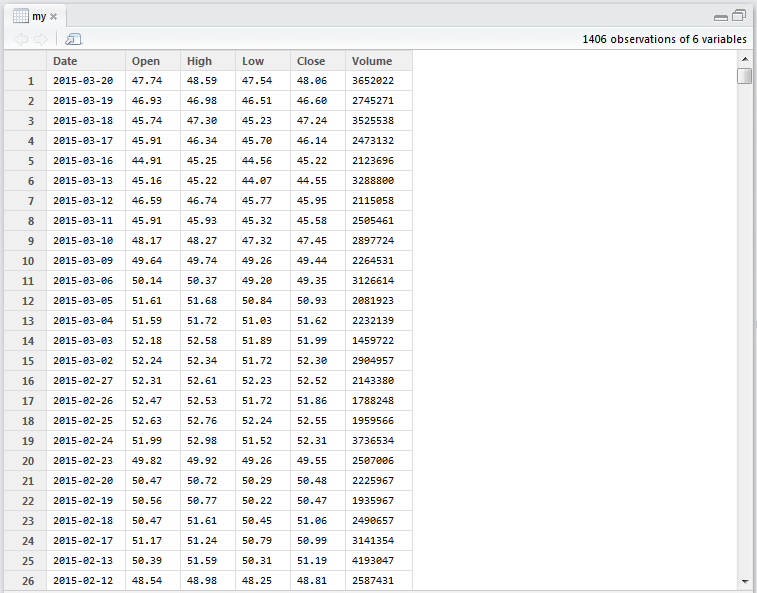
//Time series analysis

//DATA



Step 1://Reading the data into R

> my <- read.csv("C:/Users/user/Desktop/my.txt")

> View(my)

Step 2://Arrange the data into decreasing order

> mytable<-data.table(my)

> newdata <- myt[order(Date)]

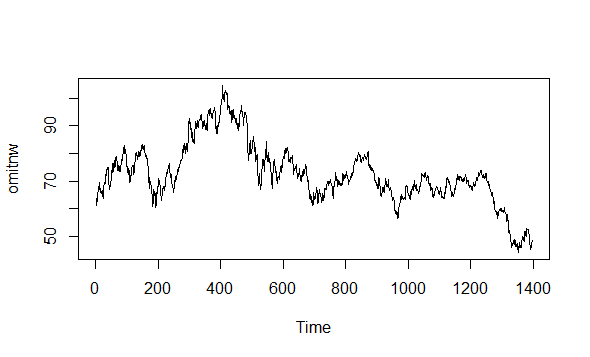
> newdata <- mytable[order(Date)]

> nw<-newdata$High

Step 3://Preprocessing the data

> omitnw<-na.omit(nw)

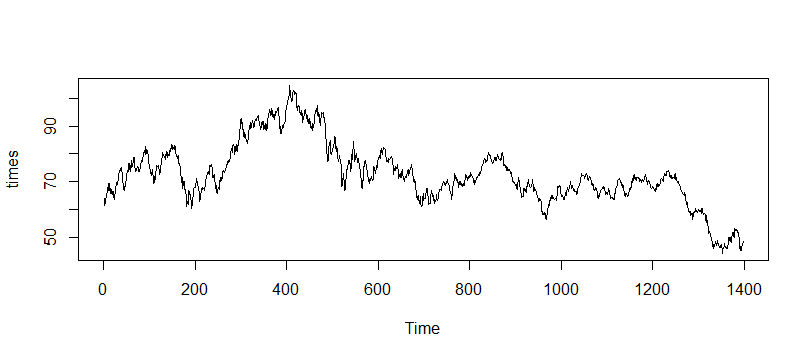
> plot.ts(omitnw)



Step 4 :Creating the time series object

> times<-ts(omitnw)

> plot.ts(times)



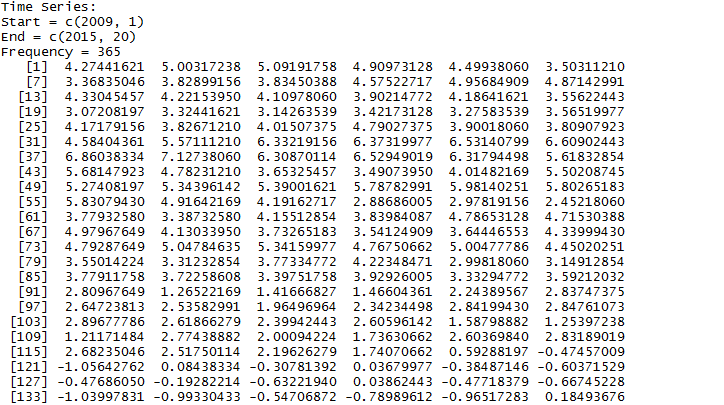
Step 5:// Decomposing the time series into 3 components:

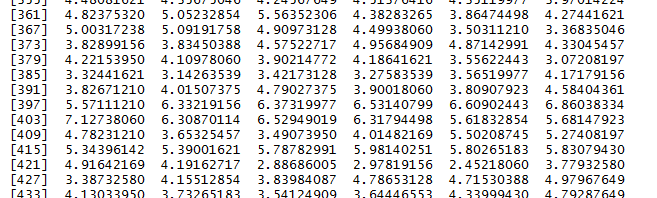
* Seasonal
* trend
* random

> times1<-ts(omitnw,frequency=365, start=c(2009,1),end=c(2015,20))

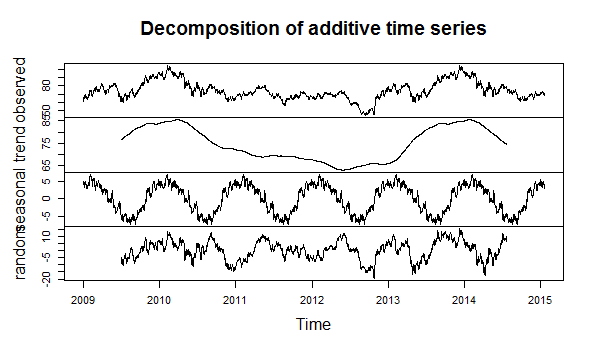
> dtimes1<-decompose(times1)

> dtimes1$seasonal





> plot(dtimes1)



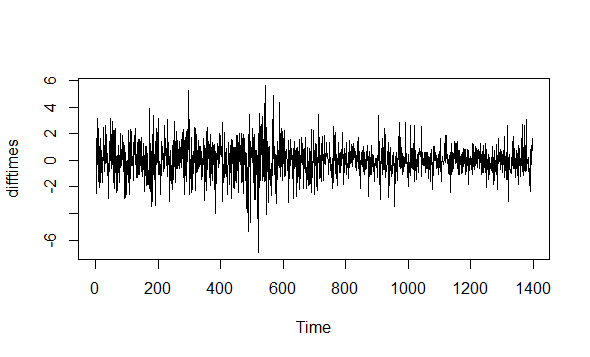
Step 6://Arima Model:

Step 6.1: Differencing the time series

//Applying DIFFERENCING

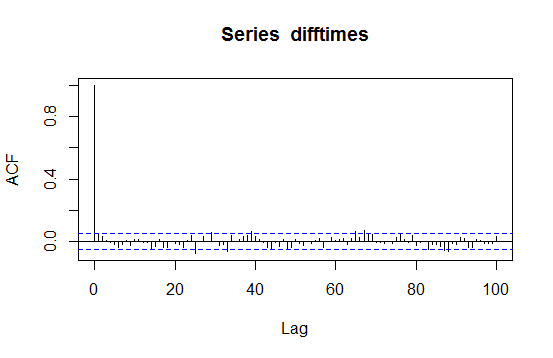
> difftimes <- diff(times, differences=1)

> plot.ts(difftimes)

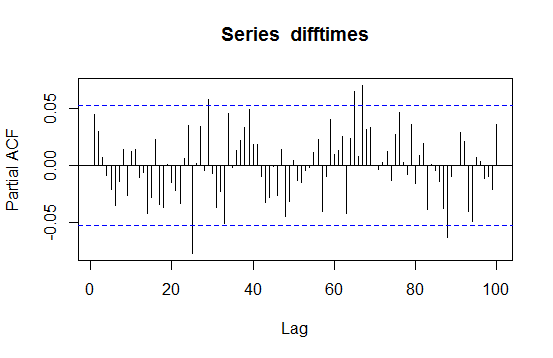


//Applying ACF

> acf(difftimes, lag.max=100)



> pacf(difftimes,lag.max=100)



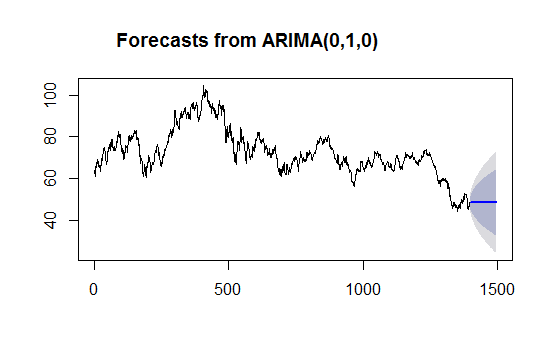
//model fitting

> finalfit<- arima(times, order=c(0,1,0))

//forcasting

> finalforcast <- forecast.Arima(finalfit, h=100)

> plot.forecast(finalforcast)



Step 7://Model Evaluation:

7.1 //Plotting forecast errors

plotForecastErrors <- function(forecasterrors)

+ {

+ # make a histogram of the forecast errors:

+ mybinsize <- IQR(forecasterrors)/4

+ mysd <- sd(forecasterrors)

+ mymin <- min(forecasterrors) - mysd\*5

+ mymax <- max(forecasterrors) + mysd\*3

+ # generate normally distributed data with mean 0 and standard deviation mysd

+ mynorm <- rnorm(10000, mean=0, sd=mysd)

+ mymin2 <- min(mynorm)

+ mymax2 <- max(mynorm)

+ if (mymin2 < mymin) { mymin <- mymin2 }

+ if (mymax2 > mymax) { mymax <- mymax2 }

+ # make a red histogram of the forecast errors, with the normally distributed data overlaid:

+ mybins <- seq(mymin, mymax, mybinsize)

+ hist(forecasterrors, col="red", freq=FALSE, breaks=mybins)

+ # freq=FALSE ensures the area under the histogram = 1

+ # generate normally distributed data with mean 0 and standard deviation mysd

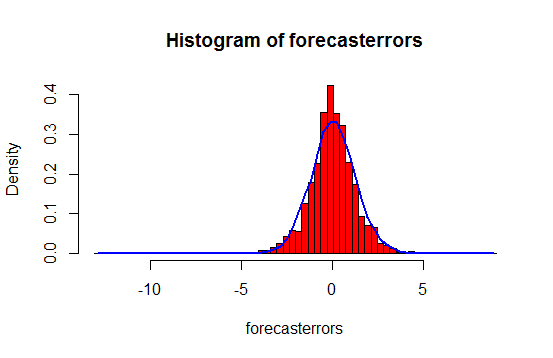
+ myhist <- hist(mynorm, plot=FALSE, breaks=mybins)

+ # plot the normal curve as a blue line on top of the histogram of forecast errors:

+ points(myhist$mids, myhist$density, type="l", col="blue", lwd=2)

+ }

> plotForecastErrors(finalforcast$residuals)



7.2//Superimposing the graphs of training and testing data

> `t1` <- read.csv("C:/Users/user/Desktop/t1.txt", header=FALSE)

> View(`t1`)

> tt1<-data.table(t1)

> ntt1 <- tt1[order(V1)]

> high<-ntt1$V3

> high<-na.omit(high)

> timestrain1<-ts(high,frequency=365, start=c(2009,1),end=c(2014,31))

> finalfit2<- arima(timestrain1, order=c(0,1,0))

> finalforcast2 <- forecast.Arima(finalfit2, h=300)

> plot(finalforcast2)

> par(new=TRUE)

> plot(times1)

