post2 test

Zishen Cheng

12/3/2017

More about Time series

Introduction & motivation

In the previous post I introduced many common functions with package time seires, such as: residual analysis, and how to interpret auto correlation function graph and partial auto correlation function graph. Also I introduced an useful model in fitting data.
In this post, I'm going to further introduce you an more advanced model(seasonal arima) and how to use them to do data prediction.
Time series is so useful in all fields that I cannot stop talking about. fitting a model is crucial to time series analysis, but doing prediction is what we eventually want.

Terminology introduction.

AIC: Akaike information criterion. The criterion provide how fit the data is to your model. The smoller the number is the better the fitting. The information number can be negative.

AR: Auto regression model. It takes previous data points as reference, weith them, and predict what should the next data point would be. Take one parameter.

MA:Moving average model. It takes prvious changing factors or residuals as reference, weight them, and predict what should come next. Take one parameter.

ARMA: Auto correlation and moving average model. Combination of AR and MA. Take two parameters: first is for AR, second is for MA.

ARIMA: Auto correlation integral moving average model. It is basically the ARMA of data's difference. Take three parametes: first for AR, second indicate how many difference taken, third for MA.

ACF: Auto correlation function. The correlation for time series data. an indicator of MA's parameter.

PACF: Partial auto correlation function. An indicator of AR's parameter.

Importing data and loading packages

the dataset is from my personal record about daily walking steps number. library required are TSA, lmtest and forecast.

```
library(RCurl)
```

```
## Loading required package: bitops
```

```
URL <- "https://raw.githubusercontent.com/zcheng7/hurry/master/walking%20data.csv"
walking_dat <- as.ts(read.csv(file = URL))

library(TSA)
```

```
## Loading required package: leaps
```

```
## Loading required package: locfit
```

```
## locfit 1.5-9.1    2013-03-22
```

```
## Loading required package: mgcv
```

```
## Loading required package: nlme
```

```
## This is mgcv 1.8-15. For overview type 'help("mgcv-package")'.
```

```
## Loading required package: tseries
```

```
##
## Attaching package: 'TSA'
```

```
## The following objects are masked from 'package:stats':
##
##     acf, arima
```

```
## The following object is masked from 'package:utils':
##
##     tar
```

```
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
##
## Attaching package: 'lmtest'
```

```
## The following object is masked from 'package:RCurl':
##
##     reset
```

```
library(forecast)
```

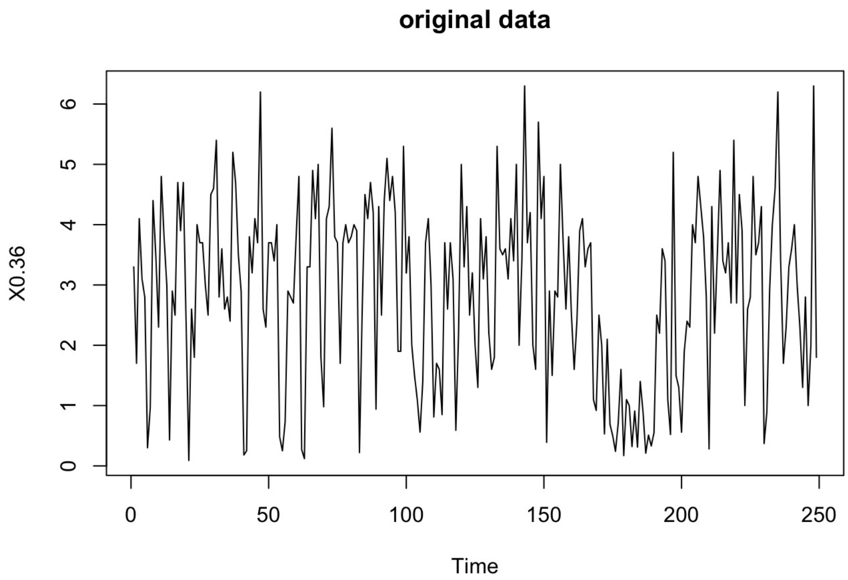
```
##
## Attaching package: 'forecast'
```

```
## The following object is masked from 'package:nlme':
##
##     getResponse
```

From previous post

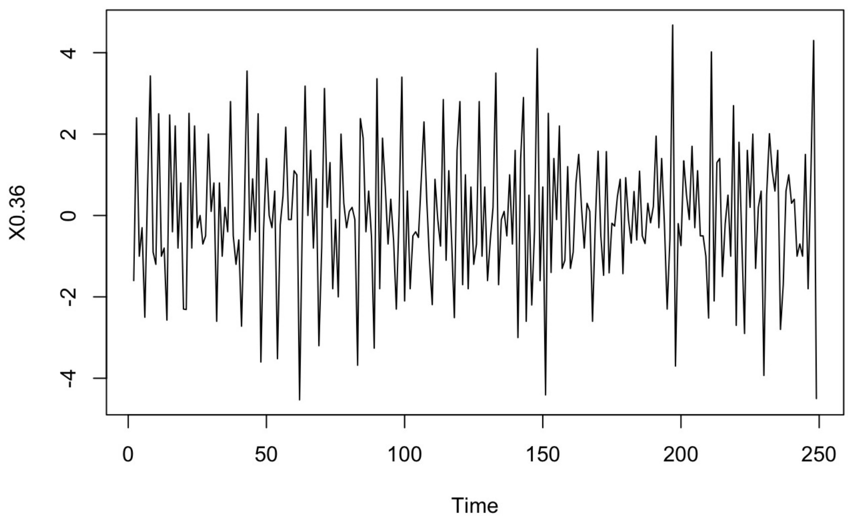
The following plots have been introduced in my first post. Here I just will plot them out.

```
plot.ts(walking_dat)
title("original data")
```



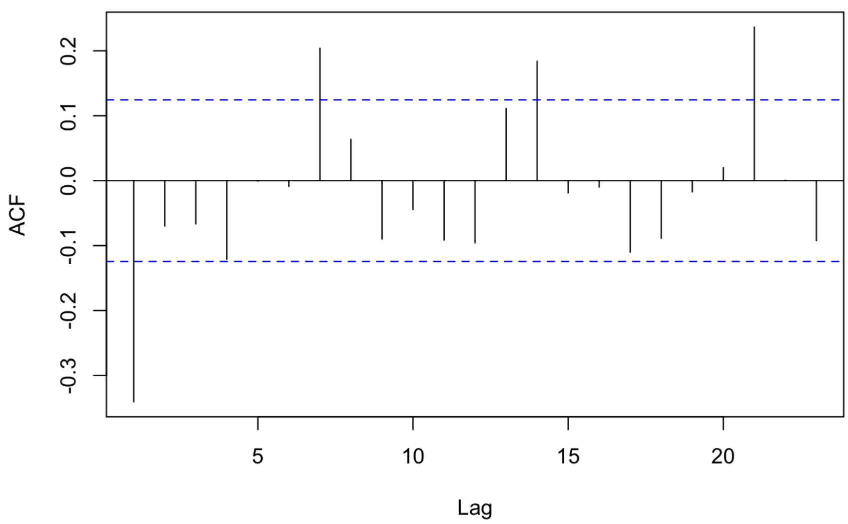

```
plot.ts(diff(walking_dat))
title("first differenced data")
```

first differenced data



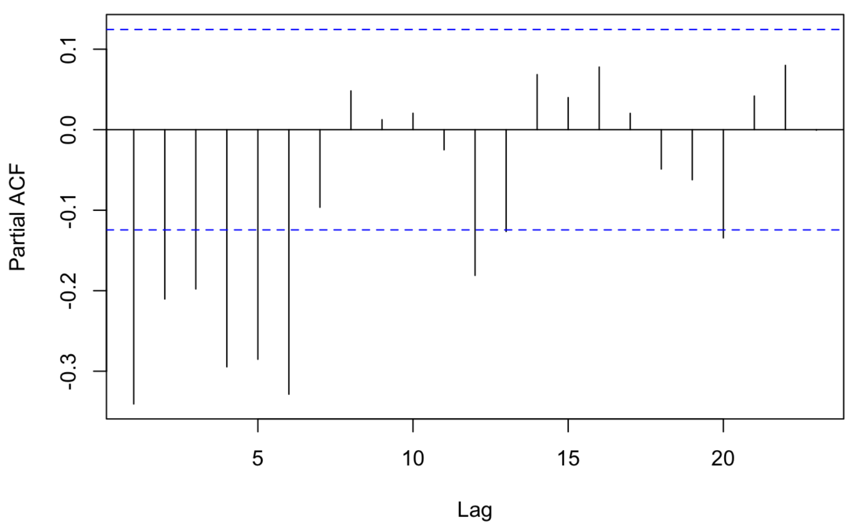

```
acf(diff(walking_dat))
```

Series diff(walking_dat)




```
pacf(diff(walking_dat))
```

Series diff(walking_dat)



```
eacf(diff(walking_dat))
```

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x o o o o o x o o o o  o  o  x
## 1 x o o x o o x x o o o  x  o  x
## 2 x o o x o o o o o o o  o  o  x
## 3 x x x o o o x o o o o  o  o  x
## 4 x x x o o o x o o o o  o  o  o
## 5 x o x x x x o o o o o  o  o  o
## 6 x x o x x x x o o o o  o  o  o
## 7 x x o o o x x o o o o  o  o  o
```

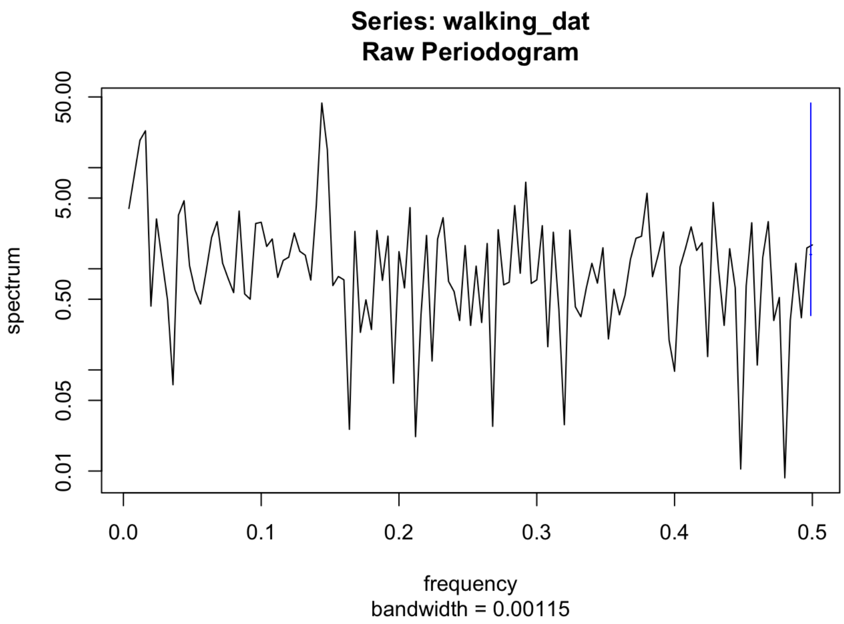
how above works

The above part was covered in my previous post and here is the link:https://github.com/zcheng7/stat133-hws-fall17/blob/master/labs/post1.Rmd However, except what we've talked about, from the above auto correlation function graph and partial auto correlation function graph, we can clearly observe some periodic pattern within them. Therefore we can draw down the conclusion that there exists a seasonality factor with in the dataset. Which means that data would follow roughly same pattern with in some period. We want to fit the differenced data with seasonality model. So following is what I'm going to do.

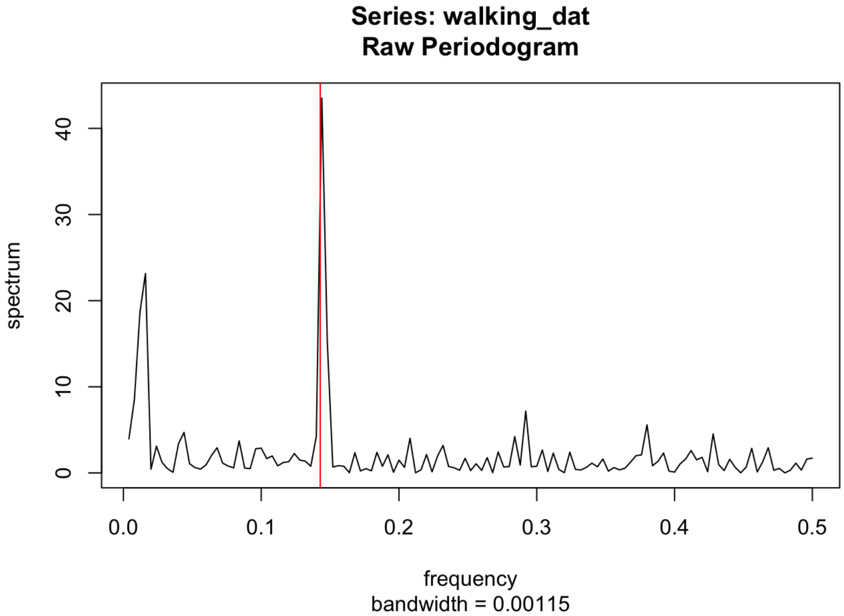
Seasonality analysis

this part is called spectrum analysis. The plot above give us an intuitive understanding that for what frequency of pattern happens most frequently in my dataset. And from the line I fitted, we can say that my data have a period of seven days. Seven days is very reasonable since I usually have same schedule for same weekday.

```
plot(spec.pgram(walking_dat, taper = 0), log = "no")
```

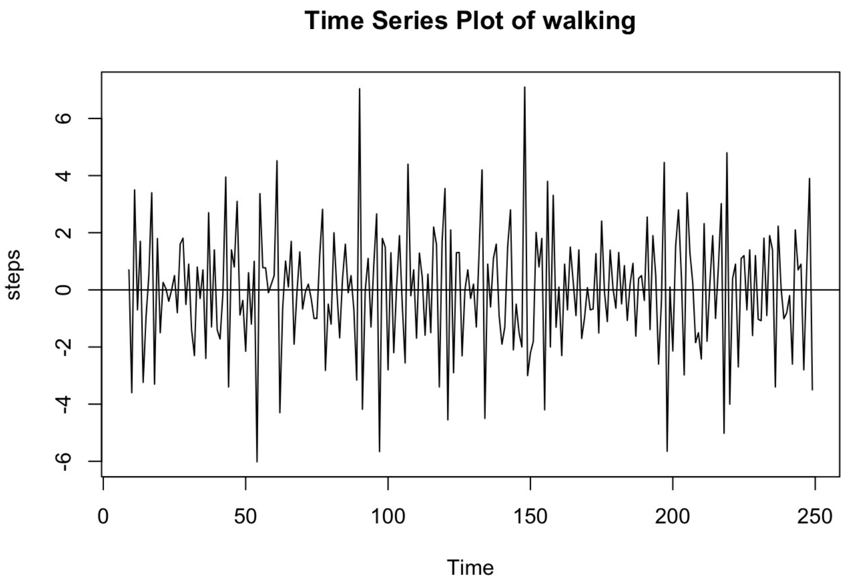



```
abline(v = 1/7, col = "red")
```

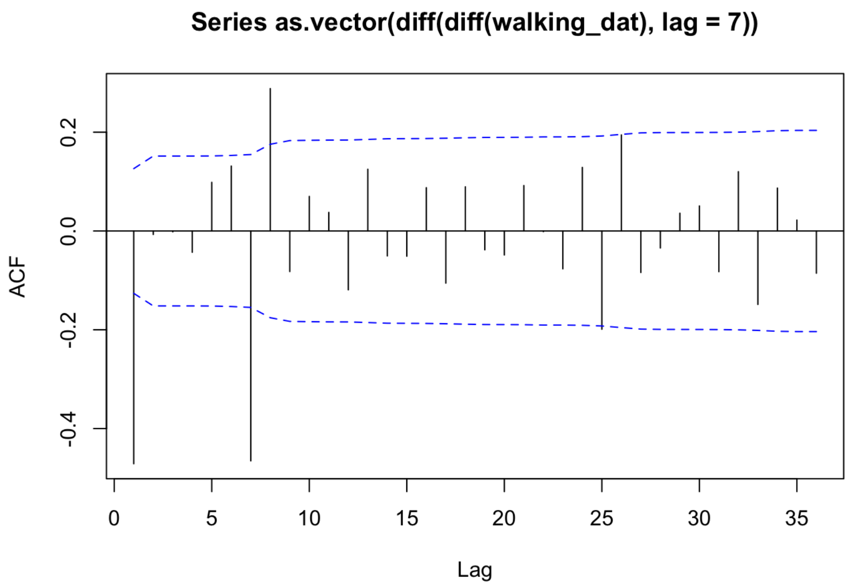


So now I'll take difference with regard to this lag of 7 and see what seasonality auto regression or moving average process can I follow.

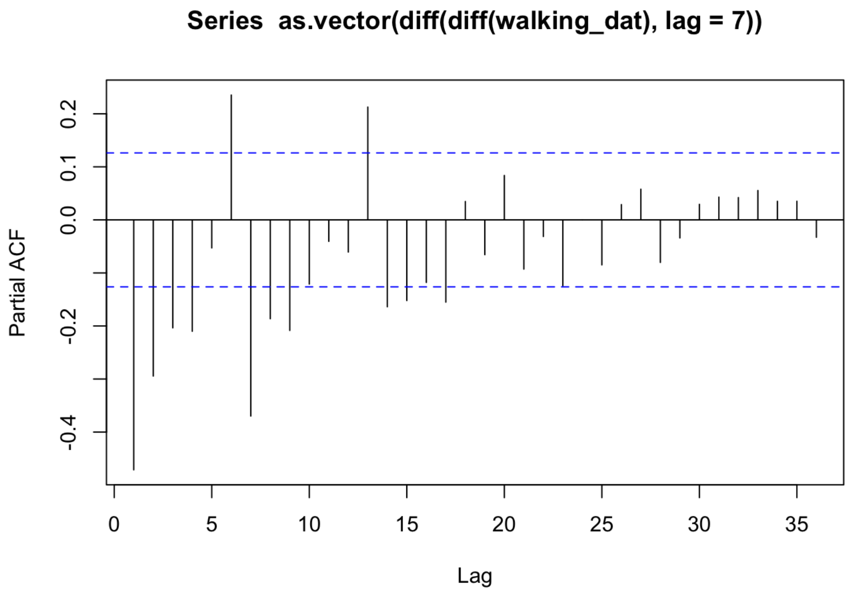
```
plot.ts(diff(diff(walking_dat),lag = 7),main="Time Series Plot of walking",xlab='Time',
     ylab='steps')
abline(h = 0)
```




```
acf(as.vector(diff(diff(walking_dat),lag=7)),lag.max=36,ci.type='ma')
```




```
pacf(as.vector(diff(diff(walking_dat),lag=7)),lag.max=36)
```




```
eacf(as.vector(diff(diff(walking_dat),lag=7)))
```

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x o o o o x x x o o o  o  o  o
## 1 x o o o o o x x x o o  o  o  o
## 2 x o o o o o x x o x o  o  o  o
## 3 x x o o o o x x o x o  o  o  o
## 4 x x x o o o x o x x o  o  o  o
## 5 x x x o o o x o o x o  o  o  o
## 6 x x x x x x x x x x o  o  o  o
## 7 x x o o o x x x x o o  o  o  x
```

Analysis

By applying same technique as analysis non-seasonal ARMA model, we can use acf and pacf to determine how many previous points should we take to predict the future value. In this example, the acf shows an significant value at first lag. Because the second is too trivial, we do not need to consider data point with more lags. And from pacf, we can guess that first, second, third and fourth lag may play important role in predition. From the eacf graph, we guess that the seasonality model might be seasonalarima(0,1,1), seasonalarima(0,1,2), seasonalarima(0,1,3).

Comparing, Training and Testing

The result given by following three model fitting is not significant. But the AIC(explained at the begining) gives us a good answer. The smallest AIC value is for model with seasonalarima(0,1,1). The arima() function helps me trained the model and give some appropriate parameter by using maximum likelihood estimation.

```r
arima(walking_dat[-c(212:250)], order = c(0, 1, 1), seasonal = list(order = c(0, 1, 1), period = 7))
```

```
##
## Call:
## arima(x = walking_dat[-c(212:250)], order = c(0, 1, 1), seasonal = list(order = c(0,
##     1, 1), period = 7))
##
## Coefficients:
##           ma1     sma1
##       -0.7767  -0.8965
## s.e.   0.0486   0.0500
##
## sigma^2 estimated as 1.524:  log likelihood = -337.16,  aic = 678.31
```

```r
arima(walking_dat[-c(212:250)], order = c(0, 1, 1), seasonal = list(order = c(0, 1, 2), period = 7))
```

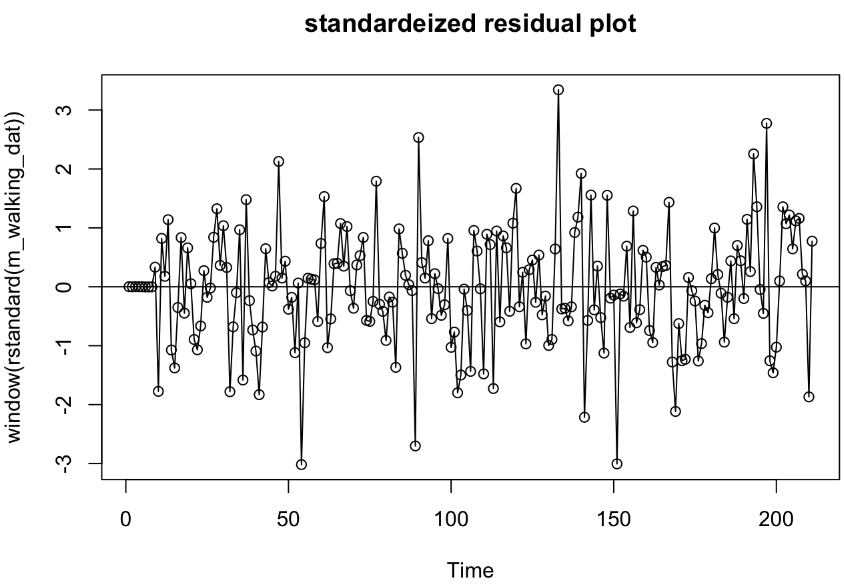
```
##
## Call:
## arima(x = walking_dat[-c(212:250)], order = c(0, 1, 1), seasonal = list(order = c(0,
##     1, 2), period = 7))
##
## Coefficients:
##           ma1     sma1    sma2
##       -0.7692  -0.9375  0.0614
## s.e.   0.0488   0.0724  0.0781
##
## sigma^2 estimated as 1.526:  log likelihood = -336.86,  aic = 679.71
```

```r
arima(walking_dat[-c(212:250)], order = c(0, 1, 1), seasonal = list(order = c(0, 1, 3), period = 7))
```

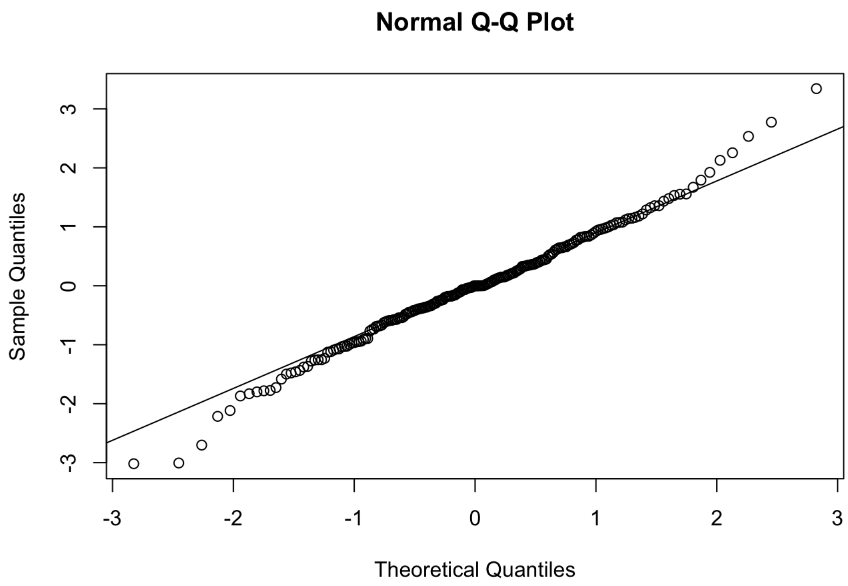
```
##
## Call:
## arima(x = walking_dat[-c(212:250)], order = c(0, 1, 1), seasonal = list(order = c(0,
##     1, 3), period = 7))
##
## Coefficients:
##           ma1     sma1    sma2     sma3
##       -0.7691  -0.9376  0.0620  -0.0005
## s.e.   0.0489   0.0767  0.1142   0.0808
##
## sigma^2 estimated as 1.526:  log likelihood = -336.86,  aic = 681.71
```

The next step to fit a model is to look at the normality of resudual, the method also introduced in my first post so here the plot shows that my residual is close to the line, which indicate the resuduals are normally distribute.

```r
m_walking_dat <- arima(walking_dat[-c(212:250)], order = c(0, 1, 1), seasonal = list(order = c(0, 1, 1), period =
7))
plot(window(rstandard(m_walking_dat)), type = "o")
title("standardeized residual plot")
abline(h = 0)
```

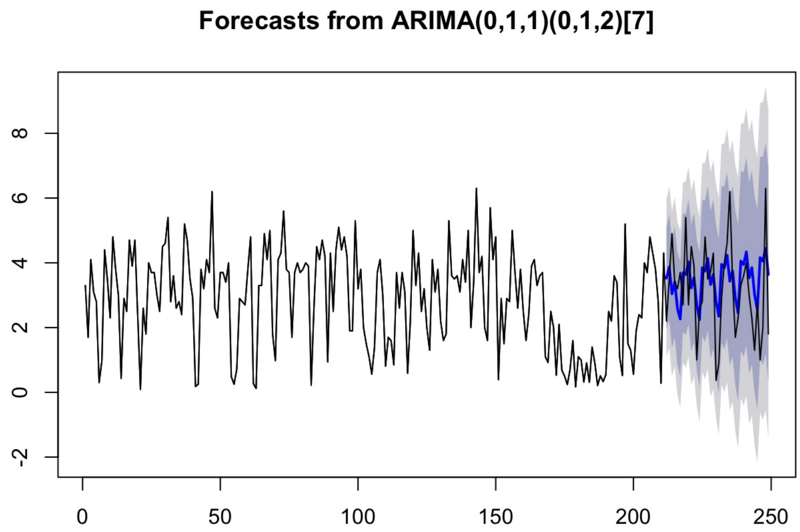



```r
qqnorm(window(rstandard(m_walking_dat))); qqline(window(rstandard(m_walking_dat)))
```



So now we need to test whether our training data works or not. I erased the last 38 data points, and applying my model to the previous 212 one. The graph plot shown in the result shows that all true value are laying within my prediction's confident interval. So therefore the model pass the test.

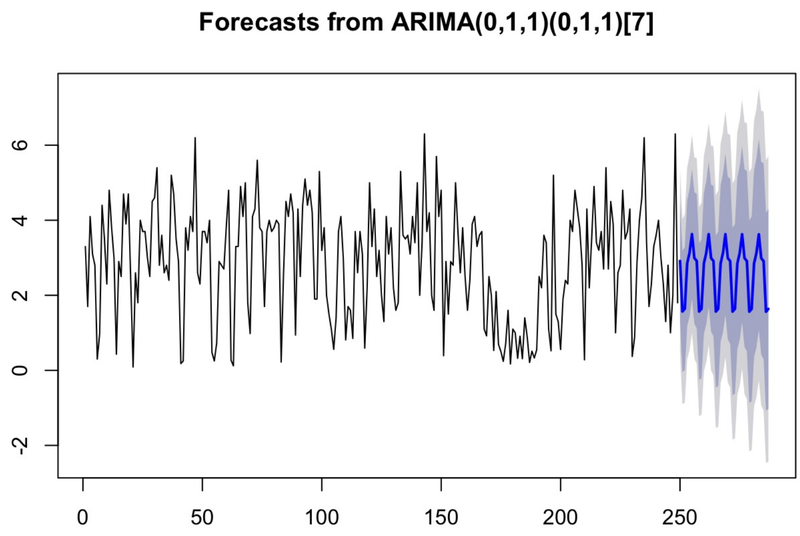
```
hold = window(ts(walking_dat), start = 212)
fit_no_holdout = stats::arima(ts(walking_dat[-c(212:250)]), order = c(0, 1, 1), seasonal = list(order = c(0, 1, 2)
, period = 7))
plot(forecast(fit_no_holdout, h = 38))
lines(ts(walking_dat))
```



Prediction

Using the model which had passed our test we got the following forecasting using forecast function from forecast package.

```
fit_holdout = stats::arima(ts(walking_dat), order = c(0, 1, 1), seasonal = list(order = c(0, 1, 1), period = 7))
plot(forecast(fit_holdout, h = 38))
```



The result looks clear and easy for me to predict what will my walking pattern be.

Conclution and takehome message

Not only applying to my walking steps. Time series prediction can be wildly applied. The underlying idea of time series analysis is to use our understanding of the present to predict the future, which is an extremely powerful tool applied in all the fields. Rather than diving into complicated time series models, this post focuses on some basic knowledge and techniques needed for the more advanced time series analysis. The major take-home message of this post is the fundamental idea of time series analysis: examine the events of the past, look for patterns, create models, and apply them to the future.

Reference:

-1.https://cran.r-project.org/web/views/TimeSeries.html
-2.https://cran.r-project.org/web/packages/lubridate/index.html
-3.https://cran.r-project.org/doc/Rnews/Rnews_2004-1.pdf
-4.[Jonathan_D._Cryer,_Kung-Sik_Chan]_Time_Series_Ana(BookFi)-edition2
-5.http://www.etsii.upm.es/ingor/estadistica/Carol/TSAtema4petten.pdf
-6.https://onlinecourses.science.psu.edu/stat510/node/33
-7.https://www.statmethods.net/advstats/timeseries.html