p + θ1wt-1 + θ2wt-2 + … + θqwt-q

**7.2 Fitting ARMA(**p,q**) models with arima()**

  The auto.arima() function in the **forecast** package will conduct an automatic search over all possible orders of ARIMA models that I specify.

**# find best ARMA(p,q) model**

**auto.arima(shanweisea, start.p = 0, max.p = 20, start.q = 0, max.q = 20, trace=1, seasonal=TRUE)**

I want to see the form for each of the models checked by auto.arima() and their associated AIC values, so I include the argument trace=1.

The report is here:

**Fitting models using approximations to speed things up...**

**ARIMA(0,0,0) with non-zero mean : 53708.61**

**ARIMA(0,0,0) with non-zero mean : 53708.61**

**ARIMA(1,0,0) with non-zero mean : -31406.58**

**ARIMA(0,0,1) with non-zero mean : -1930.227**

**ARIMA(0,0,0) with zero mean : 155550.1**

**ARIMA(2,0,0) with non-zero mean : -65609.09**

**ARIMA(3,0,0) with non-zero mean : -69951.21**

**ARIMA(4,0,0) with non-zero mean : -77841.16**

**ARIMA(5,0,0) with non-zero mean : -78280.43**

**ARIMA(6,0,0) with non-zero mean : -78585.3**

**ARIMA(7,0,0) with non-zero mean : -78608.18**

**ARIMA(8,0,0) with non-zero mean : -78610.87**

**ARIMA(9,0,0) with non-zero mean : -78608.96**

**ARIMA(8,0,1) with non-zero mean : -78669.51**

**ARIMA(7,0,1) with non-zero mean : -78666.81**

**ARIMA(9,0,1) with non-zero mean : -78724.66**

**ARIMA(10,0,1) with non-zero mean : -79672.99**

**ARIMA(10,0,0) with non-zero mean : -79621.9**

**ARIMA(11,0,1) with non-zero mean : -79869.35**

**ARIMA(11,0,0) with non-zero mean : -80011.57**

**ARIMA(12,0,0) with non-zero mean : -93223.75**

**ARIMA(13,0,0) with non-zero mean : -98246.03**

**ARIMA(14,0,0) with non-zero mean : -101831.1**

**ARIMA(15,0,0) with non-zero mean : -107259.2**

**ARIMA(16,0,0) with non-zero mean : Inf**

**ARIMA(15,0,1) with non-zero mean : Inf**

**ARIMA(14,0,1) with non-zero mean : Inf**

**ARIMA(16,0,1) with non-zero mean : Inf**

**ARIMA(15,0,0) with zero mean : Inf**

**Now re-fitting the best model(s) without approximations...**

**ARIMA(15,0,0) with non-zero mean : -107255.3**

**Best model: ARIMA(15,0,0) with non-zero mean**

**Series: shanweisea**

**ARIMA(15,0,0) with non-zero mean**

**Coefficients:**

**ar1 ar2 ar3 ar4 ar5 ar6 ar7 ar8**

**1.7893 -1.1736 0.6018 -0.4182 0.0796 0.0763 -0.1302 0.0715**

**s.e. 0.0045 0.0094 0.0108 0.0111 0.0112 0.0112 0.0112 0.0112**

**ar9 ar10 ar11 ar12 ar13 ar14 ar15 mean**

**-0.0132 -0.1510 0.2450 0.2624 -0.3329 -0.3597 0.3386 1.3119**

**s.e. 0.0112 0.0112 0.0112 0.0111 0.0108 0.0094 0.0045 0.0030**

**sigma^2 estimated as 0.00528: log likelihood=53644.67**

**AIC=-107255.3 AICc=-107255.3 BIC=-107107.4**

**>**

The result is a pure AR(15) model, I want to check whether there is some other possibilities so I lower down the order of **max.p** and **max.q:**

**# find best ARMA(p,q) model**

**auto.arima(shanweisea, start.p = 0, max.p = 13, start.q = 0, max.q = 13, trace=1, seasonal=TRUE)**

The report is here:

**Fitting models using approximations to speed things up...**

**ARIMA(0,0,0) with non-zero mean : 53708.61**

**ARIMA(0,0,0) with non-zero mean : 53708.61**

**ARIMA(1,0,0) with non-zero mean : -31406.58**

**ARIMA(0,0,1) with non-zero mean : -1930.227**

**ARIMA(0,0,0) with zero mean : 155550.1**

**ARIMA(2,0,0) with non-zero mean : -65609.09**

**ARIMA(3,0,0) with non-zero mean : -69951.21**

**ARIMA(4,0,0) with non-zero mean : -77841.16**

**ARIMA(5,0,0) with non-zero mean : -78280.43**

**ARIMA(6,0,0) with non-zero mean : -78585.3**

**ARIMA(7,0,0) with non-zero mean : -78608.18**

**ARIMA(8,0,0) with non-zero mean : -78610.87**

**ARIMA(9,0,0) with non-zero mean : -78608.96**

**ARIMA(8,0,1) with non-zero mean : -78669.51**

**ARIMA(7,0,1) with non-zero mean : -78666.81**

**ARIMA(9,0,1) with non-zero mean : -78724.66**

**ARIMA(10,0,1) with non-zero mean : -79672.99**

**ARIMA(10,0,0) with non-zero mean : -79621.9**

**ARIMA(11,0,1) with non-zero mean : -79869.35**

**ARIMA(11,0,0) with non-zero mean : -80011.57**

**ARIMA(12,0,0) with non-zero mean : -93223.75**

**ARIMA(13,0,0) with non-zero mean : -98246.03**

**ARIMA(13,0,1) with non-zero mean : -99490.78**

**ARIMA(12,0,1) with non-zero mean : -95228.84**

**ARIMA(13,0,2) with non-zero mean : -103299.7**

**ARIMA(12,0,2) with non-zero mean : Inf**

**ARIMA(13,0,3) with non-zero mean : -103594.2**

**ARIMA(12,0,3) with non-zero mean : Inf**

**ARIMA(13,0,4) with non-zero mean : Inf**

**ARIMA(12,0,4) with non-zero mean : Inf**

**ARIMA(13,0,3) with zero mean : Inf**

**Now re-fitting the best model(s) without approximations...**

**ARIMA(13,0,3) with non-zero mean : -103591**

**Best model: ARIMA(13,0,3) with non-zero mean**

**Series: shanweisea**

**ARIMA(13,0,3) with non-zero mean**

**Coefficients:**

**ar1 ar2 ar3 ar4 ar5 ar6 ar7 ar8**

**1.6797 -1.3431 0.8719 -0.3871 -0.0316 0.1365 -0.1991 0.0690**

**s.e. 0.0153 0.0367 0.0469 0.0436 0.0315 0.0187 0.0126 0.0123**

**ar9 ar10 ar11 ar12 ar13 ma1 ma2 ma3**

**-0.0119 -0.1945 0.2294 0.2923 -0.4081 0.1176 0.2898 0.1988**

**s.e. 0.0125 0.0123 0.0128 0.0140 0.0077 0.0154 0.0126 0.0111**

**mean**

**1.3119**

**s.e. 0.0019**

**sigma^2 estimated as 0.005732: log likelihood=51813.5**

**AIC=-103591 AICc=-103591 BIC=-103434.3**

**>**

The result gives us a ARMA(13,3) model.

As the routine, I checked the partial data:

**#find best ARMA(p,q) model for partial data**

**start\_data=43489**

**end\_data=43752**

**auto.arima(shanweisea[start\_data:end\_data], start.p = 0, max.p = 10, start.q = 0, max.q = 10, trace=1, seasonal=TRUE)**

The report is here:

**Fitting models using approximations to speed things up...**

**ARIMA(0,1,0) with drift : -246.4871**

**ARIMA(0,1,0) with drift : -246.4871**

**ARIMA(1,1,0) with drift : -461.4528**

**ARIMA(0,1,1) with drift : -448.3511**

**ARIMA(0,1,0) : -248.3974**

**ARIMA(2,1,0) with drift : -509.7645**

**ARIMA(3,1,0) with drift : -509.3677**

**ARIMA(2,1,1) with drift : -509.7187**

**ARIMA(1,1,1) with drift : -511.5507**

**ARIMA(1,1,2) with drift : -510.0737**

**ARIMA(0,1,2) with drift : -490.2745**

**ARIMA(2,1,2) with drift : Inf**

**ARIMA(1,1,1) : -513.5978**

**ARIMA(0,1,1) : -450.1405**

**ARIMA(1,1,0) : -463.4852**

**ARIMA(2,1,1) : -511.7774**

**ARIMA(1,1,2) : -512.1347**

**ARIMA(0,1,2) : -492.0048**

**ARIMA(2,1,0) : -511.791**

**ARIMA(2,1,2) : -534.0946**

**ARIMA(3,1,2) : -512.8023**

**ARIMA(2,1,3) : Inf**

**ARIMA(1,1,3) : -514.0881**

**ARIMA(3,1,1) : -513.7623**

**ARIMA(3,1,3) : Inf**

**Now re-fitting the best model(s) without approximations...**

**ARIMA(2,1,2) : -539.6761**

**Best model: ARIMA(2,1,2)**

**Series: shanweisea[start\_data:end\_data]**

**ARIMA(2,1,2)**

**Coefficients:**

**ar1 ar2 ma1 ma2**

**1.5452 -0.6521 -0.5183 -0.4623**

**s.e. 0.0569 0.0569 0.0712 0.0710**

**sigma^2 estimated as 0.00726: log likelihood=274.95**

**AIC=-539.91 AICc=-539.68 BIC=-522.05**

**>**

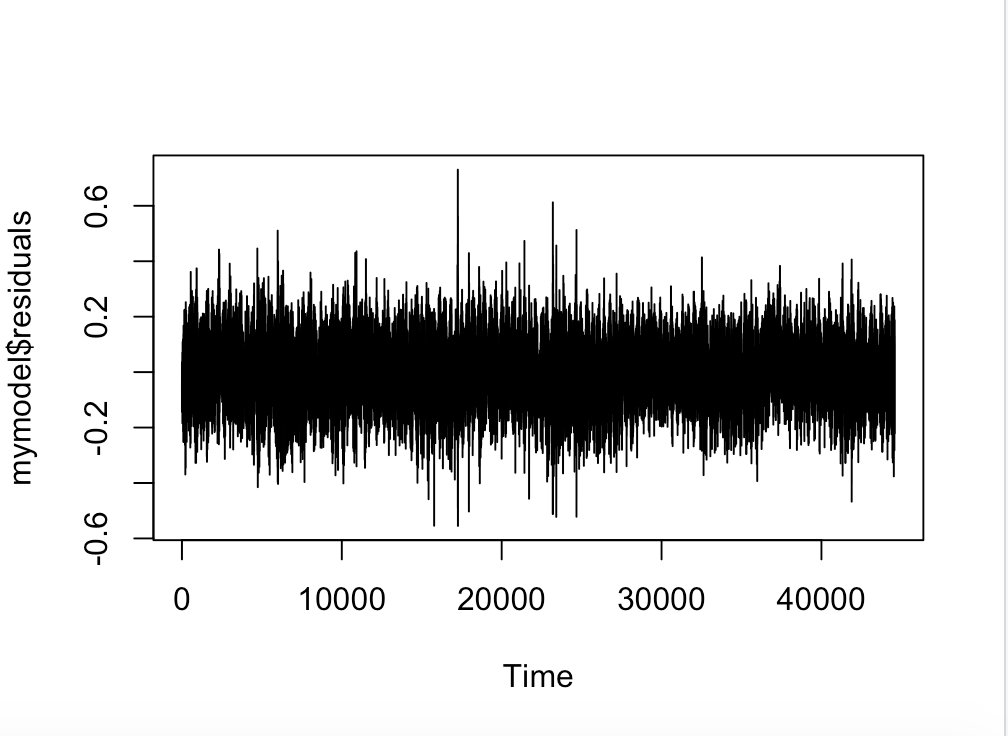
The funny is that, Dickey-Fuller Test says the partial data is stationary, however, the auto.arima() function found the best model fit for the partial data is ARIMA (2,1,2) which means the data is not stationary. What’s the contradiction here?

**7.3 Forecasting the ARIMA model**

Check whether the model is fit, Let’s plot some graphs:

**mymodel = auto.arima(shanweisea)**

**plot(mymodel$residuals)**

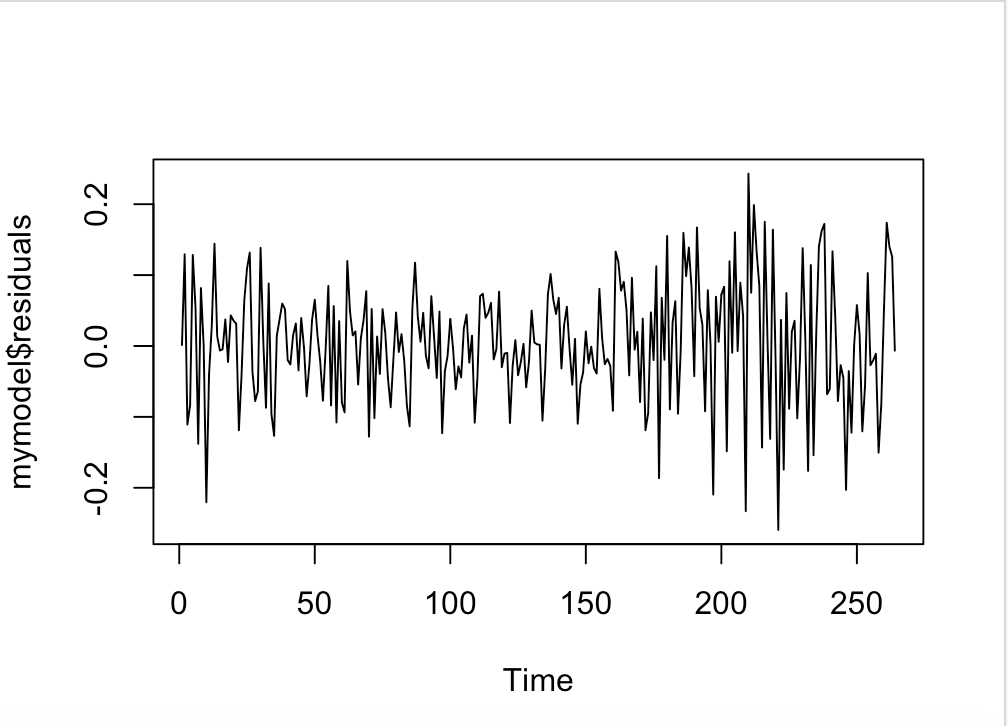
****

**start\_data=43389**

**end\_data=43752**

**mymodel = auto.arima(shanweisea[start\_data:end\_data])**

**plot(mymodel$residuals)**

****

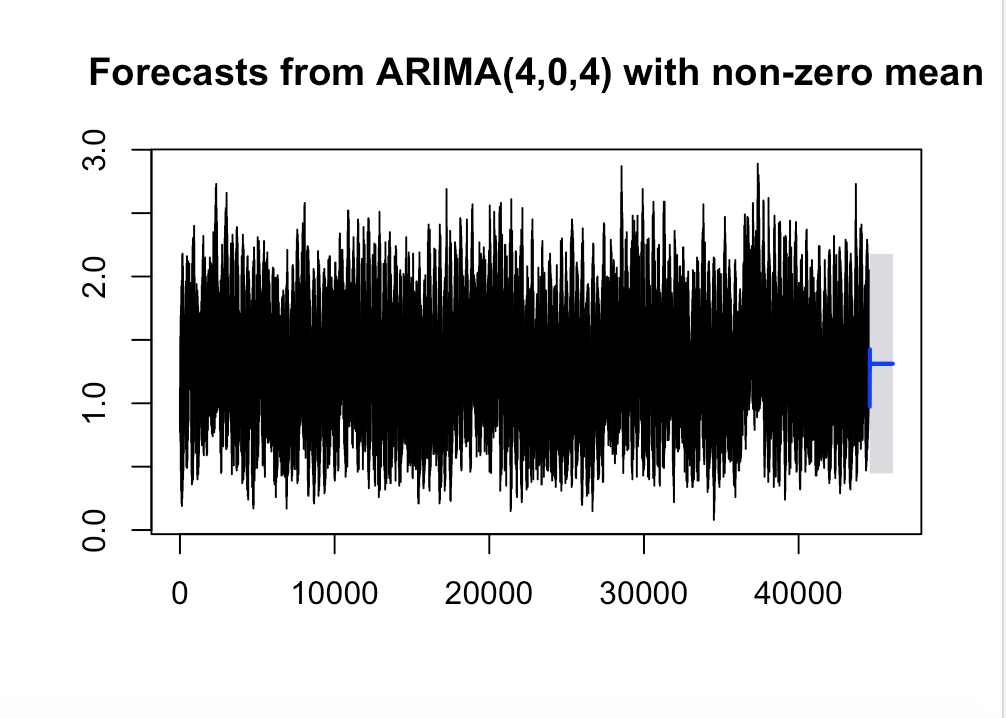
For the whole data, the residuals are pretty steady which means the model is good, for the partial data, the residuals are well except from about 170 to 250, which is exactly the happening of typhoon that time.

**Plot the forecasting:**

**mymodel = auto.arima(shanweisea)**

**myforecast = forecast(mymodel,level = c(95), h = 4000\*1)**

**plot(myforecast)**

****

**And plot the forecasting from the partial data:**

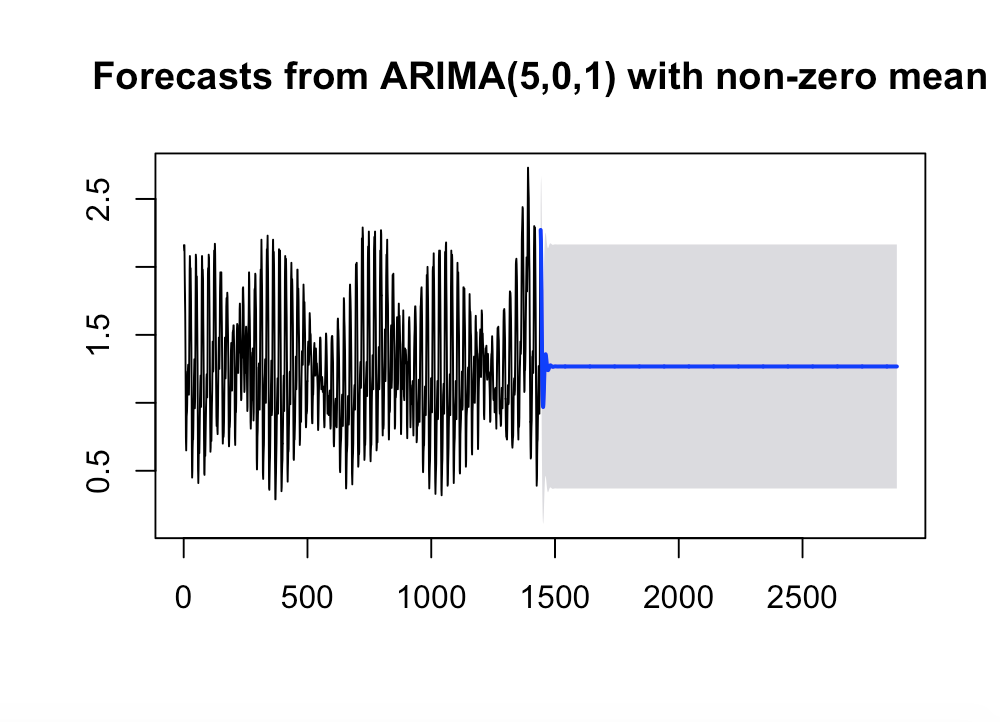
**start\_data=42312**

**end\_data=43752**

**mymodel = auto.arima(shanweisea[start\_data:end\_data])**

**myforecast = forecast(mymodel,level = c(95), h = 1440\*1)**

**plot(myforecast)**

****

Based on different chosen data, the model is different, however, the prediction is pretty flat which is abnormal. Why? That is because the data has seasonality. When tested by

**Frequency(shanweisea)**

The result is 1, which means the hourly data has a monthly period seasonality, 24\*30 = 720. However, when chose the code like this:

**start\_data=42312**

**end\_data=43752**

**mymodel = auto.arima(shanweisea[start\_data:end\_data])**

**myforecast = forecast(mymodel,level = c(95), h = 2\*720)**

**plot(myforecast)**

The plot is the same.

Temporary conclusion based on now:

1. I have a fairly complex time series which can not be easily solved with simple tools.
2. The data is stationary in tools of Dicky-Fuller test, but non-stationary on partial data.

All that suggests we may need another models .

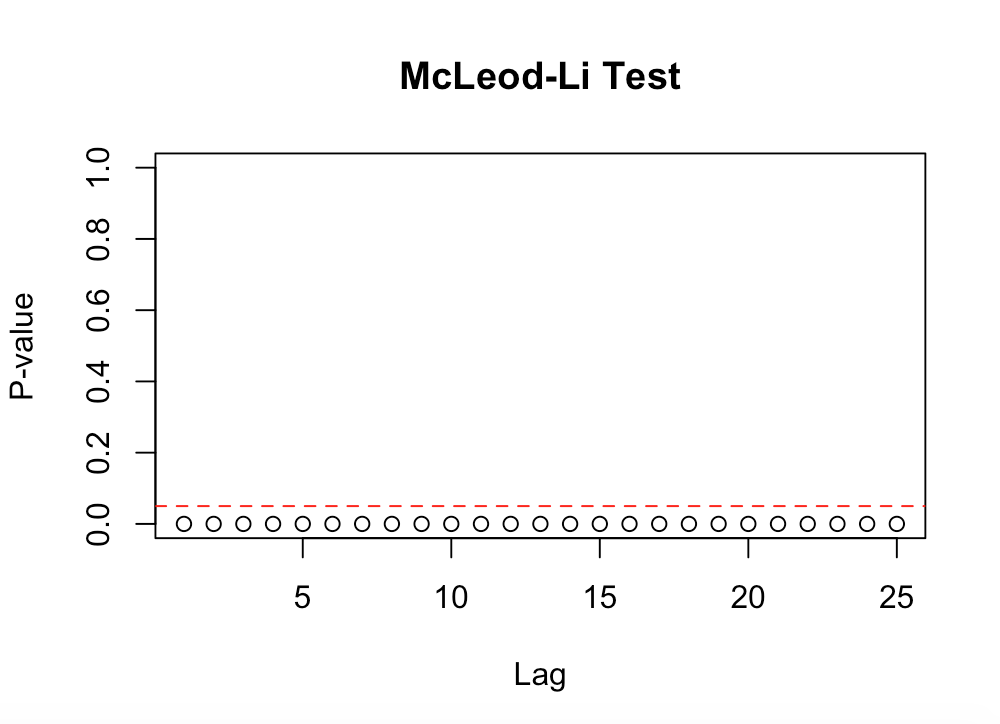
**8 Modelling: ARCH model and effect**

**8.1 McLeod-Li Test Statistics for Absolute value for partial data**

**start\_data=43389**

**end\_data=43752**

**McLeod.Li.test(y=abs(shanweisea[start\_data:end\_data]),main="McLeod-Li Test")**



McLeod-Li test give me strong belief that the volatility clustering exists.

**8.2 Order Specification**

**start\_data=43489**

**end\_data=43752**

**eacf(shanweisea[start\_data:end\_data])**

AR/MA

0 1 2 3 4 5 6 7 8 9 10 11 12 13

0 x x x x x o o x x x x x x x

1 x x x x x x o x x x x o o x

2 x o o x o x x o x x x o o o

3 x x o x o x x o o x x o o o

4 o x o o o o o o o o o x o o

5 o x x x o o o o o o o x o o

6 x x x x o o o o o o o x x o

7 x x x o o o o o o o o x o x

>

The eacf test suggest GARCH (1,6) , GARCH (1,11), GARCH (1,12), GARCH (2,1) , GARCH (2,2) , GARCH (2,4), GARCH (2,7), GARCH (2,11), GARCH (2,12) and GARCH (2,13) models for the partial data.

**8.3 Model Fitting, Estimation and Diagnostic Checking**

I will fit the models found by eacf test and check the p-values for them.

**8.3.1 GARCH (2,1)**

**garch21 = garch(shanweisea, order = c(2,1), trace = FALSE)**

**summary(garch21)**

Call:

garch(x = shanweisea[start\_data:end\_data], order = c(2, 1), trace = FALSE)

Model:

GARCH(2,1)

Residuals:

Min 1Q Median 3Q Max

0.4341 0.8127 0.9140 0.9848 1.2277

Coefficient(s):

Estimate Std. Error t value Pr(>|t|)

a0 5.109e-01 1.842e+00 0.277 0.781

a1 8.619e-01 3.641e+00 0.237 0.813

b1 1.314e-02 6.341e+00 0.002 0.998

b2 5.010e-15 3.087e+00 0.000 1.000

Diagnostic Tests:

Jarque Bera Test

data: Residuals

X-squared = 9.2094, df = 2, p-value = 0.01

Box-Ljung test

data: Squared.Residuals

X-squared = 181.25, df = 1, p-value < 2.2e-16

All other models have much bigger significant P-value. Which means GARCH model fails.

**8.3.2 ARCH (1)**

**start\_data=43489**

**end\_data=43752**

**arch.fit <- garchFit(~garch(1,0), data = shanweisea[start\_data:end\_data], trace = F)**

**summary(arch.fit)**

Title:

GARCH Modelling

Call:

garchFit(formula = ~garch(1, 0), data = shanweisea[start\_data:end\_data],

trace = F)

Mean and Variance Equation:

data ~ garch(1, 0)

<environment: 0x11bab4d58>

[data = shanweisea[start\_data:end\_data]]

Conditional Distribution:

norm

Coefficient(s):

mu omega alpha1

1.202395 0.020601 0.895472

Std. Errors:

based on Hessian

Error Analysis:

Estimate Std. Error t value Pr(>|t|)

mu 1.202395 0.027363 43.943 < 2e-16 \*\*\*

omega 0.020601 0.004495 4.583 4.58e-06 \*\*\*

alpha1 0.895472 0.121968 7.342 2.11e-13 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Log Likelihood:

-69.33813 normalized: -0.2626444

Description:

Fri May 1 23:27:28 2020 by user:

Standardised Residuals Tests:

Statistic p-Value

Jarque-Bera Test R Chi^2 12.0688 0.002394929

Shapiro-Wilk Test R W 0.9572407 5.03199e-07

Ljung-Box Test R Q(10) 439.2761 0

Ljung-Box Test R Q(15) 572.6593 0

Ljung-Box Test R Q(20) 613.4716 0

Ljung-Box Test R^2 Q(10) 65.39069 3.412263e-10

Ljung-Box Test R^2 Q(15) 90.08247 9.573453e-13

Ljung-Box Test R^2 Q(20) 93.46206 1.828049e-11

LM Arch Test R TR^2 73.9278 5.856327e-11

Information Criterion Statistics:

AIC BIC SIC HQIC

0.5480162 0.5886520 0.5477618 0.5643449

>

The p-value is in significant level.

**9** **Final Best Model Selection for ARMA + ARCH**

**9.1 ARCH (1) + ARIMA(2,1,2)**

Based on the above conclusions, we chose the best model as model ARCH(1) + ARIMA (2,1,2) for the partial data.

**Model ARCH(1) is**:

Yt  = 1.202395 + σt εt, , with σt = **√**0.020601 + 0.895472 Y2t-1

εt, ~ N(0,1) is a white noise

**Model ARIMA(2,1,2) is**: Yt – Yt-1 = μ + 1.5452 (Yt-1 – Yt-2) - 0.6521 (Yt-2 – Yt-3) – 0.5183 εt-1 – 0.4623 εt-2 + εt

**Combination of Model ARCH(1) + ARIMA(2,1,2) is**:

Yt  = 1.202395 + (**√**0.020601 + 0.895472 Y2t-1)\*εt, + Yt-1 + μ + 1.5452 (Yt-1 – Yt-2) - 0.6521 (Yt-2 – Yt-3) – 0.5183 εt-1 – 0.4623 εt-2 + εt

**Checking by codes:**

**shanweisea = shanwei$ssh**

**start\_data=43489**

**end\_data=43752**

**partial = shanweisea[43489:43752]**

**partiall = diff(partial, differences = 1)**

**model=ugarchspec(**

**variance.model = list(model = "sGARCH", garchOrder = c(0, 1)),**

**mean.model = list(armaOrder = c(2,2), include.mean = TRUE),**

**distribution.model = "norm", fixed.pars=list(mu = 1.202395, omega=0.020601, alpha1=0.895472)**

**)**

**modelfit=ugarchfit(spec=model,data=partiall)**

**modelfit**

**Reports:**

\*---------------------------------\*

\* GARCH Model Fit \*

\*---------------------------------\*

Conditional Variance Dynamics

-----------------------------------

GARCH Model : sGARCH(0,1)

Mean Model : ARFIMA(2,0,2)

Distribution : norm

Optimal Parameters

------------------------------------

Estimate Std. Error t value Pr(>|t|)

mu 1.202395 NA NA NA

ar1 1.520726 0.000274 5544.3 0

ar2 -0.347653 0.000043 -8062.7 0

ma1 -0.568267 0.000115 -4953.9 0

ma2 -0.630608 0.000126 -5004.8 0

omega 0.020601 NA NA NA

beta1 0.900191 0.000113 7999.0 0

Robust Standard Errors:

Estimate Std. Error t value Pr(>|t|)

mu 1.202395 NA NA NA

ar1 1.520726 NaN NaN NaN

ar2 -0.347653 NaN NaN NaN

ma1 -0.568267 NaN NaN NaN

ma2 -0.630608 NaN NaN NaN

omega 0.020601 NA NA NA

beta1 0.900191 NaN NaN NaN

LogLikelihood : -10666.04

Information Criteria

------------------------------------

Akaike 81.149

Bayes 81.217

Shibata 81.148

Hannan-Quinn 81.176

Weighted Ljung-Box Test on Standardized Residuals

------------------------------------

statistic p-value

Lag[1] 188.3 0

Lag[2\*(p+q)+(p+q)-1][11] 508.2 0

Lag[4\*(p+q)+(p+q)-1][19] 565.6 0

d.o.f=4

H0 : No serial correlation

Weighted Ljung-Box Test on Standardized Squared Residuals

------------------------------------

statistic p-value

Lag[1] 133.3 0

Lag[2\*(p+q)+(p+q)-1][2] 166.9 0

Lag[4\*(p+q)+(p+q)-1][5] 215.7 0

d.o.f=1

Weighted ARCH LM Tests

------------------------------------

Statistic Shape Scale P-Value

ARCH Lag[2] 66.06 0.500 2.000 4.441e-16

ARCH Lag[4] 99.16 1.397 1.611 0.000e+00

ARCH Lag[6] 110.21 2.222 1.500 0.000e+00

Nyblom stability test

------------------------------------

Joint Statistic: NaN

Individual Statistics:

ar1 NaN

ar2 88.17

ma1 NaN

ma2 NaN

beta1 NaN

Asymptotic Critical Values (10% 5% 1%)

Joint Statistic: 1.28 1.47 1.88

Individual Statistic: 0.35 0.47 0.75

Sign Bias Test

The code returns us an optimal coefficient value.

Yt  = 1.202395 + (**√**0.020601 + 0.900191 Y2t-1)\*εt, + Yt-1 + μ + 1.520726 (Yt-1 – Yt-2) -0.347653(Yt-2 – Yt-3) -0.568267εt-1 –0.630608 εt-2 + εt

**10 Forecasting**

**Remark: Rolling forecasts are used for long-term forecasting in general. However, Let’s see for fun what happens here:**

**modelfit=ugarchfit(spec=model,data=partiall)**

**fore = ugarchforecast(modelfit,data=partiall,n.ahead = 10)**

**fore**

\*------------------------------------\*

\* GARCH Model Forecast \*

\*------------------------------------\*

Model: sGARCH

Horizon: 10

Roll Steps: 0

Out of Sample: 0

0-roll forecast [T0=1970-09-20 20:00:00]:

Series Sigma

T+1 7.104e+11 87623

T+2 1.478e+12 83136

T+3 2.000e+12 78878

T+4 2.528e+12 74838

T+5 3.149e+12 71005

T+6 3.910e+12 67369

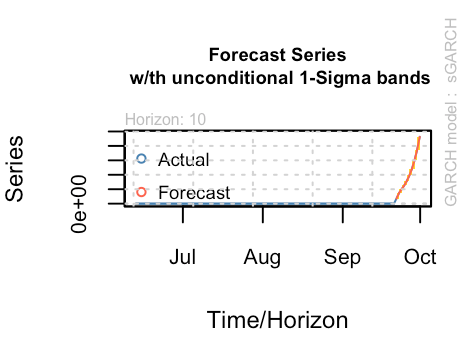
T+7 4.851e+12 63918

T+8 6.018e+12 60645

T+9 7.465e+12 57539

T+10 9.260e+12 54592

**plot (modelfit, which = “all”)**

****

**11 Conclusion**

**Conclusion: We are pretty sure the whole sea surface height obeys a steadier model but the partial data which contains the event of Typhoon displays a totally different model. The pseudo forecasting says the time series would be slightly increasing in the following days, which says it is a good model in use of influence of Typhoon. For forecasting, we could pick some period of time and fit the model in up until it fails, it prove the things are good, if succeeds, there may be a hint that something would happen soon. Different cities, different sea area may have different model to fit in, but the key point remains same: The whole data obeys a AR or ARMA model, if some things would happen, ARMR+ARCH model would fit.**