Building GARCH Model to Estimate Volatilites

Overview

This is an example of using GARCH model to estimate volatilities of S&P500 index. I divide the analysis into following steps: 1. Data Prearation and Cleaning 2. Data Visualization 3. Variable Testing 4. GARCH Modeling

Data Preparation and Cleaning

```
library(tidyverse)
library(tseries)
library(TSA)
library(forecast)
df = read.csv('sp500.csv', header = TRUE)
```

```
head(df)
```

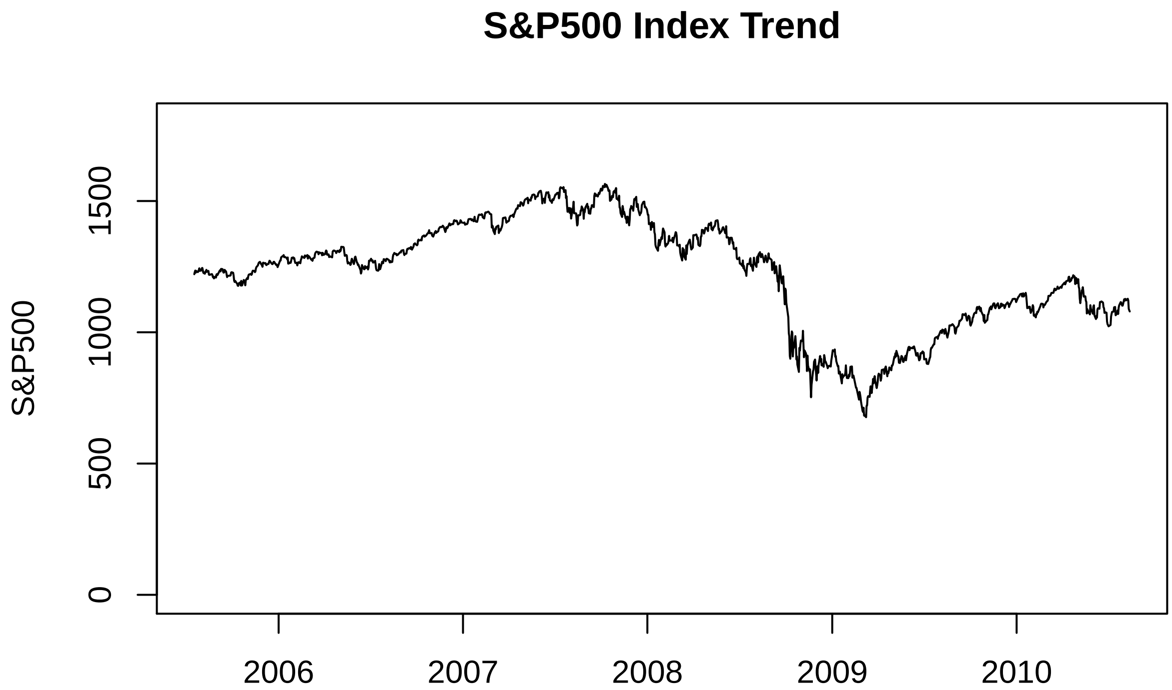
```
##        date   price
## 1 7/18/2005 1221.13
## 2 7/19/2005 1229.35
## 3 7/20/2005 1235.20
## 4 7/21/2005 1227.04
## 5 7/22/2005 1233.68
## 6 7/25/2005 1229.03
```

```
df$date = as.Date(df$date, format = "%m/%d/%Y")
```

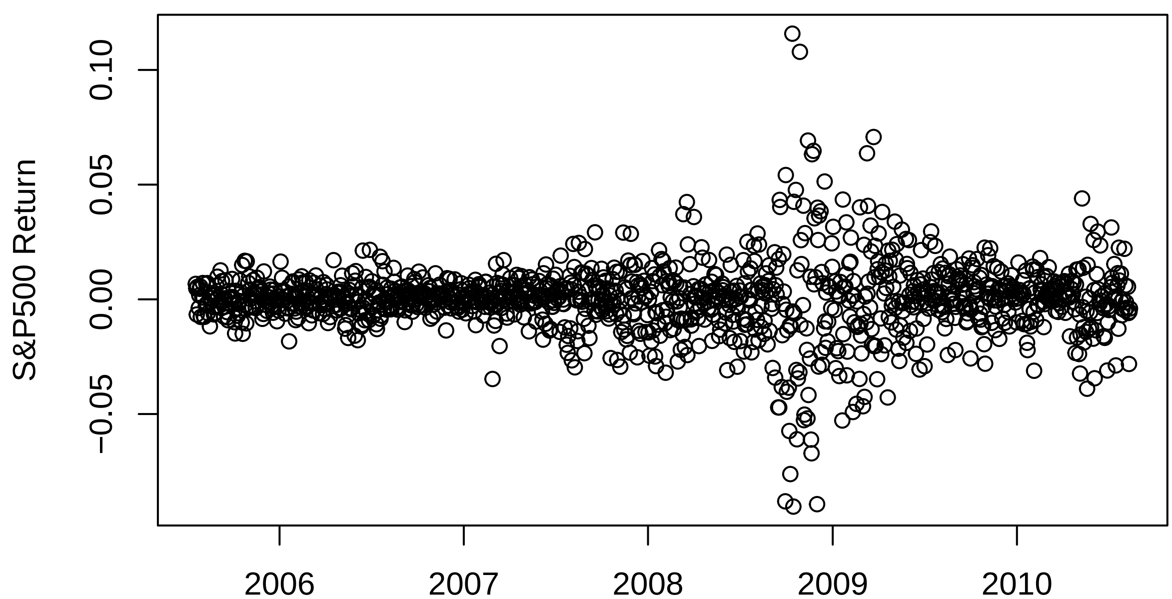
```
n = nrow(df)
ret = df$price[-1]/df$price[-n] -1
col3 = c(NA, ret)
df2 = data.frame(df, ret = col3)
```

Data Visualization

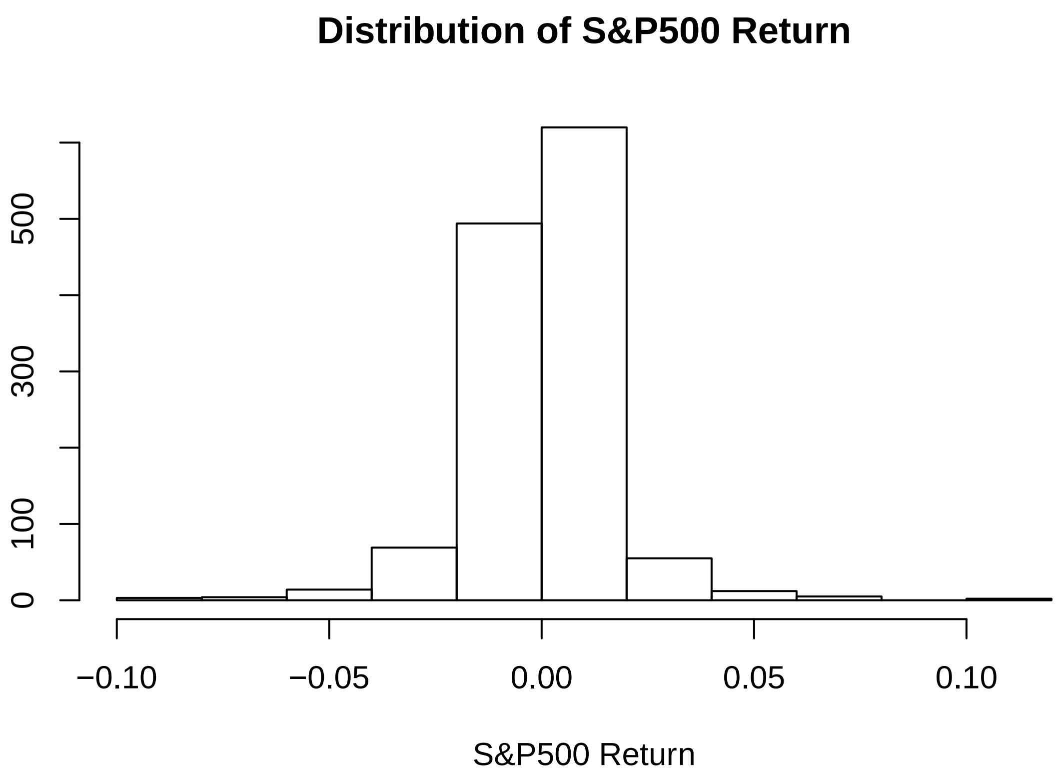
```
plot(price~date, df, type = 'l',ylim = c(0,1800), ylab = "S&P500", xlab = "", main = "S&P500 Index Trend
```




```r
plot(ret~date, df2, type = 'p', ylab = "S&P500 Return", xlab = "")
```

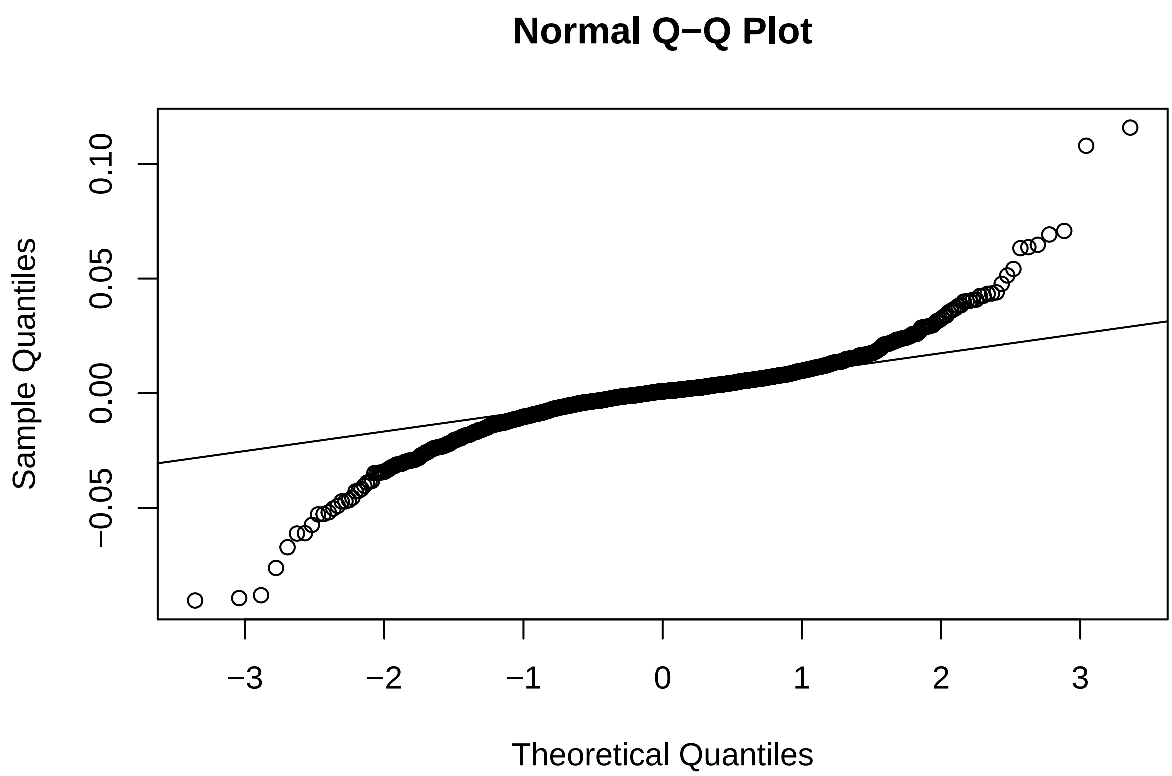



```r
hist(df2$ret, xlab = "S&P500 Return",main = "Distribution of S&P500 Return",ylab = "")
```



Variable Testing

```r
qqnorm(df2$ret)
qqline(df2$ret)
```



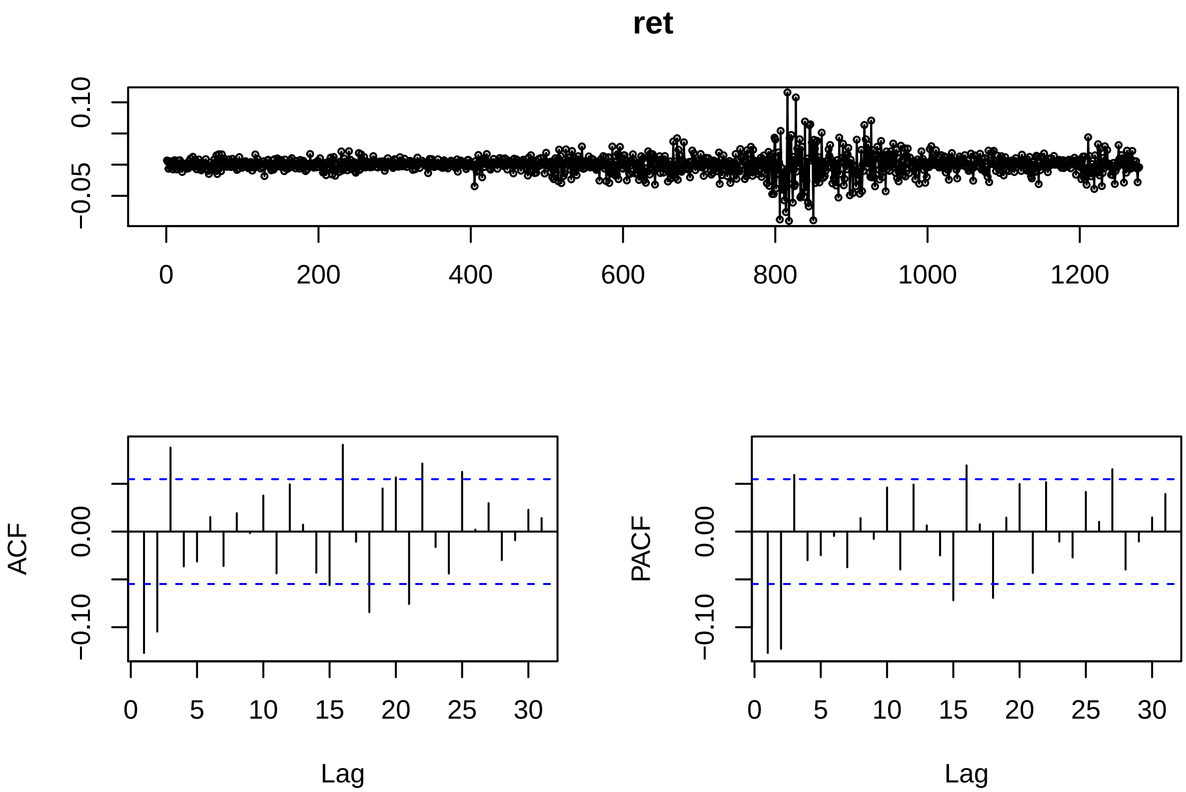
Shapiro-Wilk Test of Normality The result shows that return of S&P500 does not follow normal distribution.

```r
shapiro.test(df2$ret)
```

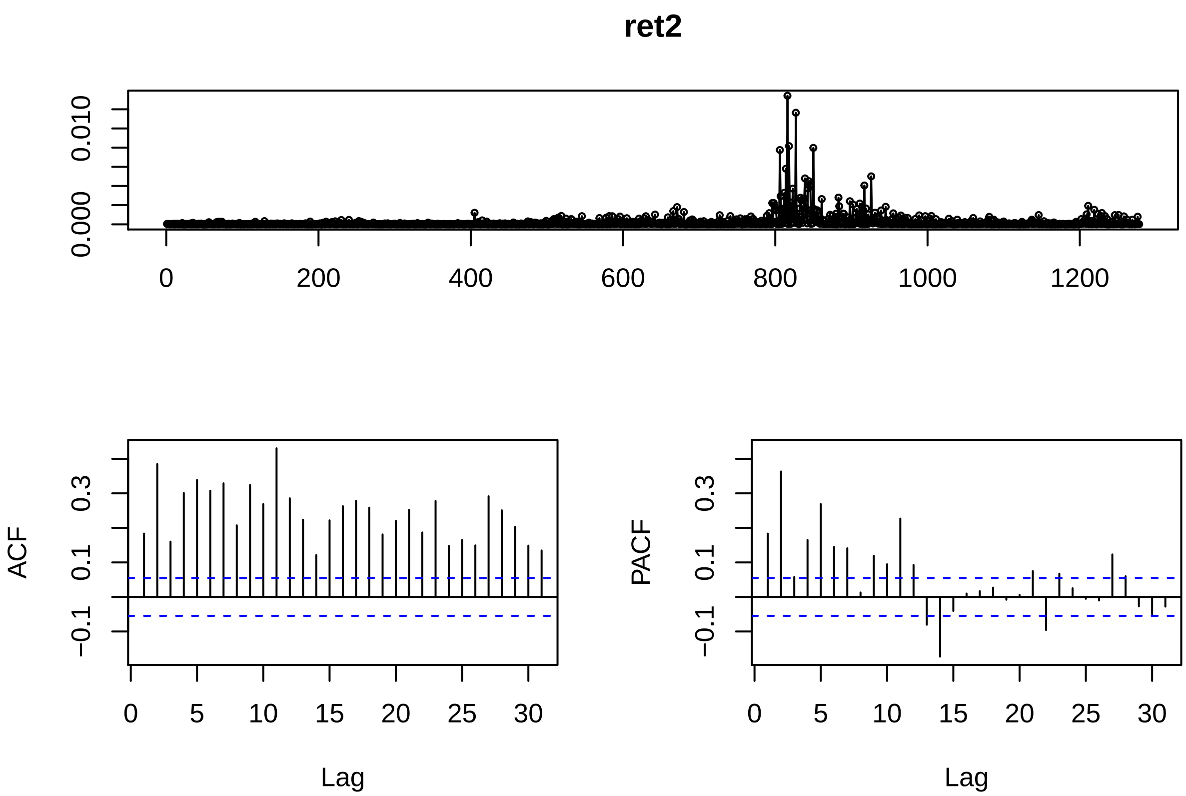
```
##
##  Shapiro-Wilk normality test
##
## data:  df2$ret
## W = 0.87371, p-value < 2.2e-16
```

Use acf and pacf to identify time-correlation of daily returns/squared returns.

```r
tsdisplay(ret)
```




```r
ret2 = (df2$ret[-1])^2
tsdisplay(ret2)
```



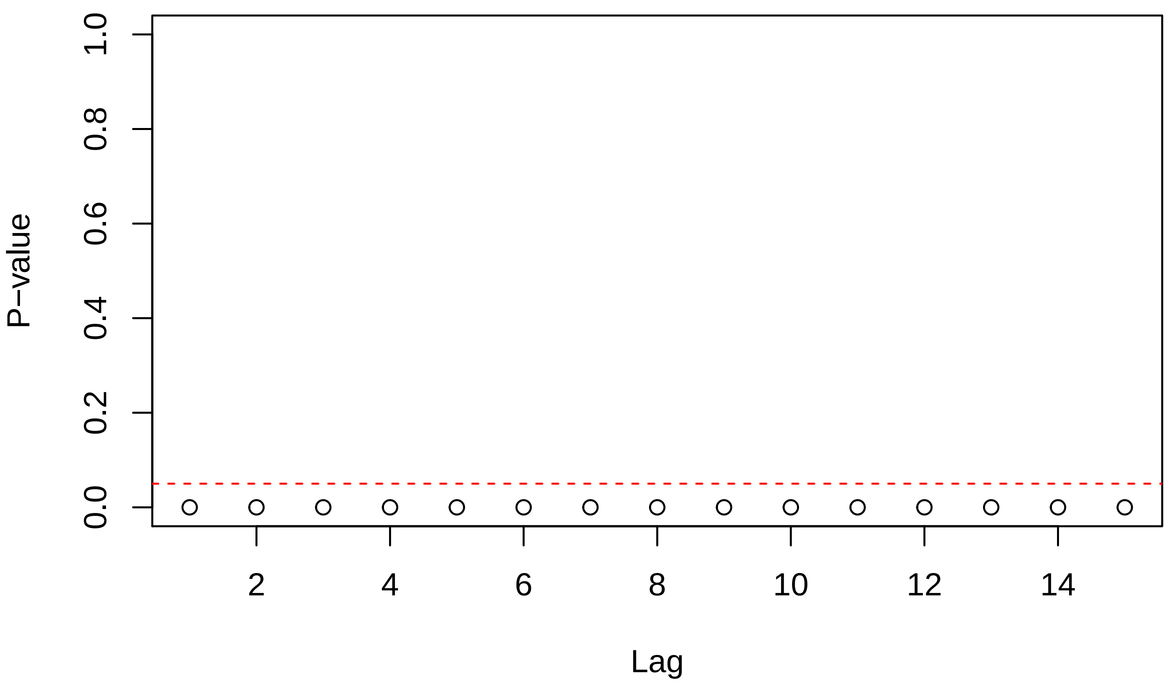
Use t-test on whether mean return is equal to zero. Since p-value is larger than 0.05, we fail to reject null hypothesis that mean return is equal to zero.

```
t.test(ret,mu=0)
```

```
##
##  One Sample t-test
##
## data:  ret
## t = 0.055256, df = 1277, p-value = 0.9559
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.0008283061  0.0008763181
## sample estimates:
##    mean of x
## 2.400598e-05
```

Use McLeod-Li Test on whether daily volatility is constant. p-value for different lags are all less than 0.05, therefore we have confidence to reject null hypothesis that daily volatility is constant.

```
McLeod.Li.test(y = ret, plot = T, gof.lag=15)
```



GARCH Modeling

```
# build garch model
ml = garch(ret)
```

```
# Extract parameters

w = coef(ml)[1]    #gamma
a = coef(ml)[2]    #alpha
b = coef(ml)[3]    #beta
varlong = w/(1-a-b) # long term variance
```

```r
summary(ml)
```

```
##
## Call:
## garch(x = ret)
##
## Model:
## GARCH(1,1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.4131 -0.5129  0.0761  0.5601  2.9629
##
## Coefficient(s):
##     Estimate  Std. Error  t value Pr(>|t|)
## a0 1.552e-06   3.201e-07    4.849 1.24e-06 ***
## a1 9.287e-02   1.262e-02    7.361 1.82e-13 ***
## b1 8.998e-01   1.282e-02   70.206  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Diagnostic Tests:
##  Jarque Bera Test
##
## data:  Residuals
## X-squared = 229.56, df = 2, p-value < 2.2e-16
##
##
##  Box-Ljung test
##
## data:  Squared.Residuals
## X-squared = 5.3351, df = 1, p-value = 0.0209
```

```r
# display the atrributes of ml
str(ml)
```

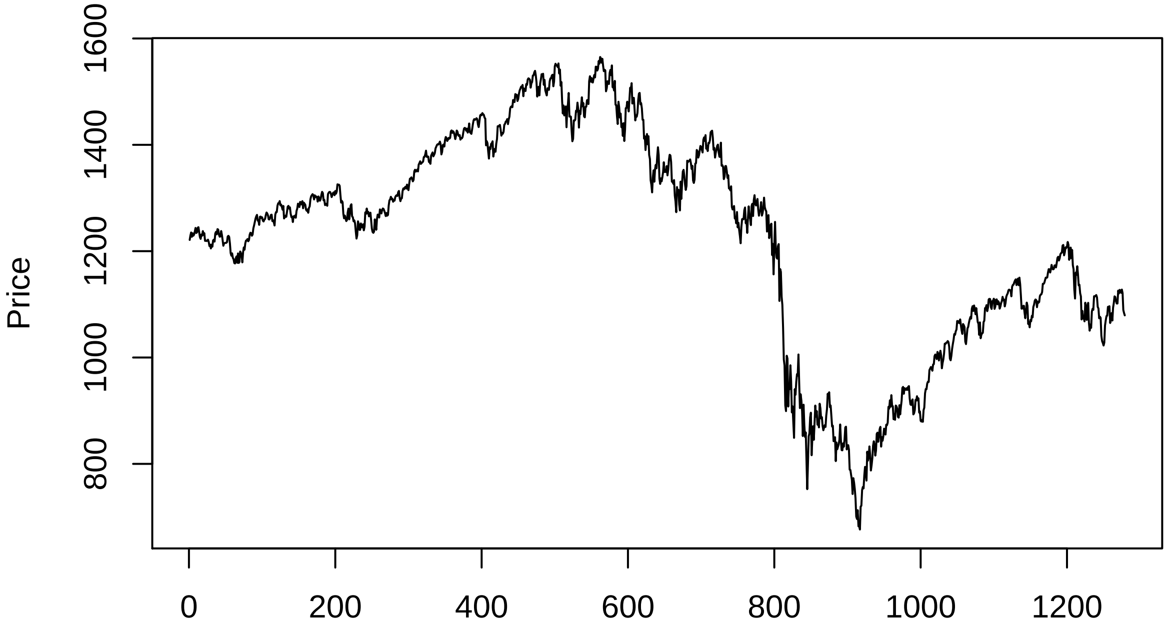
```
## List of 10
##  $ order        : Named num [1:2] 1 1
##   ..- attr(*, "names")= chr [1:2] "p" "q"
##  $ coef         : Named num [1:3] 1.55e-06 9.29e-02 9.00e-01
##   ..- attr(*, "names")= chr [1:3] "a0" "a1" "b1"
##  $ n.likeli     : num -5108
##  $ n.used       : int 1278
##  $ residuals    : num [1:1278] NA 0.34 -0.492 0.418 -0.303 ...
##  $ fitted.values: num [1:1278, 1:2] NA 0.014 0.0134 0.0129 0.0125 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr [1:2] "sigt" "-sigt"
##  $ series       : chr "ret"
##  $ frequency    : num 1
##  $ call         : language garch(x = ret)
##  $ vcov         : num [1:3, 1:3] 1.02e-13 2.72e-09 -3.52e-09 2.72e-09 1.59e-04 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:3] "a0" "a1" "b1"
##   .. ..$ : chr [1:3] "a0" "a1" "b1"
```

```
##  - attr(*, "class")= chr "garch"
```

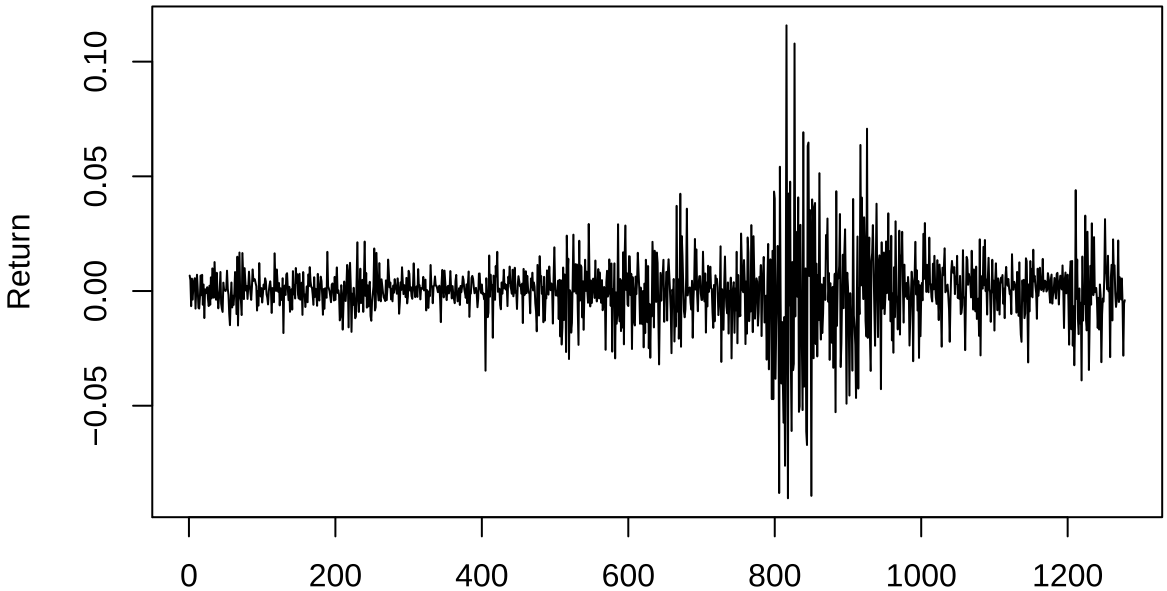
```r
# fitted daily volatilities
vfit = ml$fitted.values[-1,1]
head(vfit)
```

```
## [1] 0.01399930 0.01341628 0.01294467 0.01245169 0.01193228 0.01139924
```

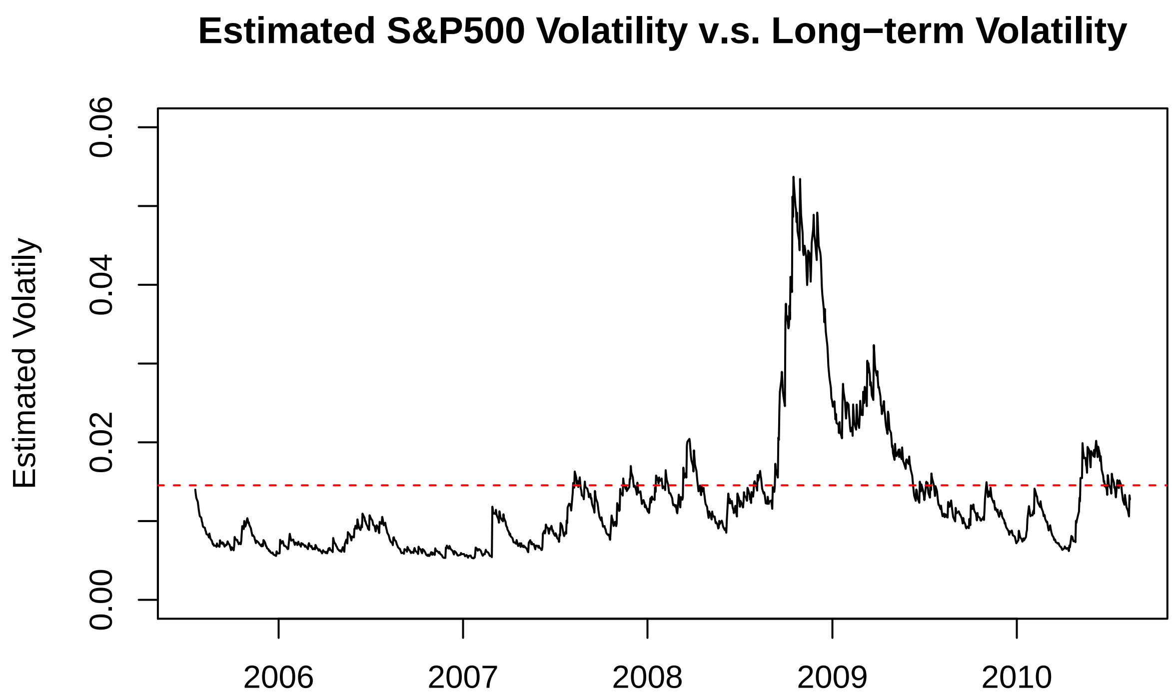
```r
#par(mfrow = c(2,1))
plot(df2$price, type="l",ylab = "Price", xlab = "")
```




```r
plot(ret, type = "l", ylab = "Return", xlab = "")
```

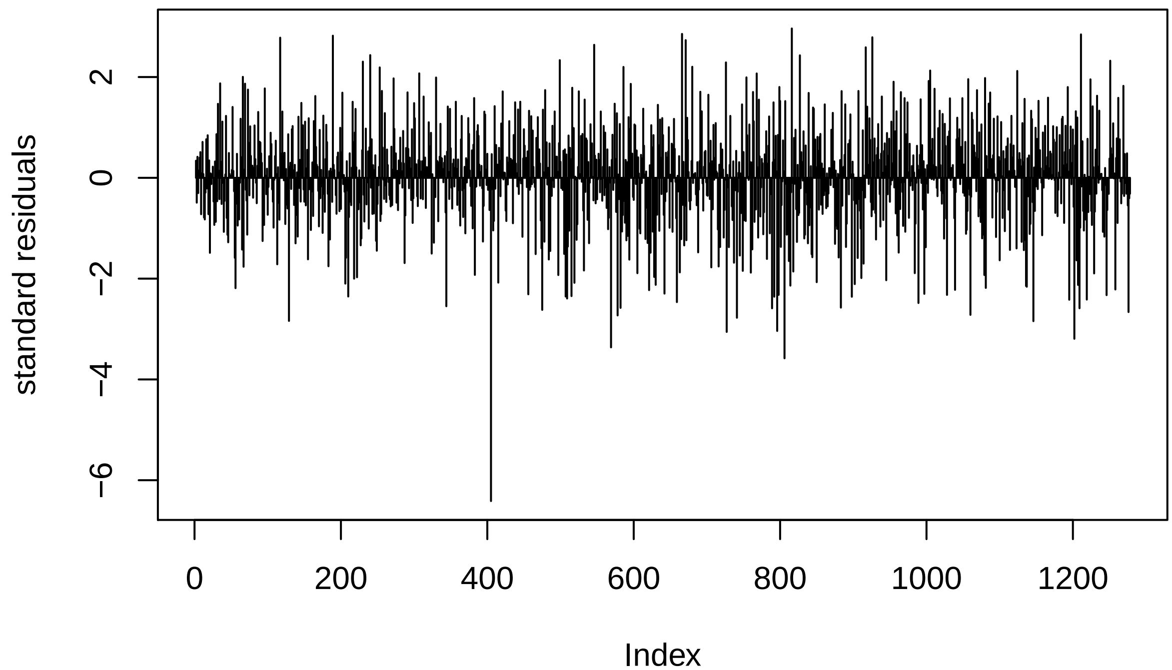



```r
dates = df$date[-c(1,2)]
plot(vfit~dates, type = 'l', ylim = c(0,0.06), xlab="",ylab = "Estimated Volatily",main = "Estimated S&
abline(h = sqrt(varlong), lty = 2, col = 'red')
```

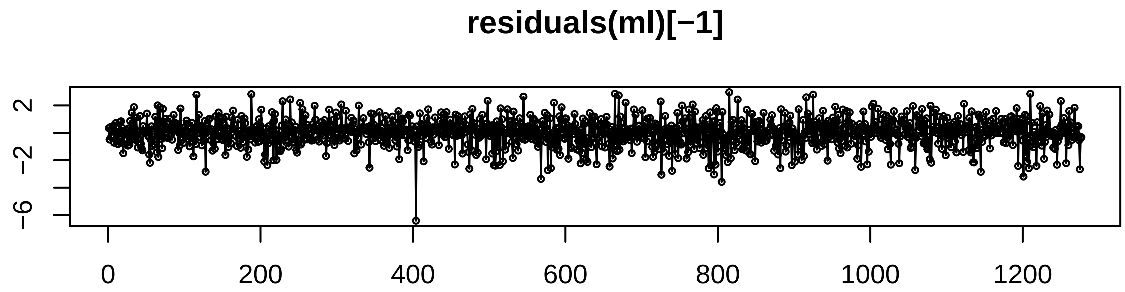


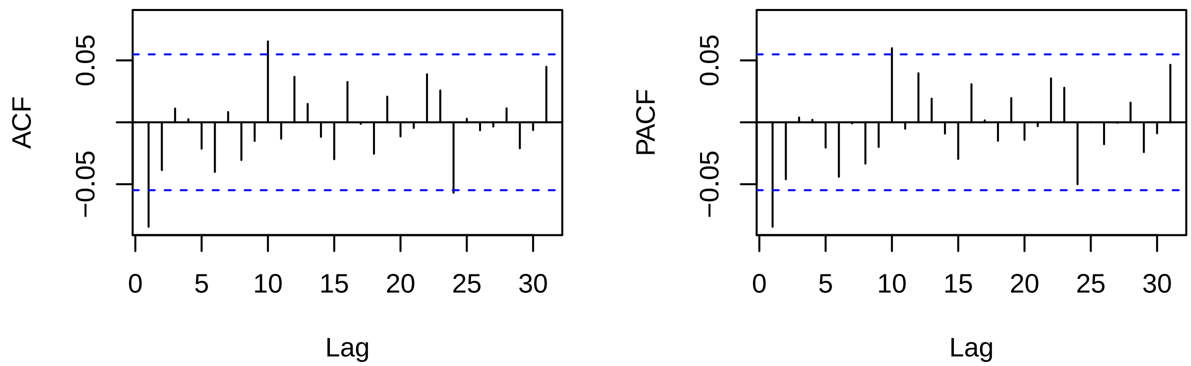
Test of Residual

```
plot(residuals(ml),type = "h", ylab = "standard residuals")
```

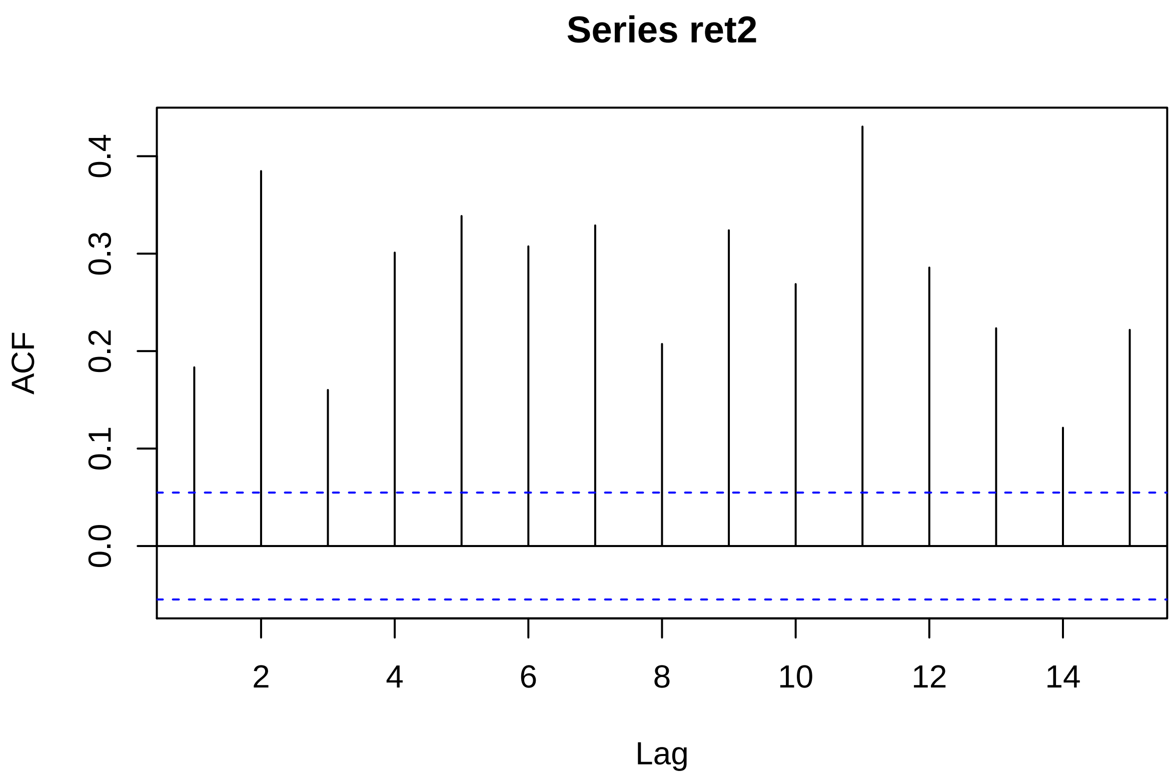



```
tsdisplay(residuals(ml)[-1])
```



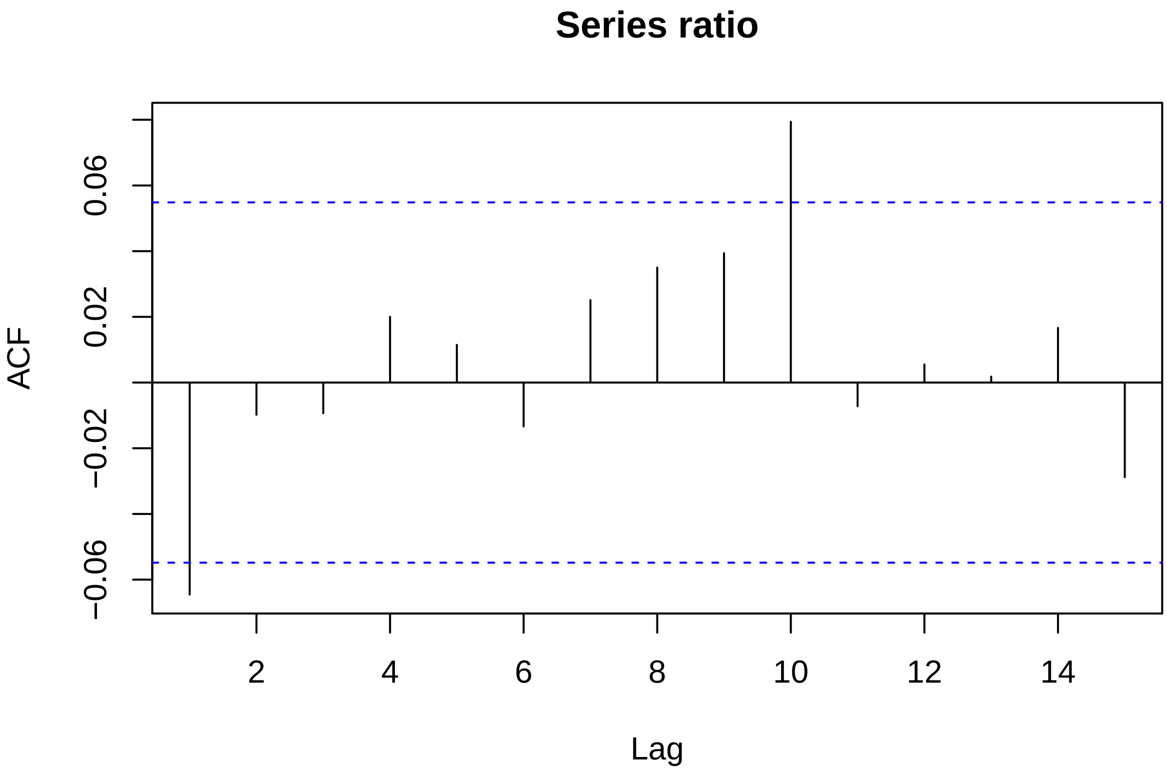



```r
# autrocorrelation for squared returns
ret2 = ret^2
m2 = acf(ret2, lag.max = 15)
```



```r
# if lag is within blue band, it is white noise
col1 = m2$acf
```

```r
# autocorrelation for the ratio of squared return over fitted daily variance
ratio = ret2[-1]/vfit^2
m3 = acf(ratio,15)
```




```r
col2 = m3$acf[-1]
```

```r
# table
col1 = m2$acf[-1]
col2 = m3$acf[-1]
df3 = data.frame(ret2 = col1, ratio = col2)
print(df3)
```

```
##         ret2        ratio
## 1  0.3847384 -0.009807882
## 2  0.1601414 -0.009358168
## 3  0.3010945  0.020028486
## 4  0.3386284  0.011479485
## 5  0.3074964 -0.013356350
## 6  0.3289945  0.025121788
## 7  0.2073292  0.035024266
## 8  0.3239728  0.039379281
## 9  0.2688047  0.079414876
## 10 0.4303820 -0.007233287
## 11 0.2857194  0.005529120
## 12 0.2234592  0.001819274
## 13 0.1213700  0.016635141
## 14 0.2218163 -0.028807675
```

```r
#ljung box test for squared return
Box.test(ret2, lag = 15, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  ret2
## X-squared = 1566.3, df = 15, p-value < 2.2e-16
```

```r
# ljung box test for ratio
Box.test(ratio, lag = 15, type = 'Ljung-Box')
```

```
##
##  Box-Ljung test
##
## data:  ratio
## X-squared = 20.545, df = 15, p-value = 0.152
```