HW 4

Biying Wan

15620161152279

```
#download data
#price
library("rjson")
#json_file="D:/研二/研二上/大数据与互联网金融/HW4/crix.json"
json_file="http://crix.hu-berlin.de/data/crix.json"
json_data <- fromJSON(paste(readLines(json_file), collapse=""))
json_df <- as.data.frame(c(json_data[[1]][1],json_data[[1]][2]))
for (i in 2:length(json_data)){
  json_df <- rbind(json_df,as.data.frame(c(json_data[[i]][1],json_data[[i]][2])))
}
json_df$date <- as.POSIXct(json_df$date)

# return
x<-json_df[,2]
return<-log(x[2:nrow(json_df)])-log(x[1:nrow(json_df)-1])
return<-c(NA,return)
json_return<-as.data.frame(cbind(json_df,return))
json_return<-json_return[-1,-2]
```

```
#save data set
save(json_df,file="D:/研二/研二上/大数据与互联网金融/HW4/crix.RData")
save(json_return,file=" D:/研二/研二上/大数据与互联网金融/HW4/ return.RData")

#clear
rm(list = ls(all = TRUE)) #rm(list = ls())
graphics.off()

# install and load packages
libraries = c("zoo", "tseries", "xts")
lapply(libraries, function(x) if (!(x %in% installed.packages())) {
  install.packages(x)
})
lapply(libraries, library, quietly = TRUE, character.only = TRUE)

#load data set
load(file = " D:/研二/研二上/大数据与互联网金融/HW4/ crix.RData")
load(file = " D:/研二/研二上/大数据与互联网金融/HW4/ return.RData")
```

```
#fig 3 in Econometrics of CRIX, plot of crix
#plot(json_df,xlab=NA,ylab=NA,type="l",col="red")
library(ggplot2)
library(scales)
ggplot(json_df)+
  geom_line(aes(x=date,y=price))+
  scale_x_datetime(breaks=date_breaks(“4
month"),labels=date_format("%Y/%m"))
```

Crix Index

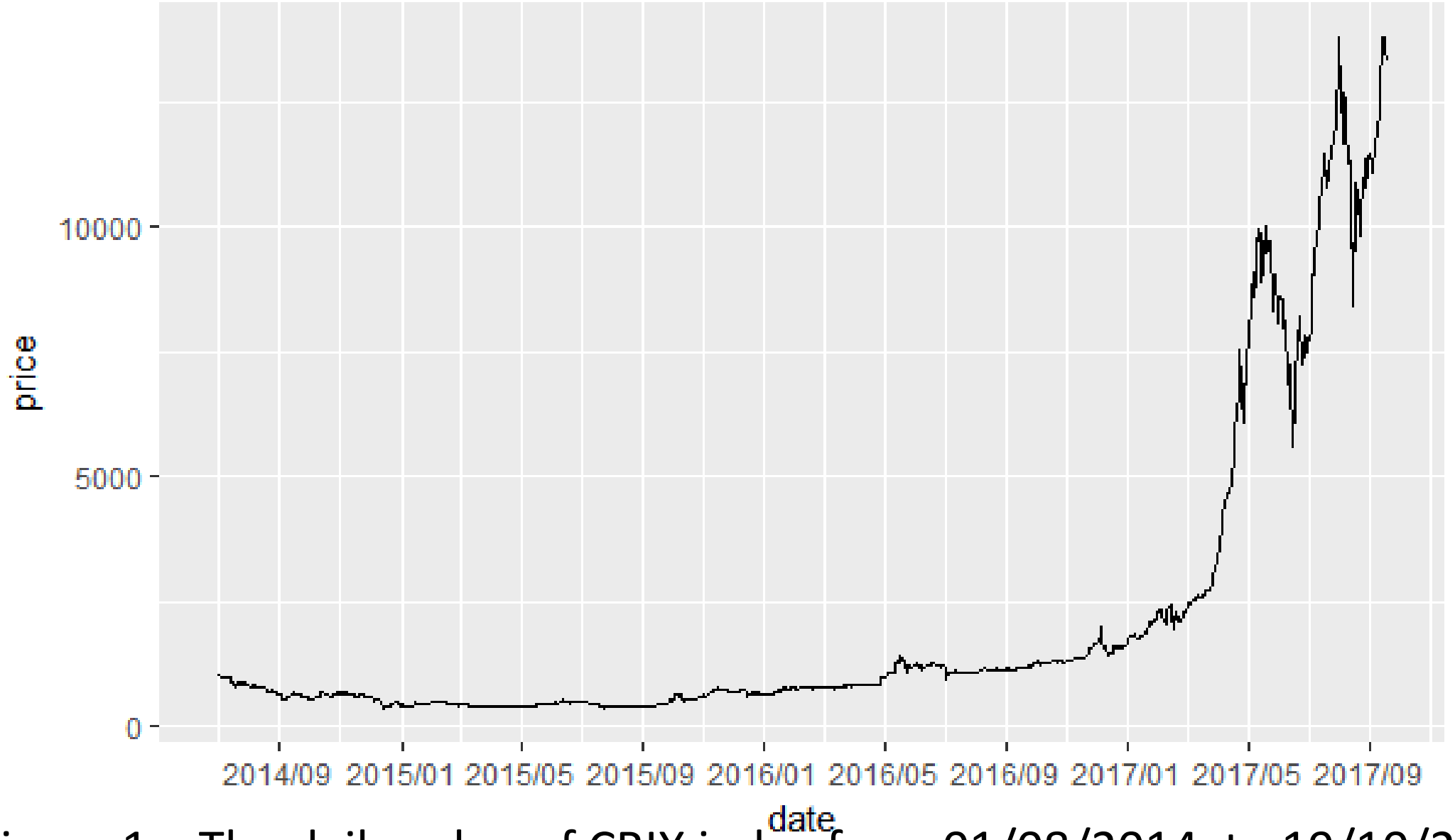

Figure 1: The daily value of CRIX index from 01/08/2014 to 19/10/2017

```
#fig 4 in Econometrics of CRIX ,plot of return
#plot(json_return,xlab=NA,ylab=NA,type="l")
ggplot(json_return)+
  geom_line(aes(x=date,y=return))+
  scale_x_datetime(breaks=date_breaks("4
month"),labels=date_format("%Y/%m"))
```

Data Description

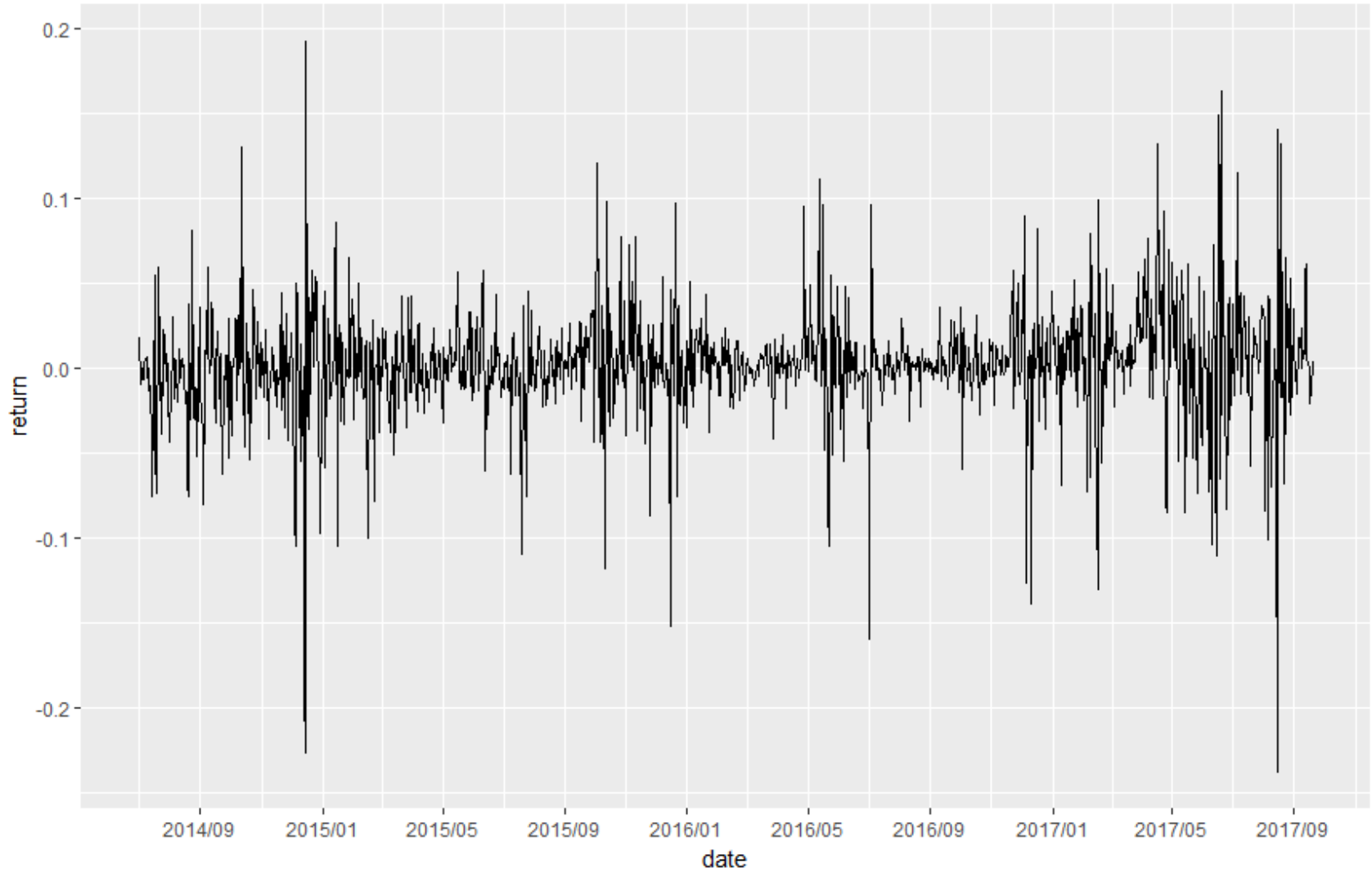

Figure 2: The log returns of CRIX index from 01/08/2014 to 19/10/2017

```
#fig 5 in Econometrics of CRIX,histogram of returns,qq-plot
return<-json_return[,2]
par(mfrow = c(1, 2))
# histogram of returns
hist(return, col = "grey", breaks = 20, freq = FALSE, ylim = c(0, 25), xlab =
"return")
lines(density(return), lwd = 2)
mu = mean(return)
sigma = sd(return)
x = seq(-4, 4, length = 100)
curve(dnorm(x, mu, sigma), add = TRUE, col = "darkblue", lwd = 2)
# qq-plot
qqnorm(return)
qqline(return, col = "blue", lwd = 3)
```

Distributional Property

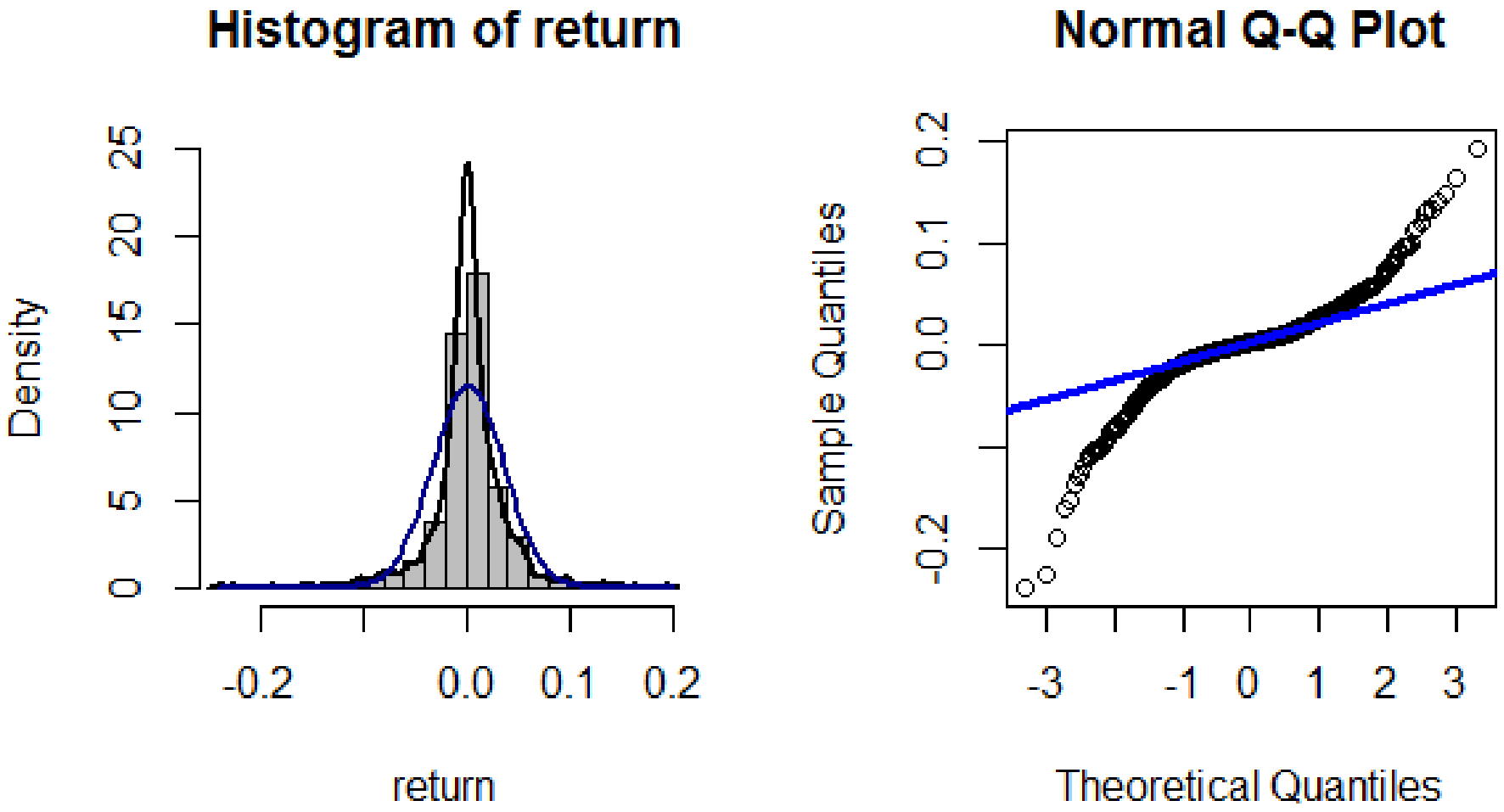

Figure 3: Histogram and QQ plot of CRIX returns from 01/08/2014 to 19/10/2017

```
#fig 6 in Econometrics of CRIX
par(mfrow = c(1, 2))
# acf plot
autocorr = acf(return, lag.max = 20, ylab = "Sample
Autocorrelation", main = NA, lwd = 2, ylim = c(-0.3, 1))
# pacf plot
autopcorr = pacf(return, lag.max = 20, ylab = "Sample Partial
Autocorrelation", main = NA, ylim = c(-0.3, 0.3), lwd = 2)
```

Lag Order

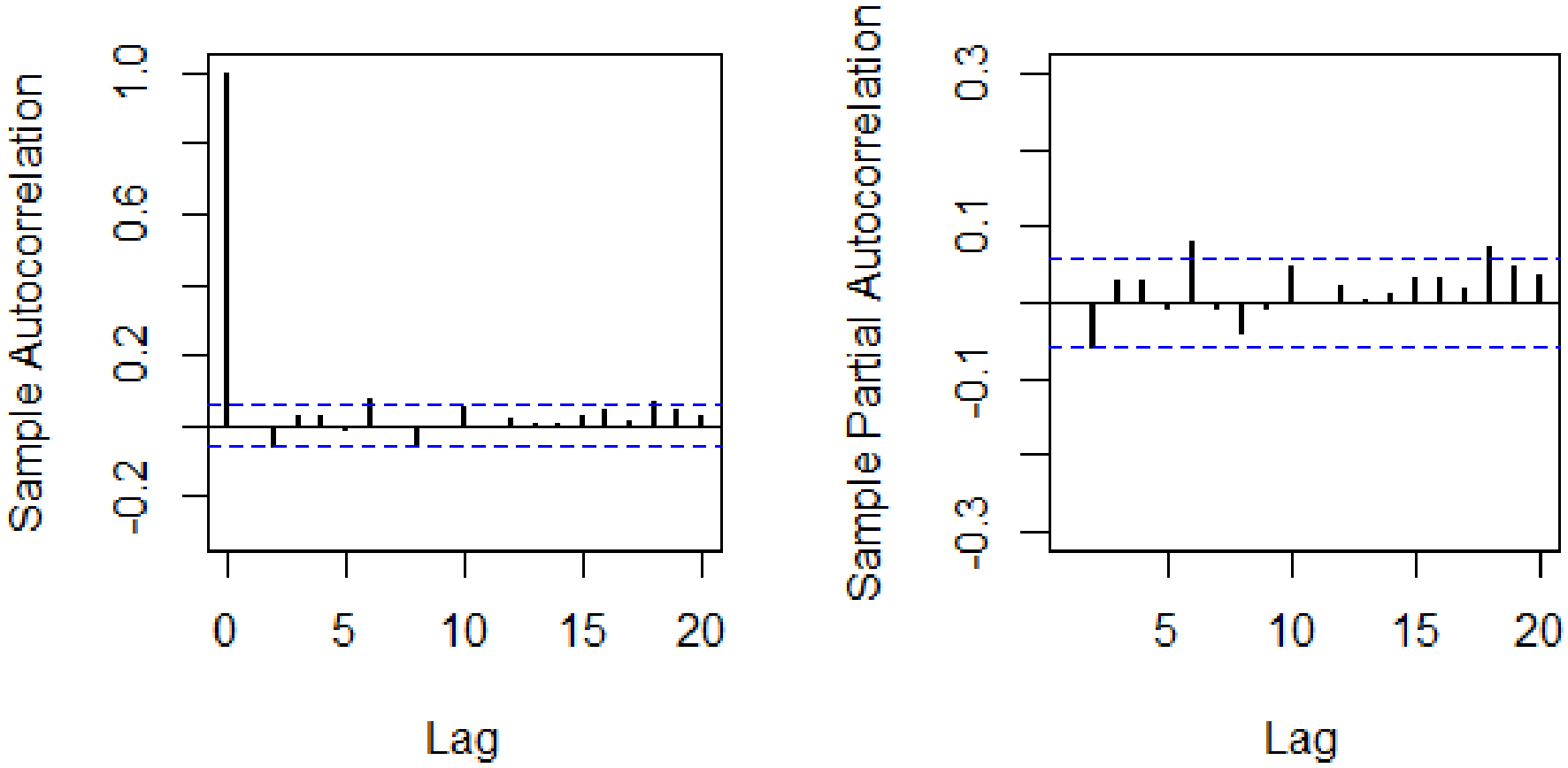

Figure 4: The sample ACF and PACF of CRIX returns from 01/08/2014 to 19/10/2017

```
#fig 7 in Econometrics of CRIX,  arima202 predict
graphics.off()
fit202 = arima(return, order = c(2, 0, 2))
crpre = predict(fit202, n.ahead = 30)
dates = seq(as.Date("01/08/2014", format = "%d/%m/%Y"), by =
"days", length = length(return))
plot(return, type = "l", xlim = c(0, 1206), ylab = "return", xlab =
"days", lwd = 1.5)
lines(crpre$pred, col = "red", lwd = 3)
lines(crpre$pred + 2 * crpre$se, col = "red", lty = 3, lwd = 3)
lines(crpre$pred - 2 * crpre$se, col = "red", lty = 3, lwd = 3)
```

ARIMA Model Forecast

With ARIMA(2,0,2) model, we predict CRIX returns for next 30 days.

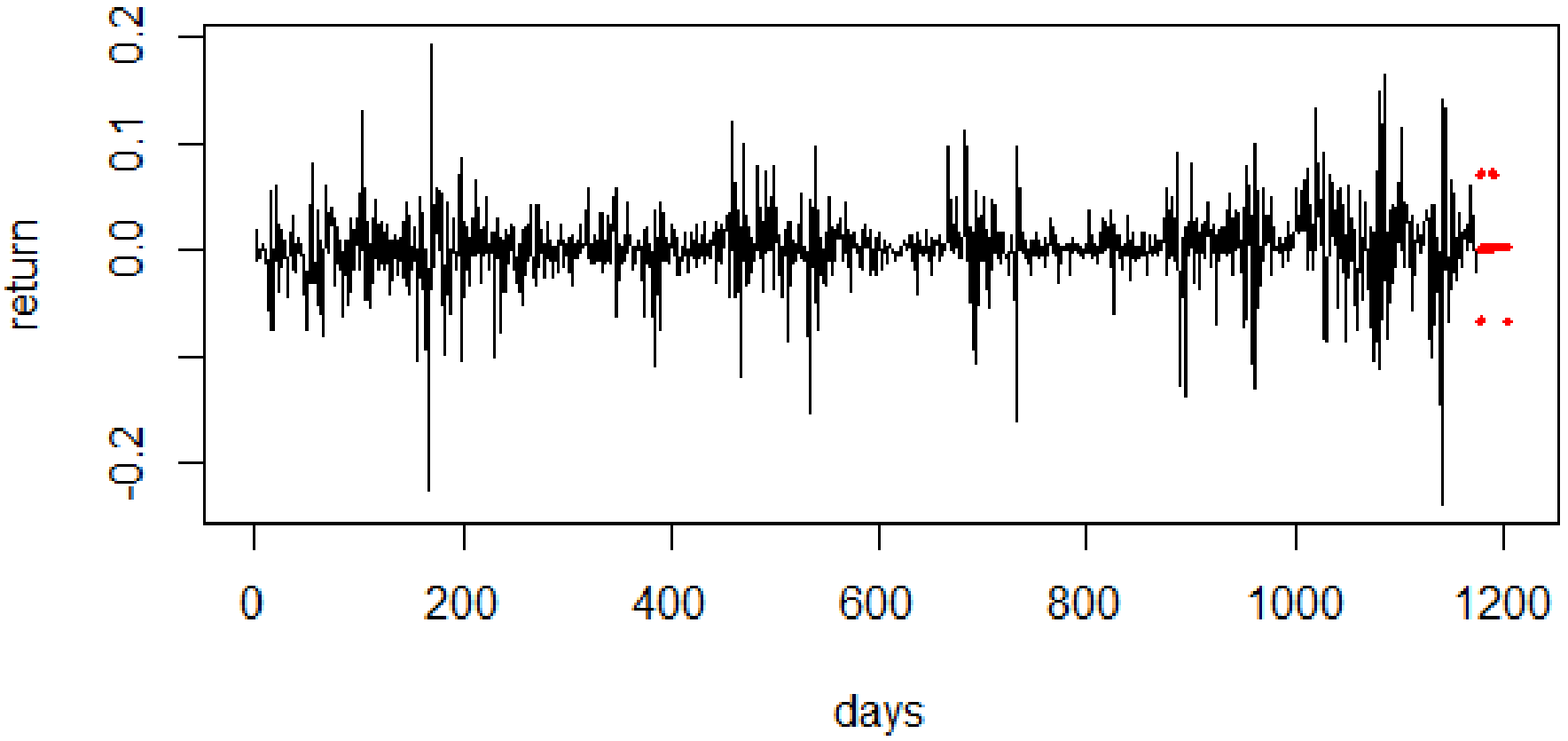

Figure 5: CRIX returns and predicted values

```
# fig 8 in Econometrics of CRIX, Volatility cluster
graphics.off()
date=json_return$date
Volatility= fit202$residuals^2
tsres202 = data.frame(date,Volatility)
#plot(tsres202, type = "l",xlab="date", ylab = "Volatility")
ggplot(tsres202)+
  geom_line(aes(x=date,y=Volatility))+
  scale_x_datetime(breaks=date_breaks("4
month"),labels=date_format("%Y/%m"))
```

Volatility Cluster

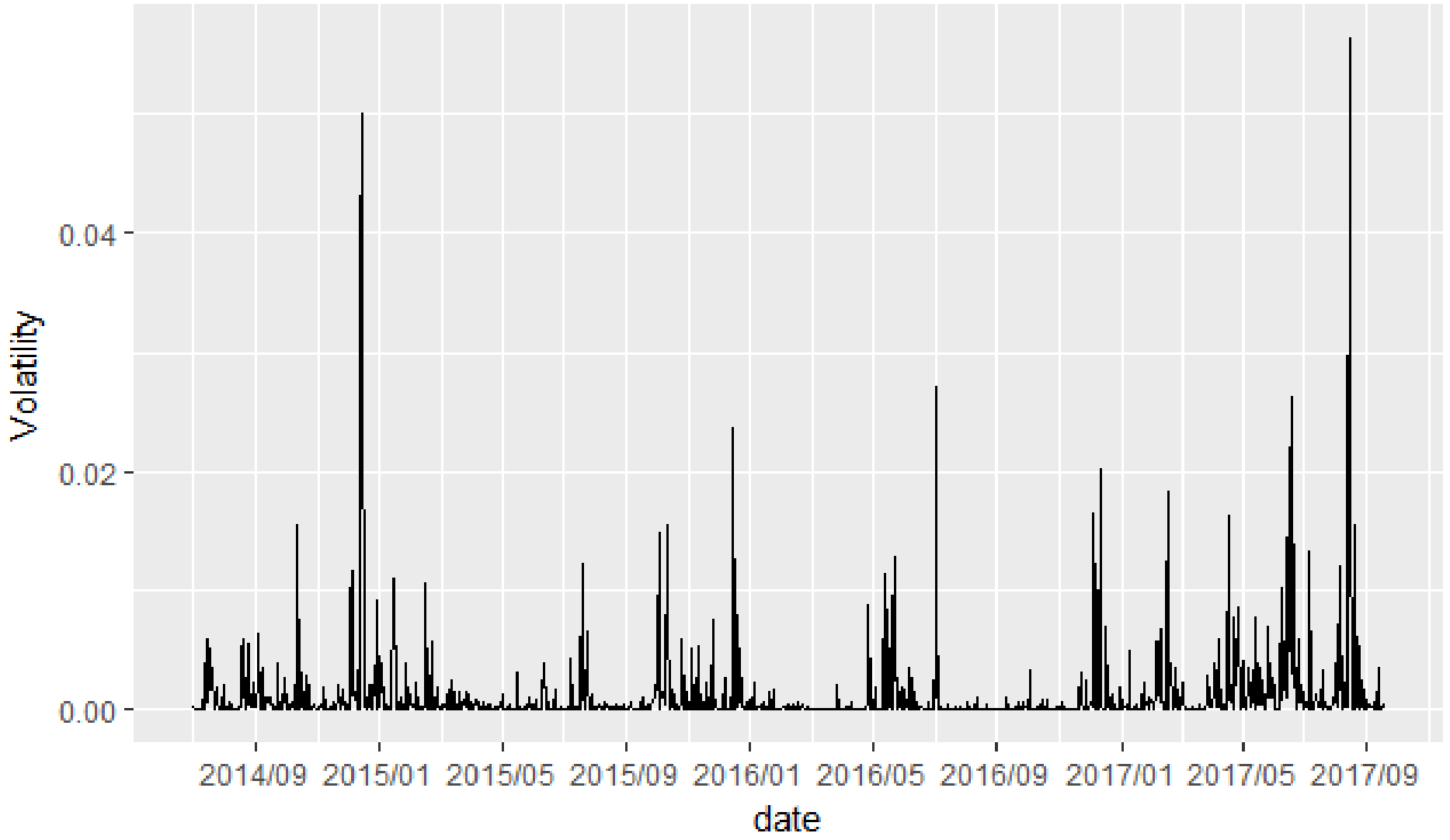

Figure 6: The squared ARIMA(2,0,2) residuals of CRIX returns.