**Homework 3**

1. **Make an R quantlet on GH to produce hash code for the 2 sentences: “I learn a lot from this class when I am proper listening to the professor”, “I do not learn a lot from this class when I am absent and playing on my Iphone”. Compare the 2 hash sequences.**

**Solution:**

The R code is as follows:

rm(list=ls())

# call the library doing the hashes

library("digest")

# now do the hash code calculation

digest("I learn a lot from this class when I am proper listening to the professor", "sha256")

digest("I do not learn a lot from this class when I am absent and playing on my Iphone","sha256")

And the hash sequences for these two sentences are:

>digest("I learn a lot from this class when I am proper listening to the professor", "sha256")

[1] "c16700de5a5c1961e279135f2be7dcf9c187cb6b21ac8032308c715e1ce9964c"

> digest("I do not learn a lot from this class when I am absent and playing on my Iphone","sha256")

[1] "2533d529768409d1c09d50451d9125fdbaa6e5fd4efdeb45c04e3c68bcb3a63e"

We can see that It is totally different between these two hash sequences.

1. **Make 3-5 slides (in PPTX) on the DSA (Digital Signature Algorithms)**

**Solution:**

The slide called DSA is uploaded to Github.

1. **Make slide with R code where you create a JSON data set that you save and read again.**

**Solution:**

The slide called JSON is uploaded to Github.

1. **Download the CRIX data and make a plot of the time series, analyze its properties, i.e. fit ARMA, ARIMA etc. Is there a GARCH effect?**

**Solution:**

First I use the following R code to download the CRIX data and make a plot of the time series:

rm(list=ls())

library(rjson)

library(zoo)

library(ggplot2)

#download CRIX file

Json\_file = "http://crix.hu-berlin.de/data/crix.json"

Json\_data = fromJSON(file=Json\_file)

#transfer the JSON file to a standardized data frame format

crix\_data\_frame = as.data.frame(Json\_data)

num1 = seq(from=1,to=2349,by=2)

num2 = seq(from=2,to=2350,by=2)

date = as.Date(t(crix\_data\_frame[,num1]))

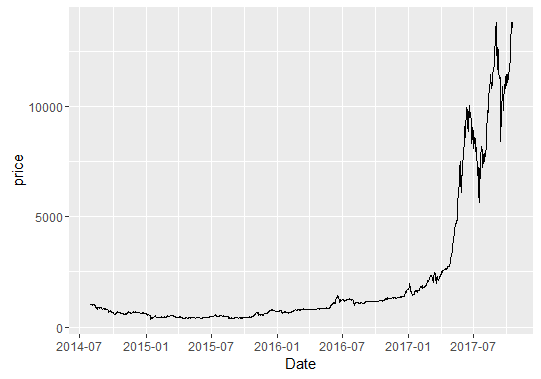
price = as.numeric(t(crix\_data\_frame[,num2]))

graph = data.frame(time=date,price=price)

#plot the time series

p = ggplot(graph,aes(x=time,y=price))

p + geom\_line(colour = 'black') + xlab('Date') + ylab('Price')



Then I used the function auto.arima in package forecast to fit the ARIMA model, and the code is as follows:

library(forecast)

zoo = zoo(price,order.by = date)

price\_ts = ts(zoo)

fit = auto.arima(price\_ts)

fit

The results shows that the sequence is fitted ARIMA(5,2,0)

Now, we have to justify if there is a Garch effect.

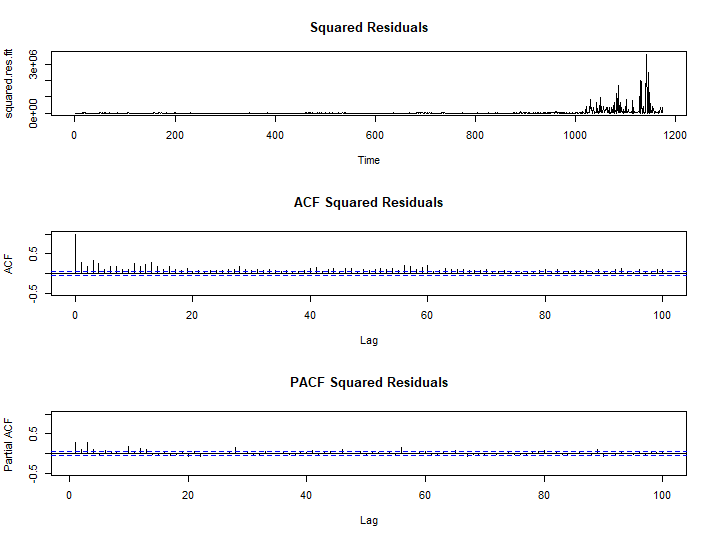
Firstly, check if residual plot displays any cluster of volatility. Next, observe the squared residual plot. If there are clusters of volatility, ARCH/GARCH should be used to model the volatility of the series to reflect more recent changes and fluctuations in the series. Finally, ACF & PACF of squared residuals will help confirm if the residuals (noise term) are not independent and can be predicted. If the residuals are strict white noise, they are independent with zero mean, normally distributed, and ACF & PACF of squared residuals displays no significant lags.

Followings are the plots of squared residuals:

• Squared residuals plot shows cluster of volatility at some points in time

• ACF seems to die down

• PACF cuts off after lag 13 even though some remaining lags are significant



So it is convinced that there is an ARCH effect.