Homework 5 Solutions

Ian Cranfield SID [redacted]

Due 6/2/2020 5pm

Packages

```
library(urca)
library(vars)
library(lmtest)
library(seasonal)
library(lubridate)
library(dplyr)
library(forecast)
library(timeSeries)
library(tseries)
library(strucchange)
library(tis)
library(tidyverse)
library(tseries)
library(forecast)
library(readxl)
library(TSA)
library(rugarch)
library(fGarch)
library(ggfortify)
library(quantmod)
library(dynlm)
library(FinTS)
library(data.table)
library(AnalyzeTS)
```

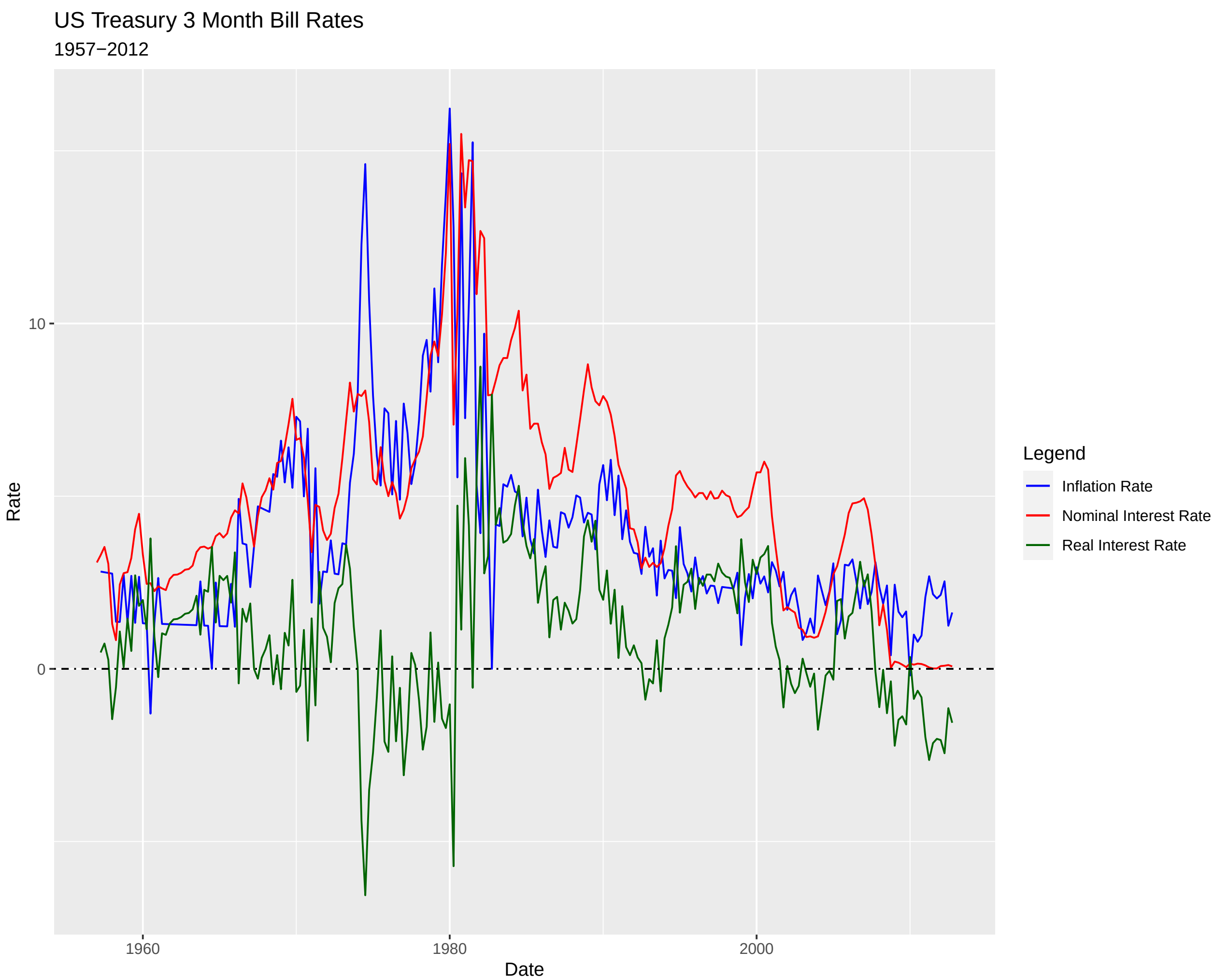
Problem 12.4:

```
UStreasury <- read_xls("Chapter12_exercises_data.xls", sheet = 3)

UStreasury$pastCPI = shift(UStreasury$cpi, n = 1L, fill = NA, type = "lag")

UStreasury$inf = (((UStreasury$cpi / UStreasury$pastCPI)^4 - 1) * 100)

UStreasury$real = (UStreasury$nominal - UStreasury$inf)

inflation <- ts(UStreasury$inf, start = 1957, frequency = 4)

nom.rates <- ts(UStreasury$nominal, start = 1957, frequency = 4)
```

```r
real.rates <- ts(UStreasury$real, start = 1957, frequency = 4)

#autoplot(inflation, color = "blue") + autolayer(nom.rates) + autolayer(real.rates, color = "dark green
```

```r
cols <- c("Inflation Rate"="blue", "Nominal Interest Rate"="red" , "Real Interest Rate" = "dark green")
ggplot(UStreasury , aes(x = date)) +
  geom_line(aes(y = inf, color="Inflation Rate")) +
  geom_line(aes(y = nominal, color="Nominal Interest Rate")) +
  geom_line(aes(y = real, color = "Real Interest Rate")) +
  geom_hline(yintercept = 0, linetype = 4) +
  ggtitle("US Treasury 3 Month Bill Rates", "1957-2012") +
  xlab("Date") +
  ylab("Rate") +
  scale_color_manual("Legend", values = cols)
```

```
## Warning: Removed 1 row(s) containing missing values (geom_path).

## Warning: Removed 1 row(s) containing missing values (geom_path).
```




```r
adf.test(inflation[2:224])
```

```
##
```

```
##  Augmented Dickey-Fuller Test
##
## data:  inflation[2:224]
## Dickey-Fuller = -2.4966, Lag order = 6, p-value = 0.3675
## alternative hypothesis: stationary
```

```r
adf.test(UStreasury$nominal)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  UStreasury$nominal
## Dickey-Fuller = -2.9123, Lag order = 6, p-value = 0.1929
## alternative hypothesis: stationary
```

```r
adf.test(UStreasury$real[2:224])
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  UStreasury$real[2:224]
## Dickey-Fuller = -3.0454, Lag order = 6, p-value = 0.1371
## alternative hypothesis: stationary
```

Case II: Critical Value -2.86, reject (it is stationary) if it is less than -2.86

-2.5 < -2.86 -> Inflation Rate is Non Stationary

-2.9 < -2.86 -> Nominal Interest rate is Stationary

-3.04 < -2.86 -> Real Interest Rate is Stationary

Inflation is non-stationary, but Nominal Interest Rate and real interest rate are stationary. Inflation and Nominal have a cointegration relation in real interest rate.

Problem 12.5:

```r
long.short.rates <- read_excel("Chapter12_exercises_data.xls", sheet = 4)

long.short.rates$obs <- gsub("M", "/01/", long.short.rates$obs)

long.short.rates$obs <- as.Date(long.short.rates$obs, format = '%Y/%d/%m')

long.short.rates$TB3Month <- as.numeric(long.short.rates$TB3Month)
```

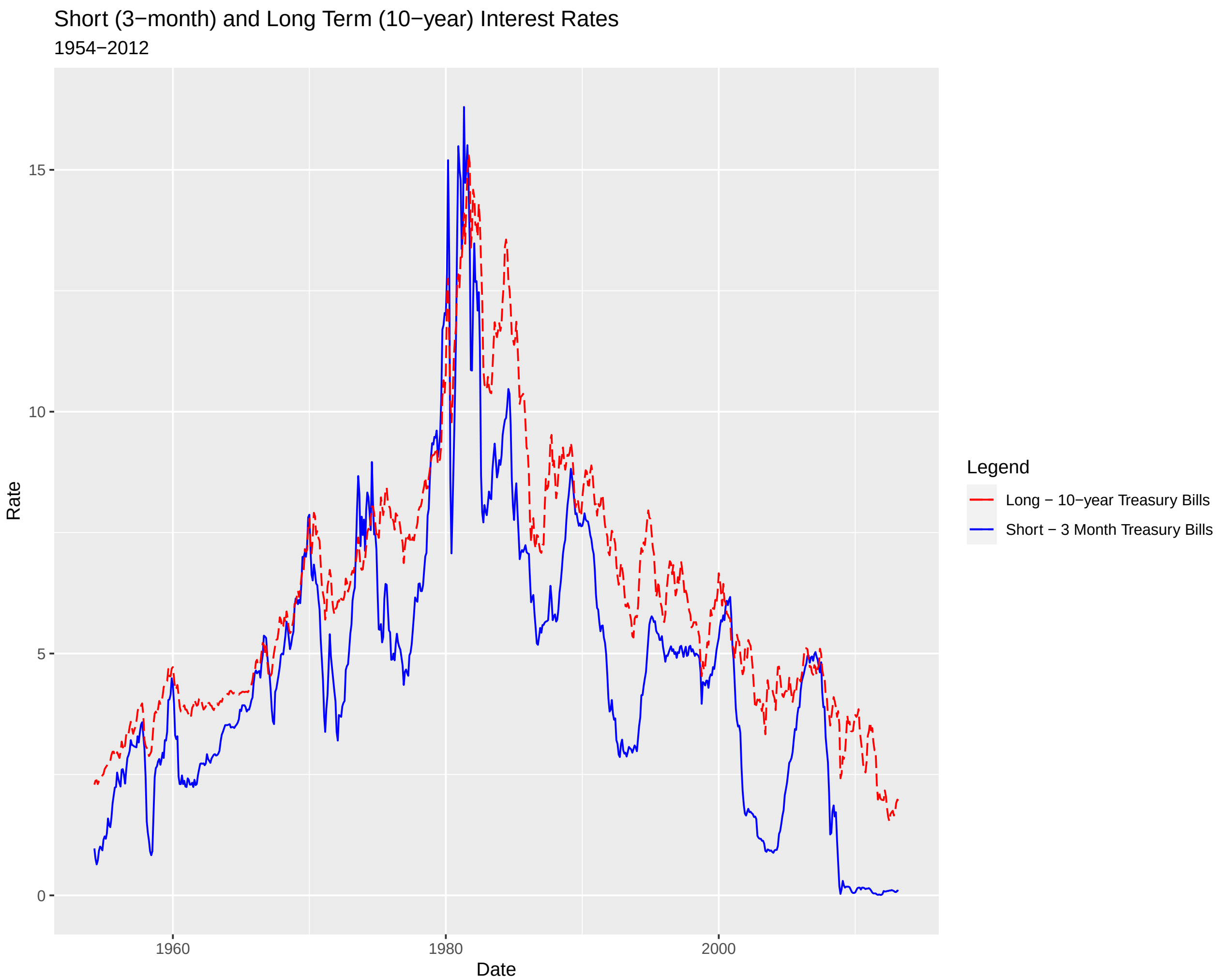
```
## Warning: NAs introduced by coercion
```

```r
long.short.rates$TB10Year <- as.numeric(long.short.rates$TB10Year)
```

```
## Warning: NAs introduced by coercion
```

```r
long.short.rates <- na.omit(long.short.rates)

TB3Month <- ts(long.short.rates$TB3Month, start = c(1954,4), frequency = 12)

TB10Year <- ts(long.short.rates$TB10Year, start = c(1954,4), frequency = 12)
```

```r
cols2 <- c("Short - 3 Month Treasury Bills" = "blue", "Long - 10-year Treasury Bills" = "red")
ggplot(long.short.rates, aes(x = obs)) +
   geom_line(aes(y = TB3Month, color="Short - 3 Month Treasury Bills")) +
  geom_line(linetype = 5, aes(y = TB10Year, color="Long - 10-year Treasury Bills")) +
  ggtitle("Short (3-month) and Long Term (10-year) Interest Rates", "1954-2012") +
  xlab("Date") +
  ylab("Rate") +
  scale_color_manual("Legend", values = cols2)
```




```r
adf.test(TB3Month[4:708])
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  TB3Month[4:708]
## Dickey-Fuller = -2.2334, Lag order = 8, p-value = 0.4795
## alternative hypothesis: stationary
```

```r
adf.test(TB10Year[4:708])
```

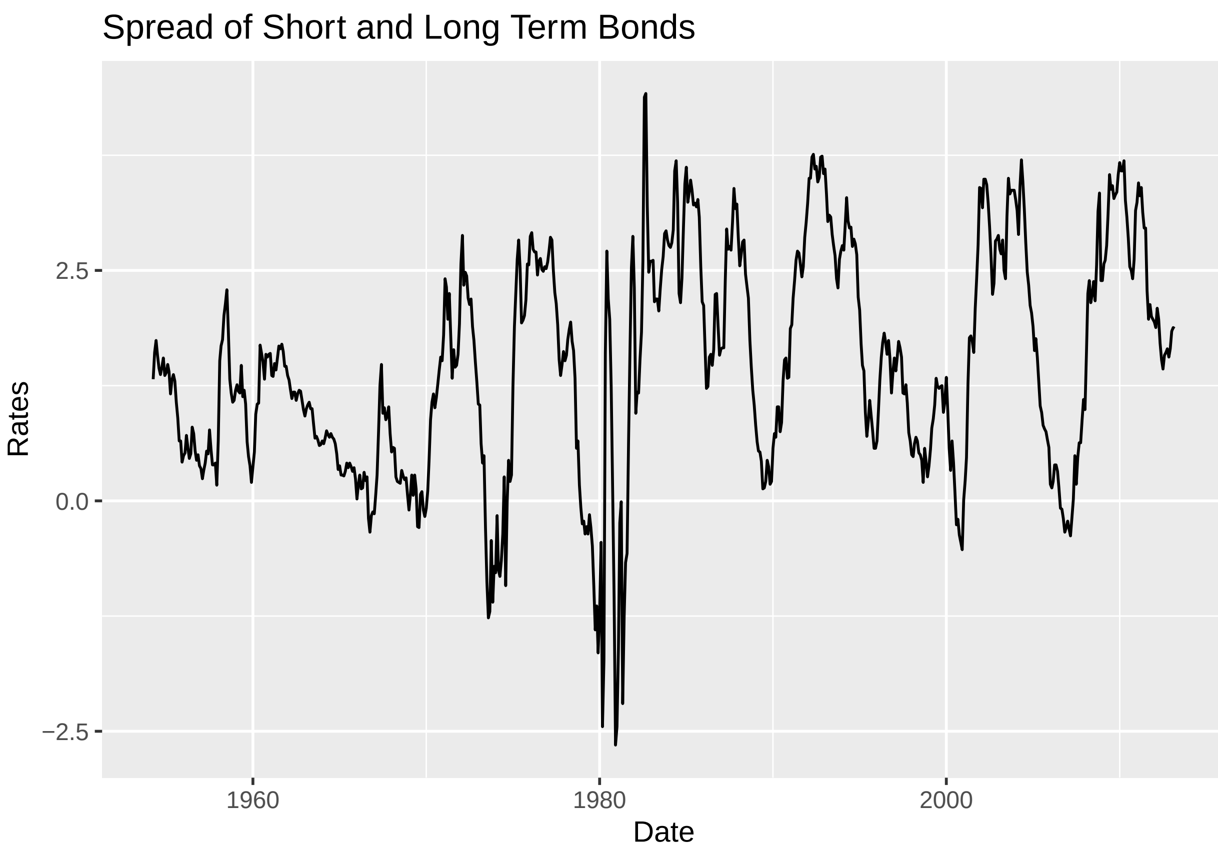
```
##
##  Augmented Dickey-Fuller Test
```

```
##
## data:  TB10Year[4:708]
## Dickey-Fuller = -1.4909, Lag order = 8, p-value = 0.7938
## alternative hypothesis: stationary
```

These fail to reject the null hypothesis and therefore, they are non-stationary.

I then calculate the spread which is the difference between 10 Year Treasury Bond and 3 Month Treasury Bill

```
long.short.rates$spread <- (long.short.rates$TB10Year - long.short.rates$TB3Month)

spreadts <- ts(long.short.rates$spread , start = c(1954,4), frequency = 12)

ggplot(long.short.rates, aes(x = obs, y = spread)) +
  geom_line() +
  ggtitle("Spread of Short and Long Term Bonds") +
  ylab("Rates") +
  xlab("Date")
```




```
adf.test(spreadts)
```

```
## Warning in adf.test(spreadts): p-value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  spreadts
## Dickey-Fuller = -4.5083, Lag order = 8, p-value = 0.01
## alternative hypothesis: stationary
```

Both the long term and short term Treasury Bonds are non-stationary because the critical values are greater than the tau statistic critical value at 5% level.

Then, I constructed the spread of the Long and Short term bonds.

By doing the Augmented Dickey-Fuller Test of the spread, it has a lower critical value and smaller p-value so we can reject the hypothesis for a unit root test and we can conclude the spread series is stationary and cointegrating between long and short term interest rates.

Problem 12.6:

```
library(urca)
library(vars)

#Bind Data for Var model
vecm_data = cbind(TB10Year,  TB3Month)

## cointegration

#do VAR select
VARselect(vecm_data)
```

```
## $selection
## AIC(n)  HQ(n)  SC(n) FPE(n)
##     10     10      3     10
##
## $criteria
##                    1            2            3            4            5
## AIC(n) -4.656152814 -4.874166620 -4.944705765 -4.960375243 -4.960218931
## HQ(n)  -4.641037746 -4.848974840 -4.909437273 -4.915030039 -4.904797016
## SC(n)  -4.617056375 -4.809005888 -4.853480740 -4.843085925 -4.816865321
## FPE(n)  0.009502953  0.007641464  0.007121019  0.007010317  0.007011429
##                    6            7           8            9           10
## AIC(n) -4.957633545 -5.020254230 -5.02619264 -5.033471806 -5.054629077
## HQ(n)  -4.892134917 -4.944678890 -4.94054059 -4.937743042 -4.948823601
## SC(n)  -4.788215641 -4.824772034 -4.80464615 -4.785861024 -4.780954002
## FPE(n)  0.007029604  0.006602935  0.00656388  0.006516324  0.006379964
```

```
VARselect(vecm_data, lag.max = 10, type="const")$selection
```

```
## AIC(n)  HQ(n)  SC(n) FPE(n)
##     10     10      3     10
```

```
# Choose Lag 10

# Conduct Eigen test (Cointegration Test)
VECM <- ca.jo(vecm_data, K = 10, type = "trace", ecdet = "const", spec = "transitory")

#Make a ca.jo object to convert in vecm and var (Lags K should be minimum 2)

VECM@teststat[2] #Test statistic H0 r=0, to be rejected
```

```
## [1] 35.07363
```

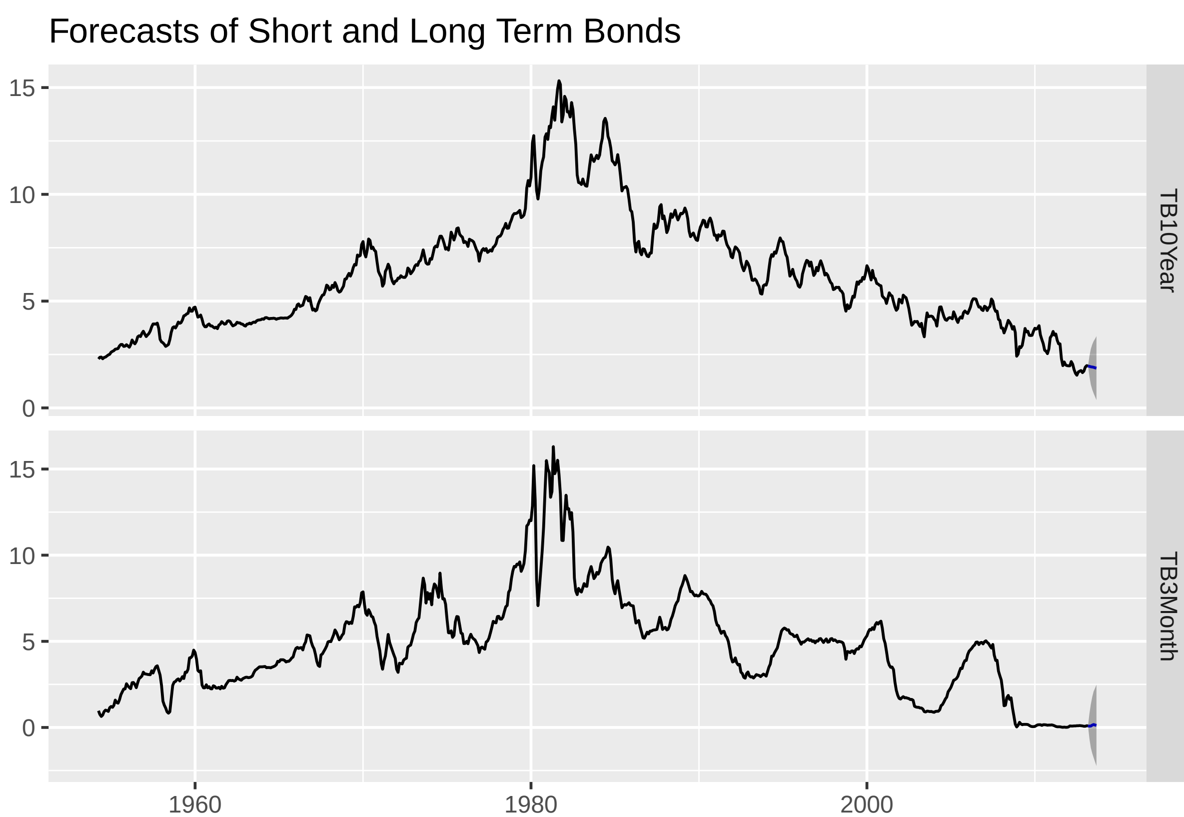
```r
VECM@teststat[1] #Test statisitc H0 r=1 should not be rejected
```

```
## [1] 2.898508
```

```r
#Take critical values
VECM@cval
```

```
##           10pct  5pct  1pct
## r <= 1 |  7.52  9.24 12.97
## r = 0  | 17.85 19.96 24.60
```

```r
### Cointest [2] = 35.07363 > 19.96, reject null that they are not cointegrated and say that they are c

# WE CAN Run VECM
vecm <- cajorls(VECM) #convert to vecm, select on just "cajorls(cointest)" and enter to see the model

var <- vec2var(VECM, r = 1)

predict(var,pred.var=var, n.ahead = 6) %>%
  autoplot() + ggtitle("Forecasts of Short and Long Term Bonds")
```




```r
predict(var, pred.var = var, n.ahead = 6)
```

```
## $TB10Year
##          fcst     lower    upper        CI
## [1,] 1.945097 1.4671903 2.423003 0.4779063
## [2,] 1.924695 1.0919989 2.757391 0.8326959
## [3,] 1.914591 0.8587613 2.970420 1.0558295
## [4,] 1.902685 0.6824303 3.122939 1.2202542
## [5,] 1.877635 0.5205966 3.234674 1.3570387
```

```
## [6,] 1.862078 0.3656690 3.358487 1.4964088
##
## $TB3Month
##              fcst      lower     upper        CI
## [1,] 0.07585386 -0.6478168 0.7995245 0.7236706
## [2,] 0.08193180 -1.1809576 1.3448212 1.2628894
## [3,] 0.13583691 -1.4988867 1.7705605 1.6347236
## [4,] 0.17051722 -1.7524447 2.0934792 1.9229620
## [5,] 0.14024564 -2.0011074 2.2815986 2.1413530
## [6,] 0.12294537 -2.2351002 2.4809910 2.3580456
```

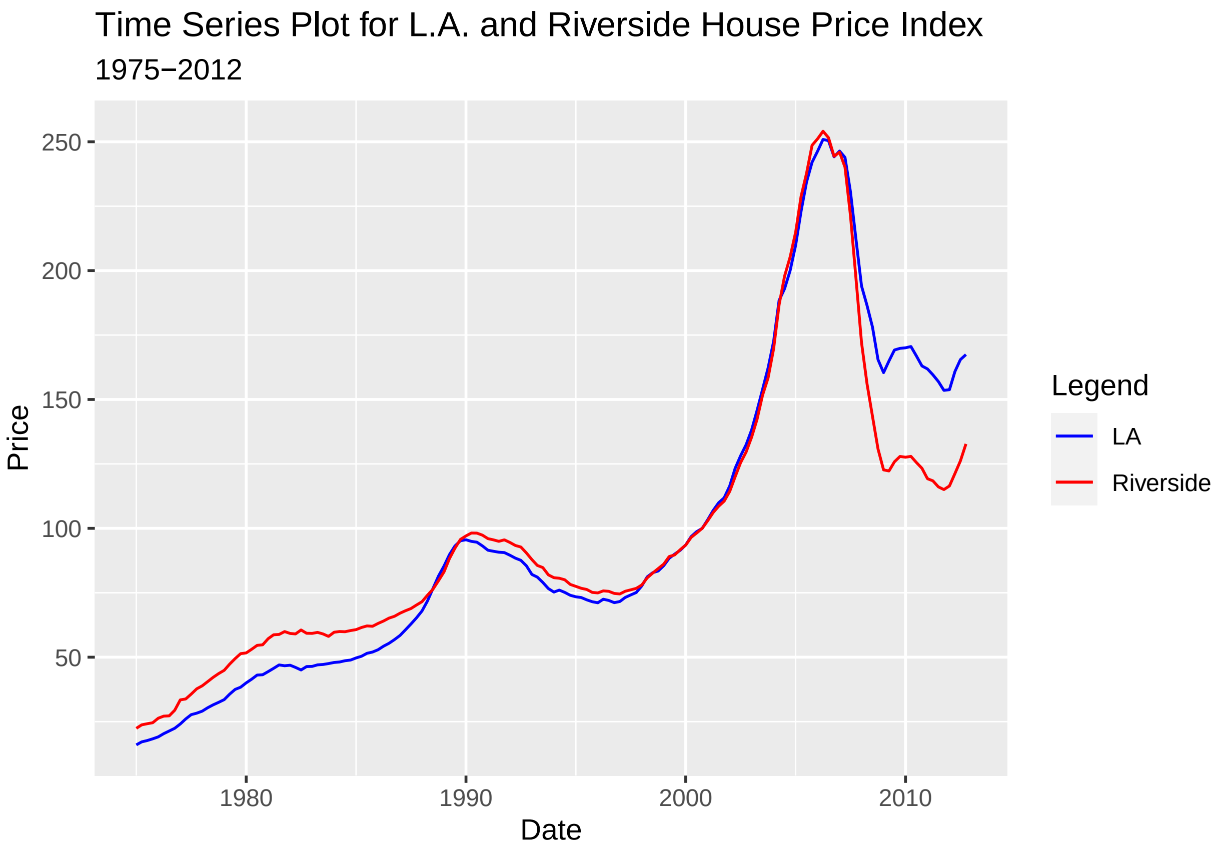
Built a forecasting model using Vector Error Correction Model and make it into a VAR model.

Problem 12.7:

```
house.price.LA.River <- read_excel("Chapter12_exercises_data.xls", sheet = 5)[1:3]
```

```
## New names:
## * `` -> ...4
## * `` -> ...5
```

```
house.price.LA.River$obs <- gsub("Q1", "/01/01", house.price.LA.River$obs)
house.price.LA.River$obs <- gsub("Q2", "/04/01", house.price.LA.River$obs)
house.price.LA.River$obs <- gsub("Q3", "/07/01", house.price.LA.River$obs)
house.price.LA.River$obs <- gsub("Q4", "/10/01", house.price.LA.River$obs)

house.price.LA.River$obs <- as.Date(house.price.LA.River$obs, fformat = "%Y/%m/%d")


LAts <- ts(house.price.LA.River$ILA, start = 1975, frequency = 4)

riverts <- ts(house.price.LA.River$IRV, start = 1975, frequency = 4)

cols3 <- c("LA" = "blue", "Riverside" = "red")
ggplot(house.price.LA.River, aes(x = obs)) +
   geom_line(aes(y = ILA, color="LA")) +
  geom_line(aes(y = IRV, color="Riverside")) +
  ggtitle("Time Series Plot for L.A. and Riverside House Price Index ", "1975-2012") +
  xlab("Date") +
  ylab("Price") +
  scale_color_manual("Legend", values = cols3)
```



Plotted the time series of Los Angeles and Riverside House Price Index.

```
adf.test(LAts)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  LAts
## Dickey-Fuller = -2.9351, Lag order = 5, p-value = 0.1867
## alternative hypothesis: stationary
```

```
adf.test(riverts)
```

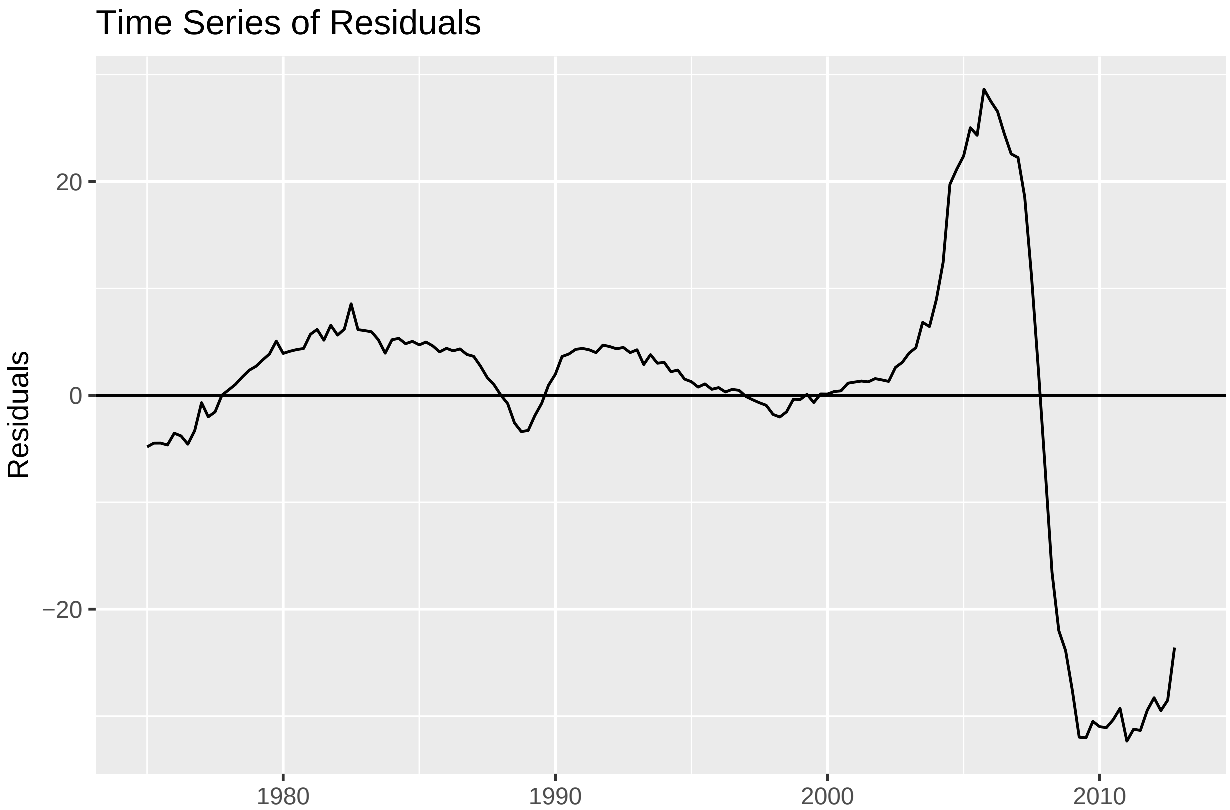
```
##
##  Augmented Dickey-Fuller Test
##
## data:  riverts
## Dickey-Fuller = -3.3513, Lag order = 5, p-value = 0.06532
## alternative hypothesis: stationary
```

```
OLS.LA.River <- lm(IRV ~ ILA, data = house.price.LA.River)

summary(OLS.LA.River)
```

```
##
## Call:
## lm(formula = IRV ~ ILA, data = house.price.LA.River)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -32.343  -0.817   1.843   4.497  28.641
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 13.59770    1.94832   6.979 8.92e-11 ***
## ILA          0.85267    0.01686  50.579  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.75 on 150 degrees of freedom
## Multiple R-squared:  0.9446, Adjusted R-squared:  0.9442
## F-statistic:  2558 on 1 and 150 DF,  p-value: < 2.2e-16
```

```
OLS.LA.ts <- ts(OLS.LA.River$residuals, start = 1975, frequency = 4)

autoplot(OLS.LA.ts, color = "dark green") + ylab("Residuals") + ggtitle("Time Series of Residuals") + g
```




```
adf.test(OLS.LA.ts[1:100])
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  OLS.LA.ts[1:100]
## Dickey-Fuller = -3.024, Lag order = 4, p-value = 0.1524
## alternative hypothesis: stationary
```

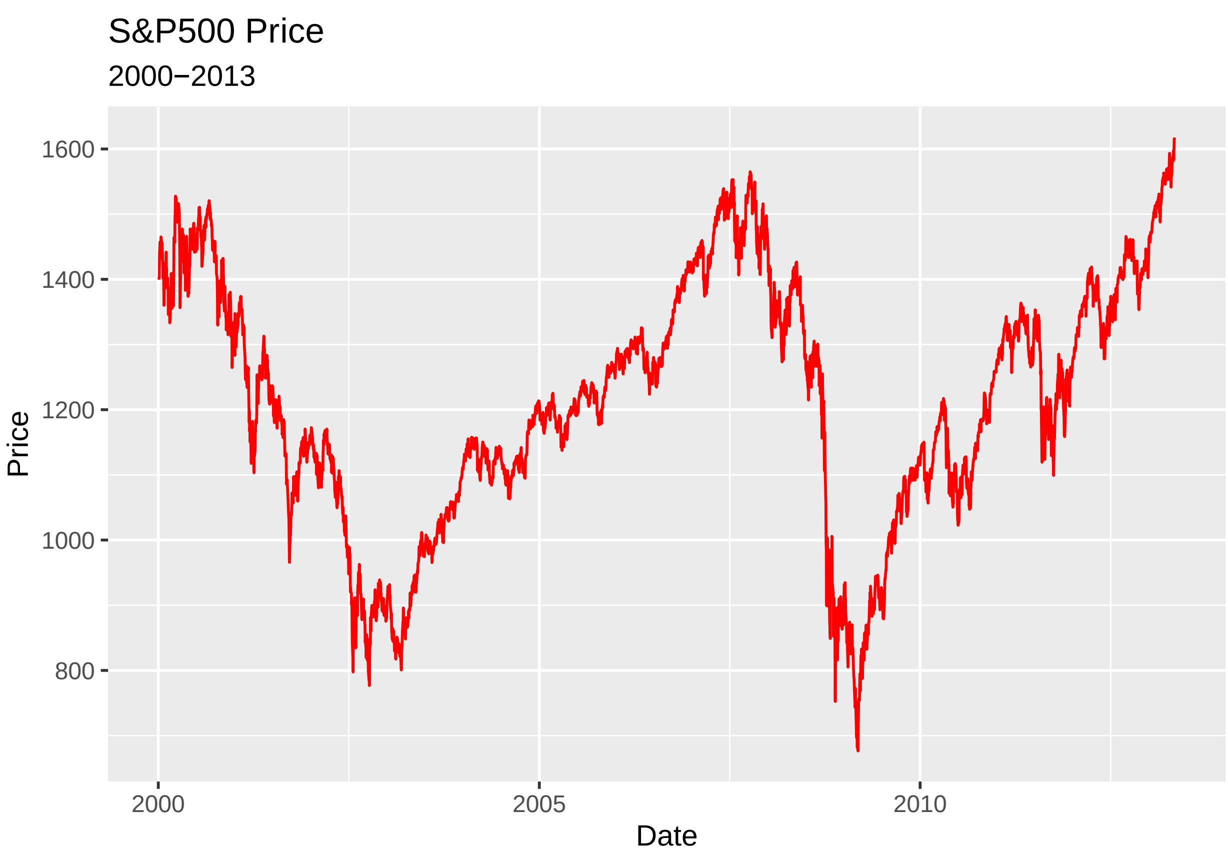
Fail to reject the null hypothesis of unit root at 5% significance level of the series of LA and Riverside.

Then I estimated a regression model, and running the ADF test, it has a larger value than the 5% critical value so therefore we fail to reject the null hypothesis that the reaidual series has a unit root.

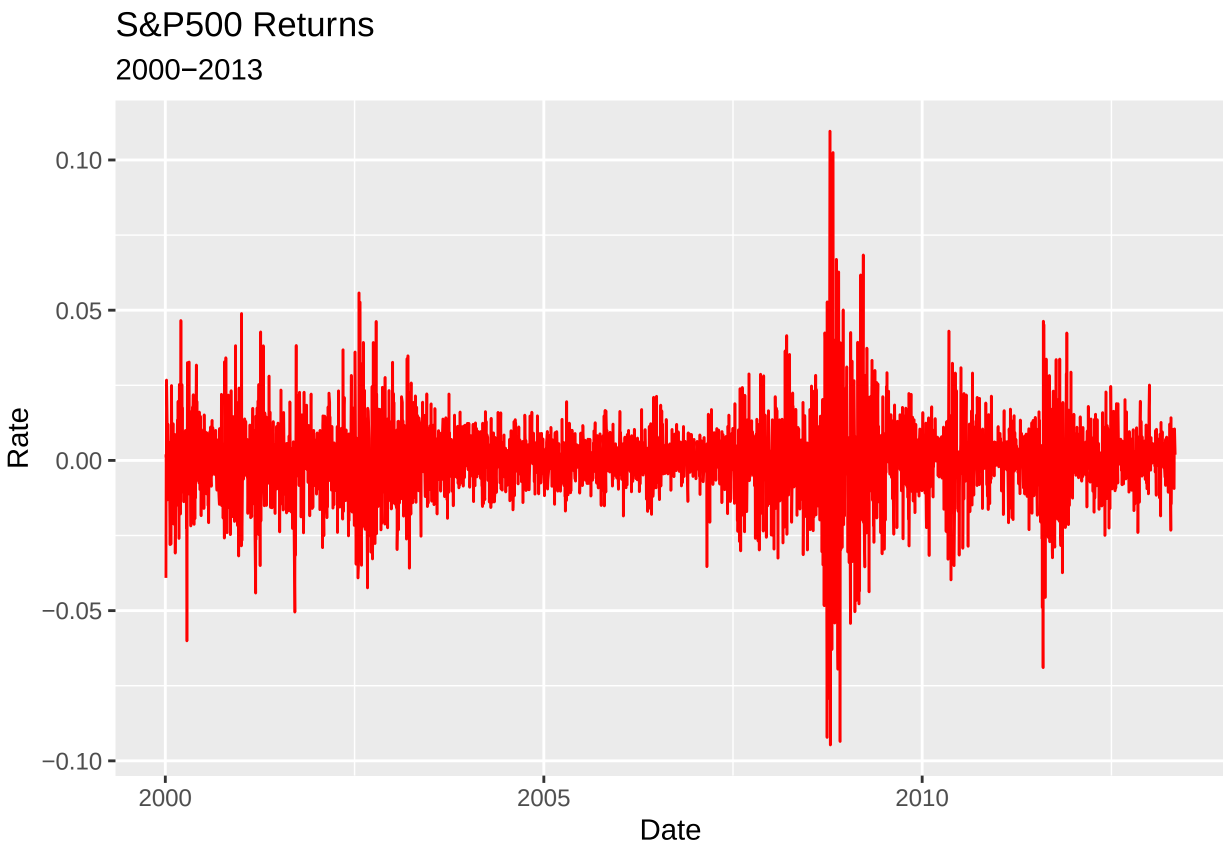
With this, I conclude that no cointegration exists between Los Angeles and Riverside house prices.

Problem 14.4:

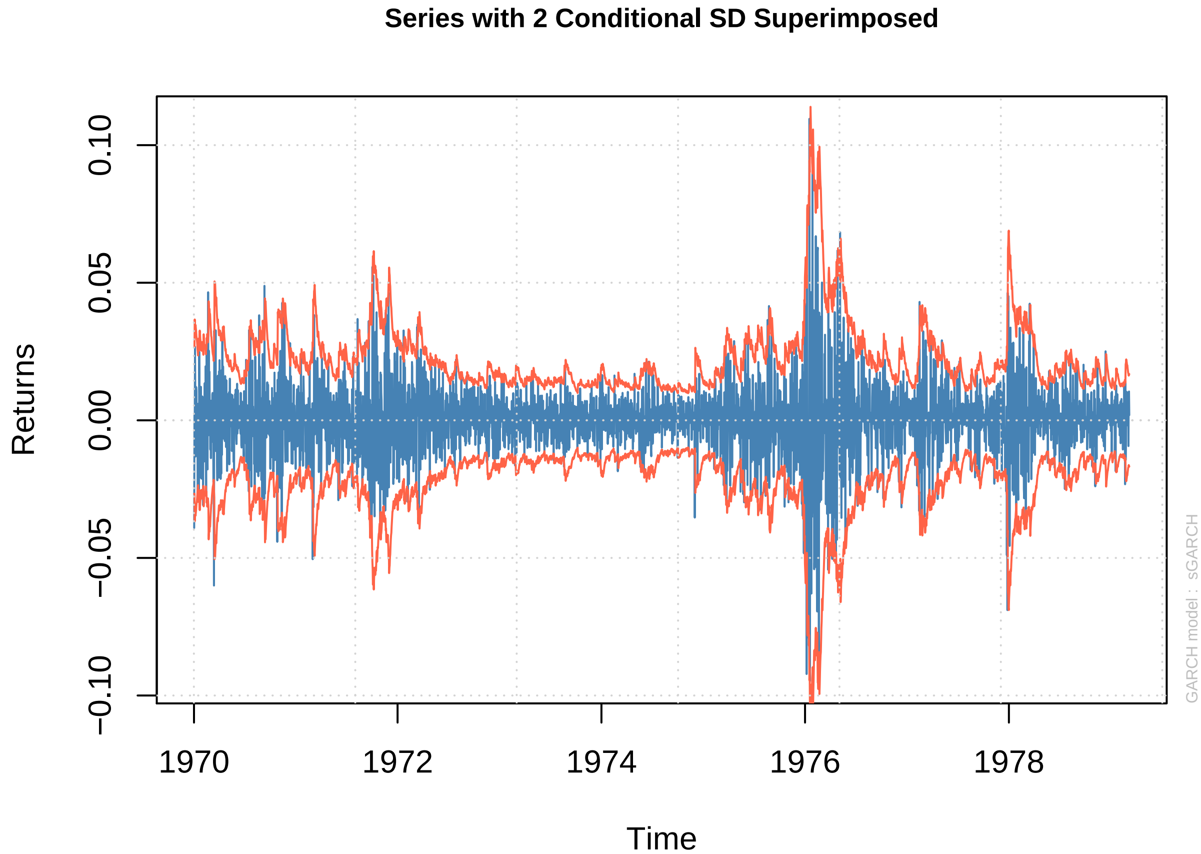
```
SP500index <- read_excel("Chapter14_exercises_data.xls", sheet = 1)

SPIndex <- data.frame(SP500index$Date, SP500index$`Adj Close`) %>%
  set_names("Date", "Adj.Closed") %>%
  mutate(
    Return = c(NA,diff(log(Adj.Closed), lag =1))
    )

SPIndex <- na.omit(SPIndex)

ggplot(SPIndex, aes(x = Date, y = Adj.Closed)) +
  geom_line(color="red") +
  ggtitle("S&P500 Price", "2000-2013") +
  xlab("Date") +
  ylab("Price")
```



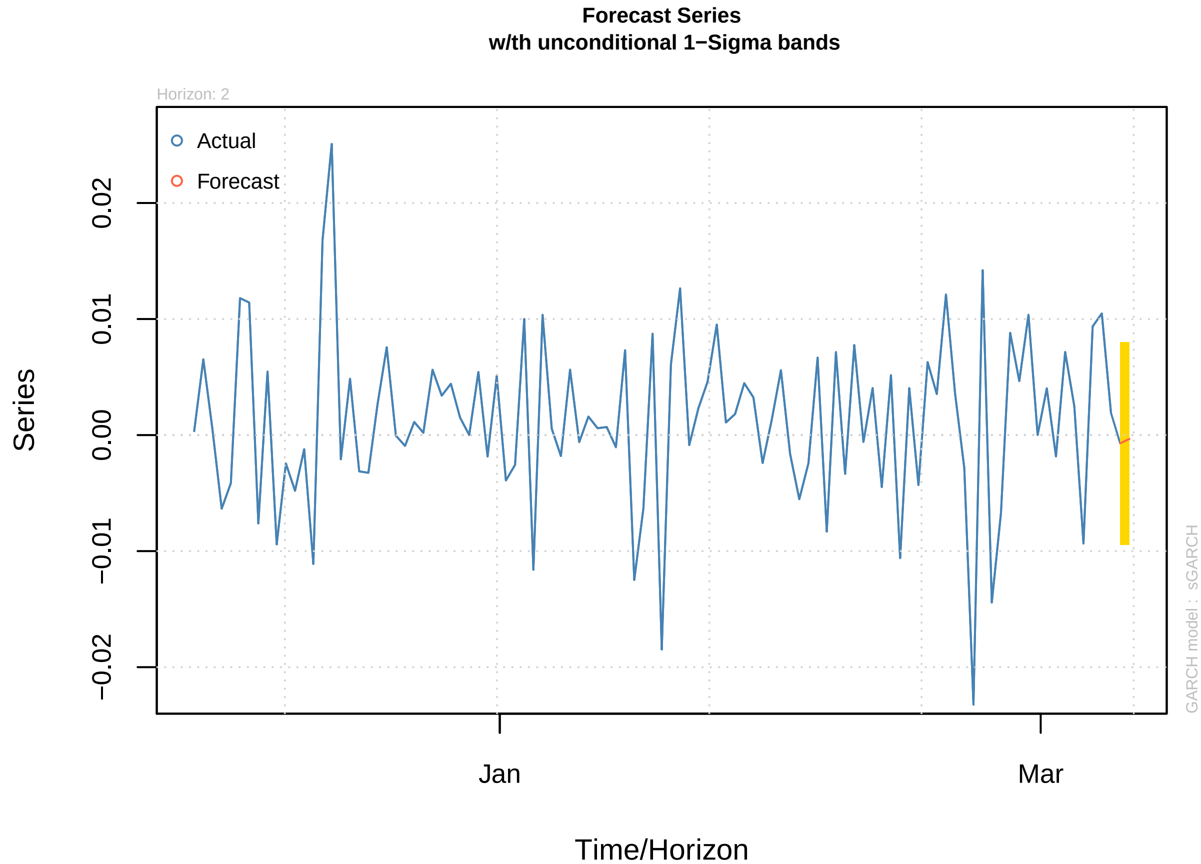

```
ggplot(SPIndex, aes(x = Date, y = Return)) +
  geom_line(color="red", na.rm=TRUE) +
  ggtitle("S&P500 Returns", "2000-2013") +
  xlab("Date") +
  ylab("Rate")
```



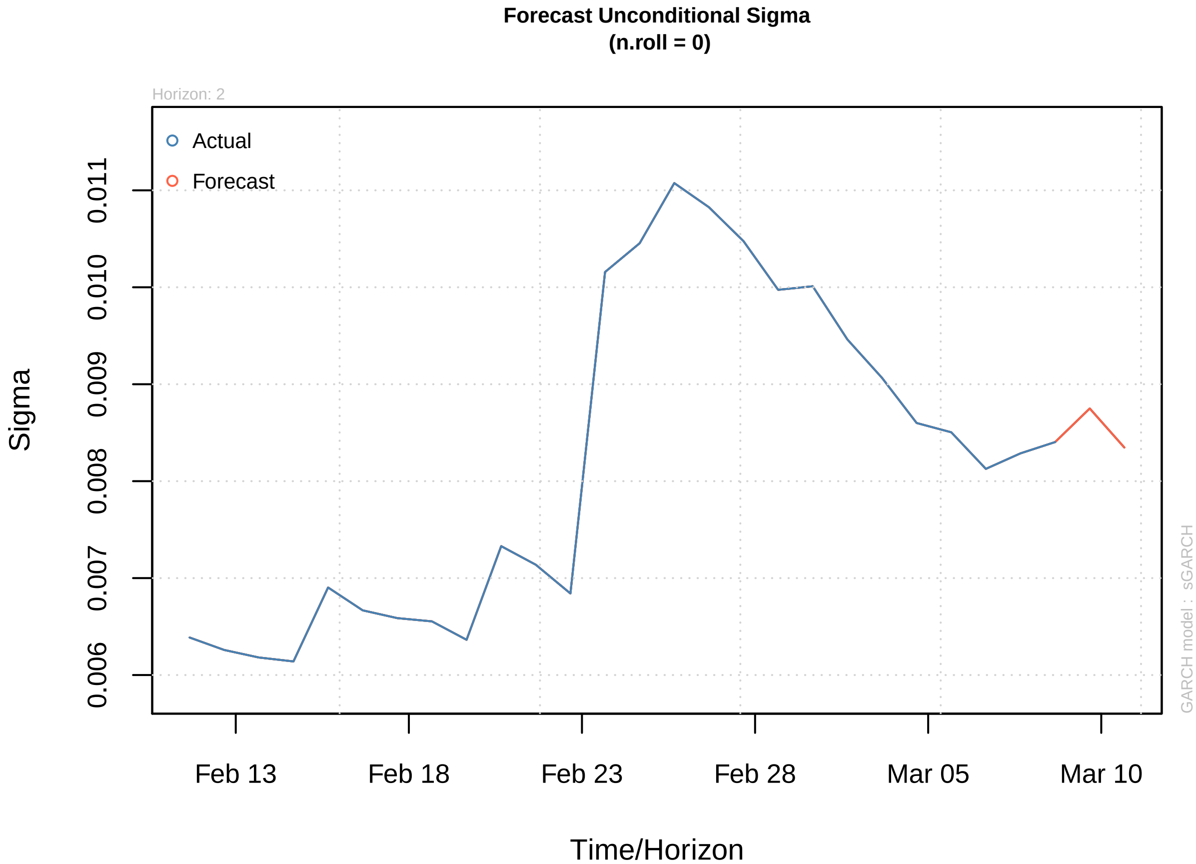

```r
# One and two step ahead forecasts

modelSP500 <- ugarchspec( variance.model = list(model = "sGARCH", garchOrder = c(2, 1)), mean.model = li

modelfitSP500 = ugarchfit(spec = modelSP500 , data = SPIndex$Return )
plot(modelfitSP500, which = 1)
```




```
modelforSP500 = ugarchforecast(modelfitSP500, data = NULL, n.ahead = 2, n.roll = 0, out.sample = 0)
plot(modelforSP500, which = 1)
```




```
plot(modelforSP500, which = 3)
```

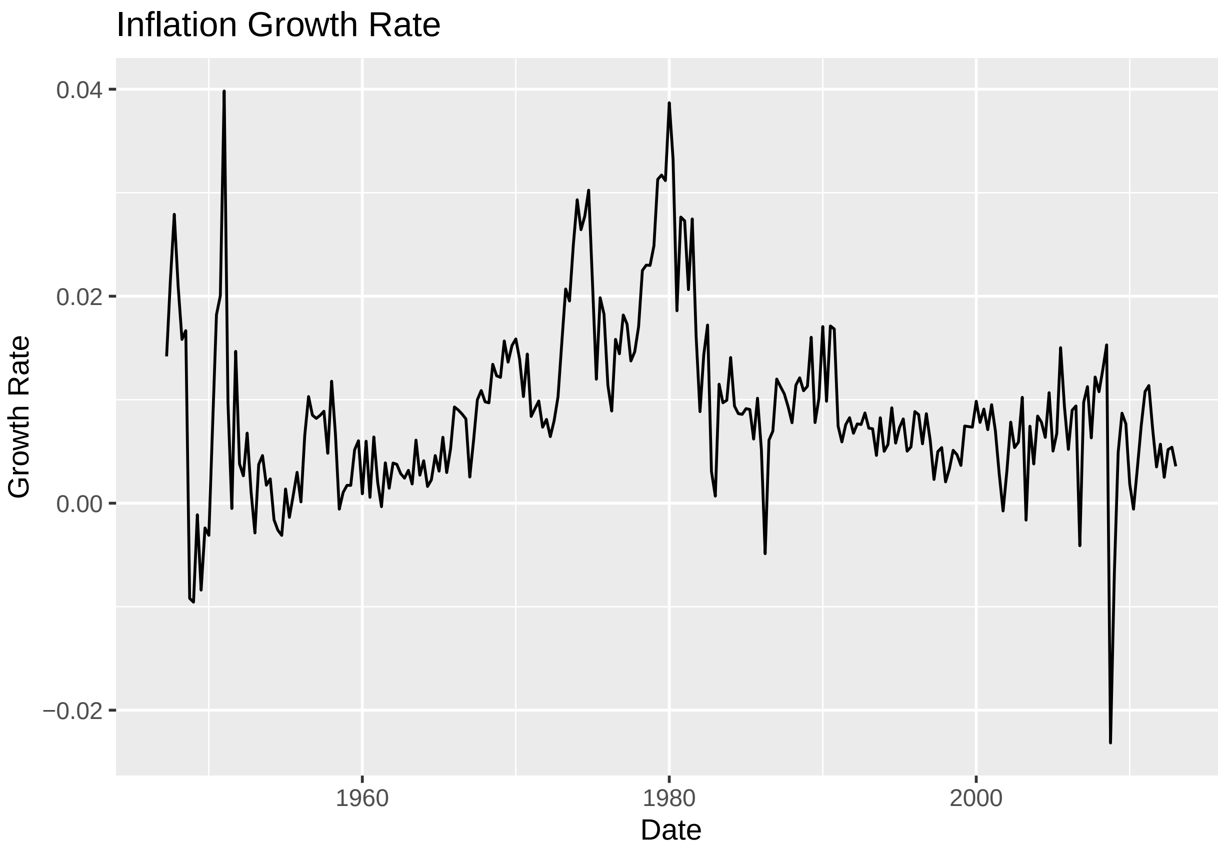


Here is my forecast using an ARMA(1,2) and GARCH order of (2,1). It looks like it fits pretty well and has a strong forecast.

Problem 14.5:

```
usCPI <- suppressMessages(read_excel("Chapter14_exercises_data.xls", sheet = 2)[1:2])
names(usCPI) <- c("Date", "CPI")

# make ts cpi
tscpi <- ts(usCPI$CPI, start = c(1947,1), frequency = 12)

# turn cpi into quarterly data, easier to work with
cpi <- ts(aggregate(tscpi, nfrequency = 4)/3, start = 1947, frequency = 4)

usGDP <- suppressMessages(read_excel("Chapter14_exercises_data.xls", sheet = 3)[1:2])
names(usGDP) <- c("Date", "GDP")

gdp <- ts(usGDP$GDP, start = 1947, frequency = 4)
```

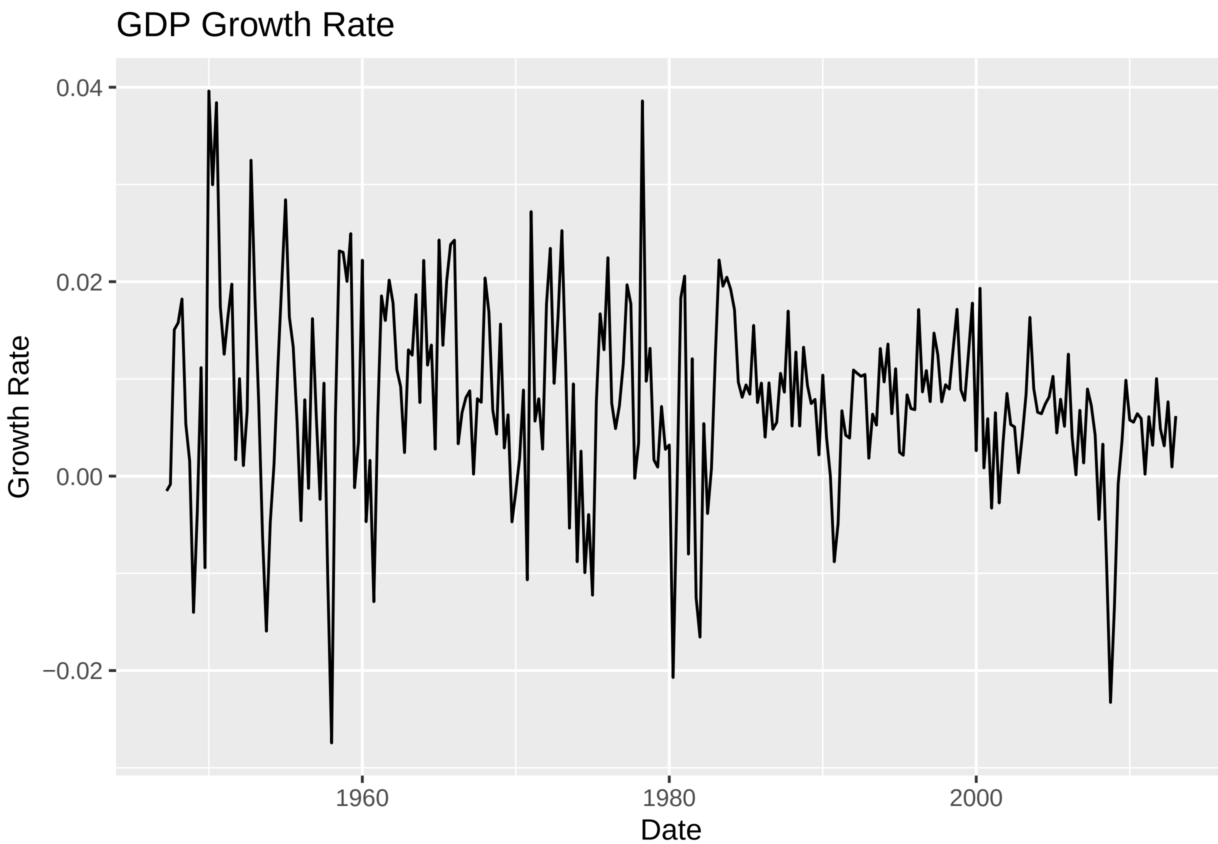
```
### cpi
cpi.growthts <- diff(log(cpi))

cpi.growth <- data.frame(date = c(time(cpi.growthts)), cpi = c(cpi.growthts))

autoplot(cpi.growthts) + ggtitle("Inflation Growth Rate") + ylab("Growth Rate") + xlab("Date")
```




```
## unconditional mean for out of sample forecast 2009Q1 to 2013Q1
mean(cpi.growth$cpi[1:247])
```

```
## [1] 0.009262981
```

```
### GDP
gdp.growthts <- diff(log(gdp))

gdp.growth <- data.frame(date = c(time(gdp.growthts)), gdp = c(gdp.growthts))


autoplot(gdp.growthts) + ggtitle("GDP Growth Rate") + ylab("Growth Rate") + xlab("Date")
```




```r
## unconditional mean for out of sample forecast 2009Q1 to 2013Q1
mean(gdp.growth$gdp[1:247])
```

```
## [1] 0.008034706
```

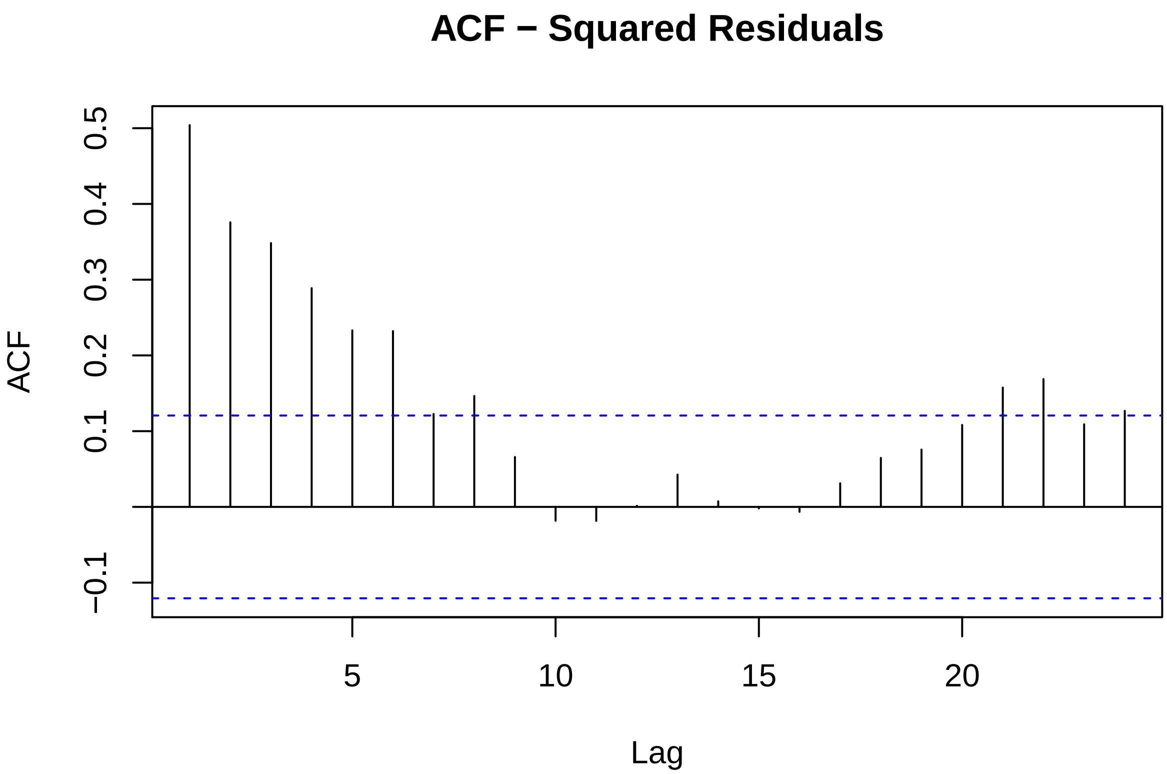
The unconditional mean of GDP Growth Rate is 0.80% and the unconditional mean of Inflation Growth Rate is 0.92%

```r
#### for CPI


# Step 1: Estimate mean equation r = beta + error
cpi.mean <- dynlm(cpi ~ 1, data = cpi.growth)

summary(cpi.mean)
```

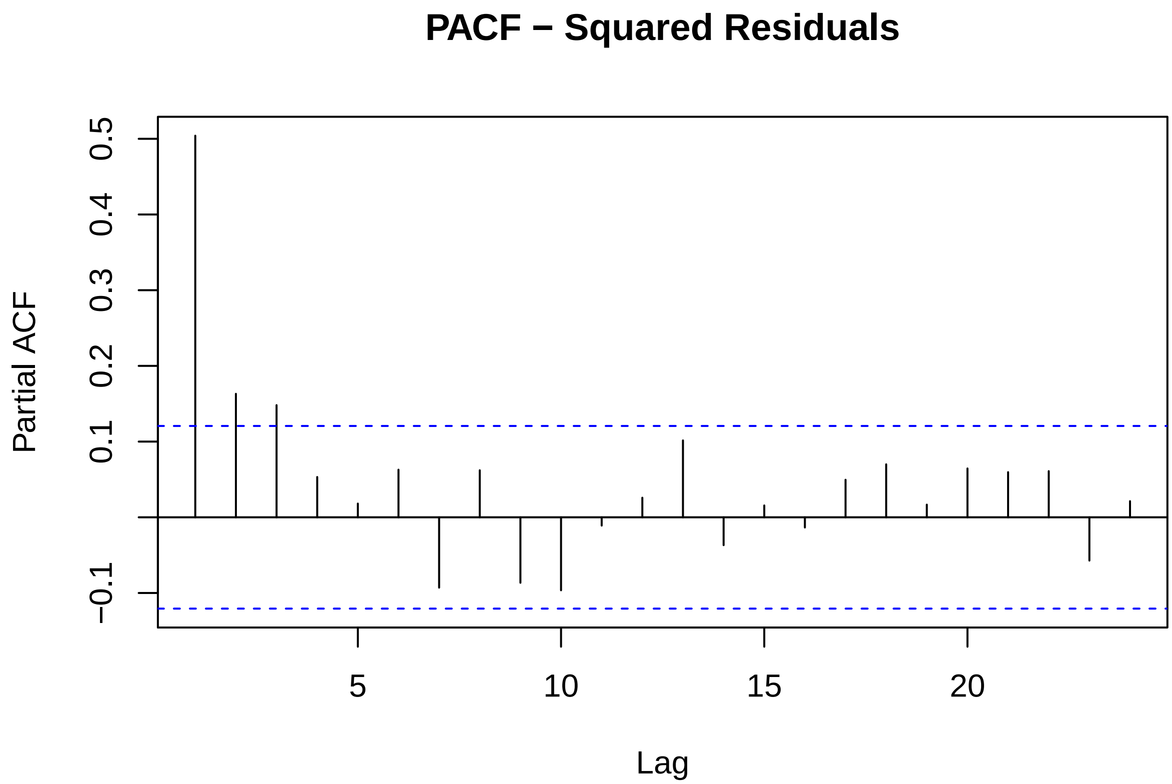
```
##
## Time series regression with "numeric" data:
## Start = 1, End = 264
##
## Call:
## dynlm(formula = cpi ~ 1, data = cpi.growth)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.032145 -0.004497 -0.001149  0.002856  0.030851
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) 0.0089768  0.0005008   17.93   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.008137 on 263 degrees of freedom
```

```r
# Step 2: Retrieve the residuals from the former model and square them
ehatsq <- ts(resid(cpi.mean)^2)
acf(ehatsq, main = "ACF - Squared Residuals")
```




```r
pacf(ehatsq, main = "PACF - Squared Residuals")
```




```r
# Step 3: regress squared residuals on one-lagged squared residuals
cpi.arch <- dynlm(ehatsq ~ L(ehatsq), data = ehatsq)

summary(cpi.arch)
```

```
##
## Time series regression with "ts" data:
## Start = 2, End = 264
##
## Call:
## dynlm(formula = ehatsq ~ L(ehatsq), data = ehatsq)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -5.122e-04 -3.508e-05 -2.871e-05  3.450e-06  9.804e-04
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.278e-05  8.162e-06   4.016 7.75e-05 ***
## L(ehatsq)   5.042e-01  5.345e-02   9.433  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0001193 on 261 degrees of freedom
## Multiple R-squared:  0.2543, Adjusted R-squared:  0.2514
## F-statistic: 88.99 on 1 and 261 DF,  p-value: < 2.2e-16
```

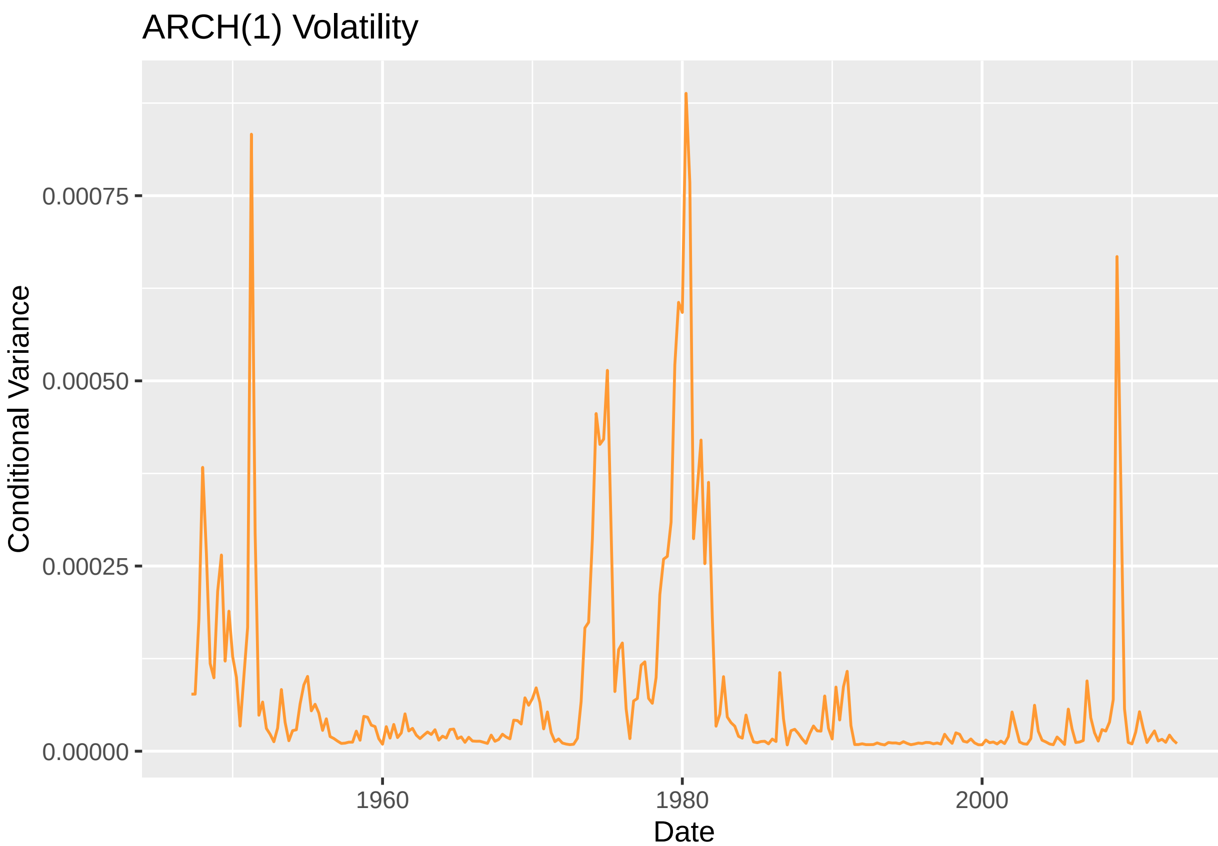
```r
## check ARCH effect

cpi.archTest <- ArchTest(cpi.growth, lags = 1, demean = TRUE)
```

```
cpi.archTest
```

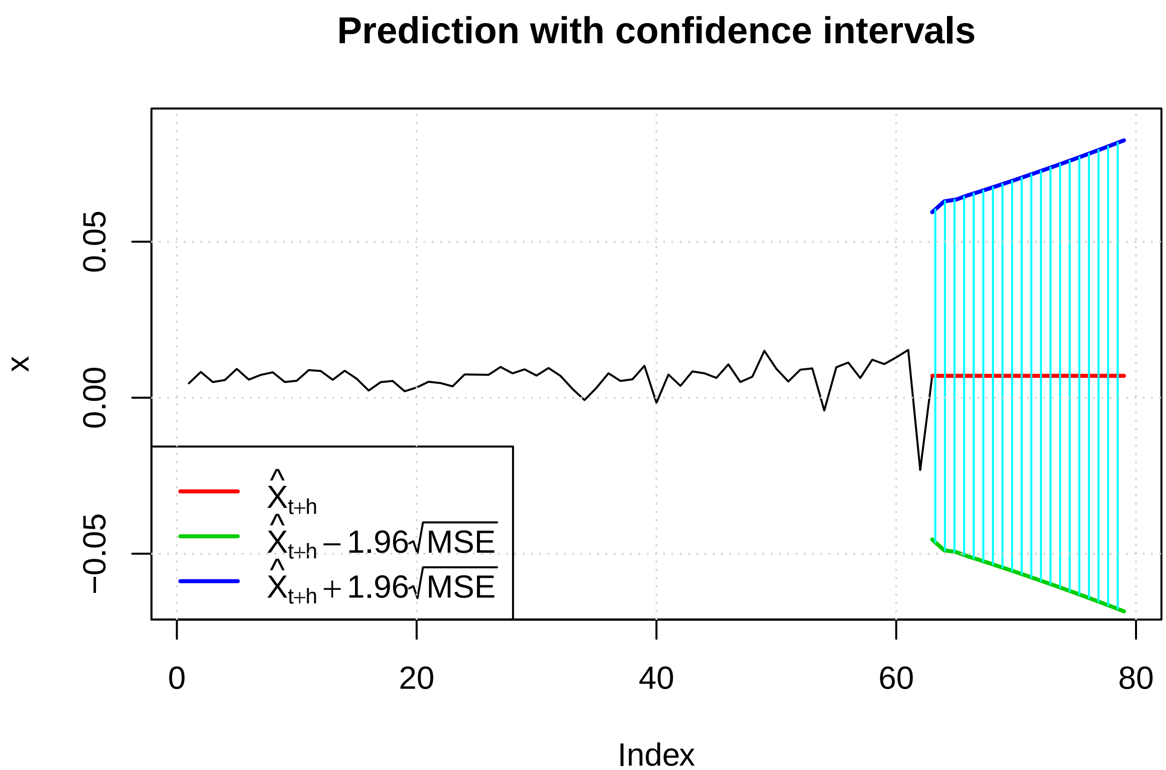
```
##
##  ARCH LM-test; Null hypothesis: no ARCH effects
##
## data:  cpi.growth
## Chi-squared = 262.78, df = 1, p-value < 2.2e-16
```

```
# low p-value so conclude presence of ARCH 1 effects

### estimate ARCH model
arch.fit <- garchFit(~garch(2,0), data = cpi.growth$cpi, trace = FALSE)
arch.fit1 <- garchFit(~garch(2,0), data = cpi.growth$cpi[1:247], trace = FALSE)
```

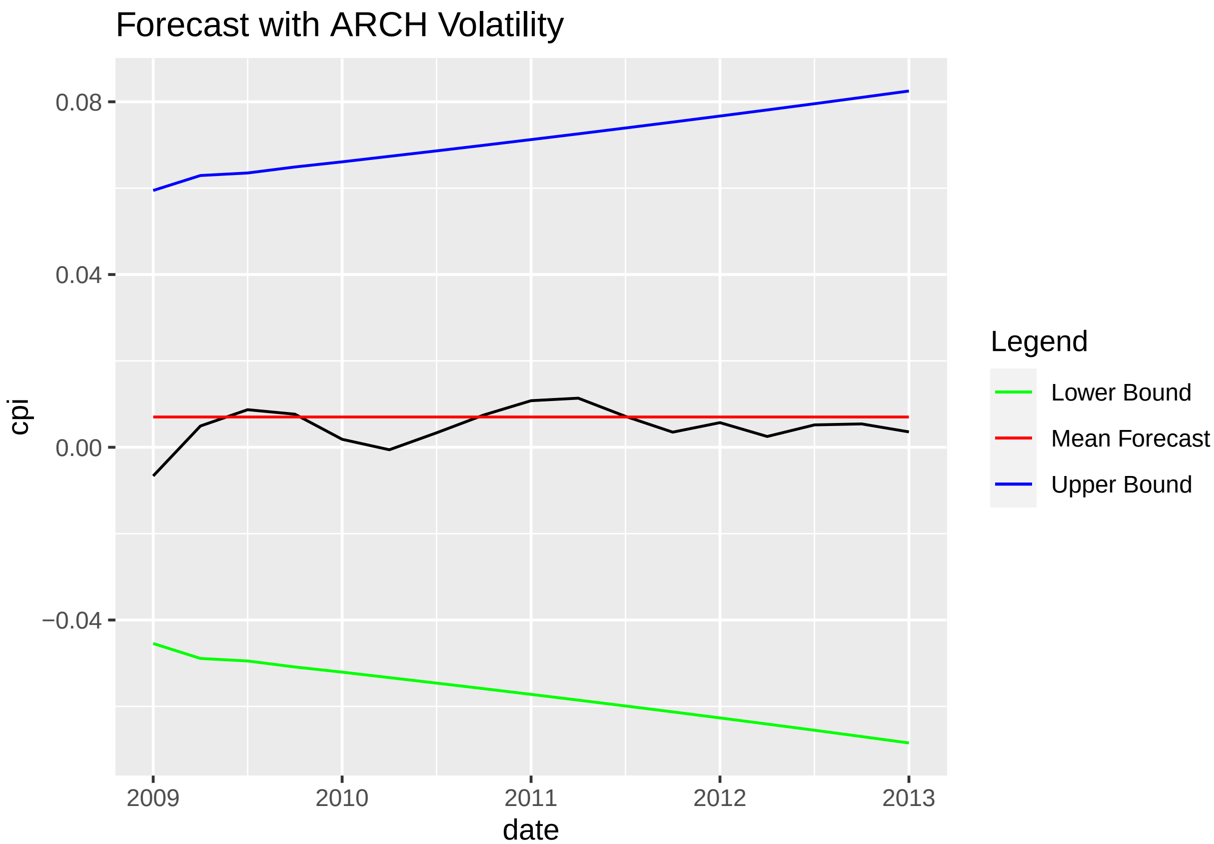
```
### cpi ht = estimated variance
cpi.growth$ht <- arch.fit@h.t

ggplot(cpi.growth, aes(y = ht, x = date)) + geom_line(col = '#ff9933') + ylab('Conditional Variance') +
```




```
predicts <- predict(arch.fit1, n.ahead = 17, plot = TRUE)
```




```r
cols4 <- c("Mean Forecast" = "red", "Lower Bound" = "green", "Upper Bound" = "blue")
ggplot(cpi.growth[248:264,], aes(y = cpi, x = date)) +
  geom_line() +
  geom_line(aes(y=predicts$meanForecast, color = "Mean Forecast")) +
   geom_line(aes(y=predicts$upperInterval, color = "Upper Bound")) +
  geom_line(aes(y=predicts$lowerInterval, color = "Lower Bound")) +
  ggtitle("Forecast with ARCH Volatility") +
  scale_color_manual("Legend", values = cols4)
```

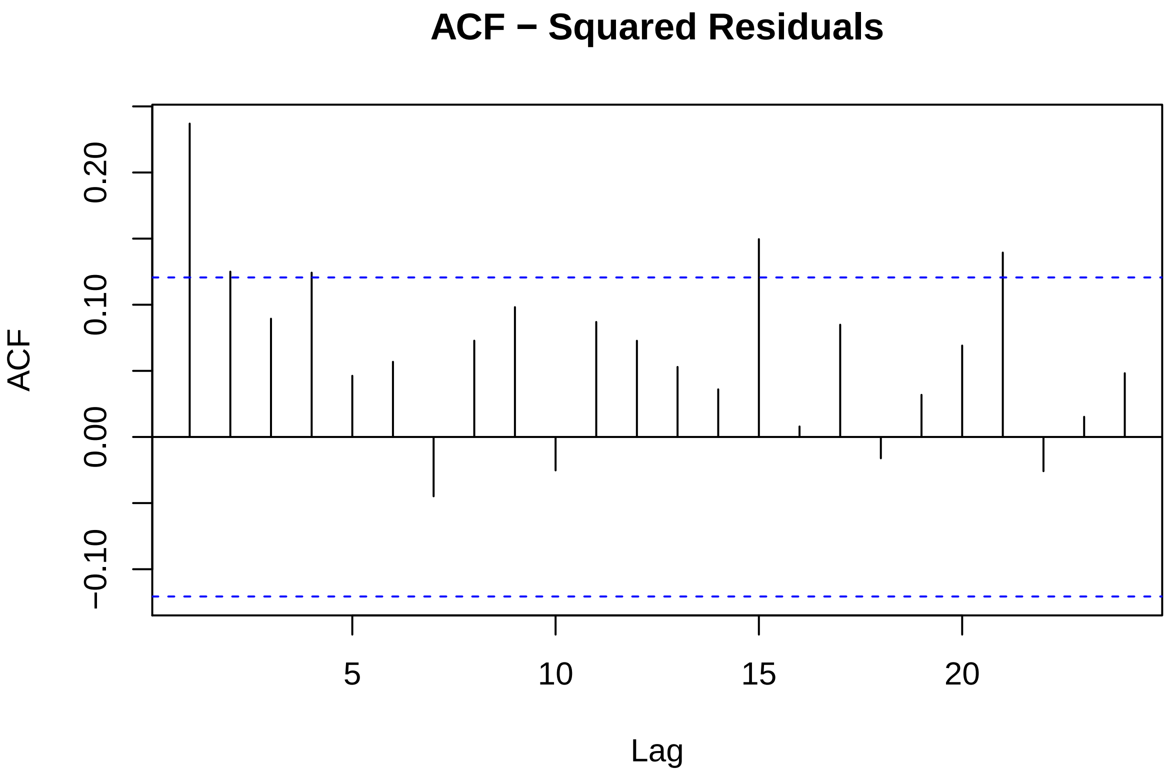


I plotted a graph of the forecast for the out of sample and by looking at the actual value, the forecast does a good job of capturing it and inflation fluctuate arounds its unconditional mean. I used a ARCH(2,0) model.

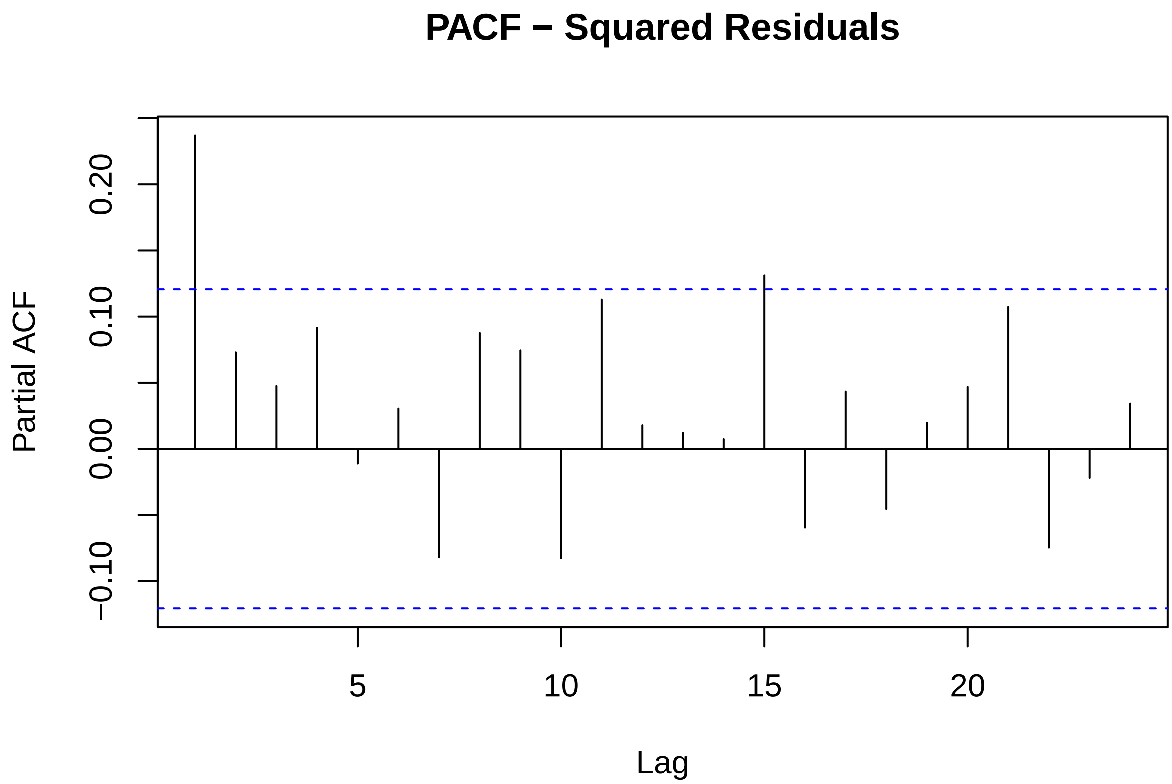
```r
#### for gdp

# Step 1: Estimate mean equation r = beta + error
gdp.mean <- dynlm(gdp ~ 1, data = gdp.growth)

summary(gdp.mean)
```

```
##
## Time series regression with "numeric" data:
## Start = 1, End = 264
##
## Call:
## dynlm(formula = gdp ~ 1, data = gdp.growth)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.035212 -0.004990 -0.000181  0.005258  0.031850
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.0077639  0.0006054   12.82   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.009837 on 263 degrees of freedom
```

```r
# Step 2: Retrieve the residuals from the former model and square them
ehatsqGDP <- ts(resid(gdp.mean)^2)
acf(ehatsqGDP, main = "ACF - Squared Residuals")
```




```r
pacf(ehatsqGDP, main = "PACF - Squared Residuals")
```




```r
# Step 3: regress squared residuals on one-lagged squared residuals
GDP.arch <- dynlm(ehatsqGDP ~ L(ehatsqGDP), data = ehatsqGDP)

summary(GDP.arch)
```

```
##
## Time series regression with "ts" data:
## Start = 2, End = 264
##
## Call:
## dynlm(formula = ehatsqGDP ~ L(ehatsqGDP), data = ehatsqGDP)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -3.650e-04 -7.438e-05 -6.103e-05  1.481e-05  1.087e-03
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.349e-05  1.221e-05   6.018 5.94e-09 ***
## L(ehatsqGDP) 2.372e-01  6.016e-02   3.943 0.000103 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0001741 on 261 degrees of freedom
## Multiple R-squared:  0.05622,    Adjusted R-squared:  0.05261
## F-statistic: 15.55 on 1 and 261 DF,  p-value: 0.0001034
```

```r
## check ARCH effect
gdp.archTest <- ArchTest(gdp.growth, lags = 1, demean = TRUE)
gdp.archTest
```

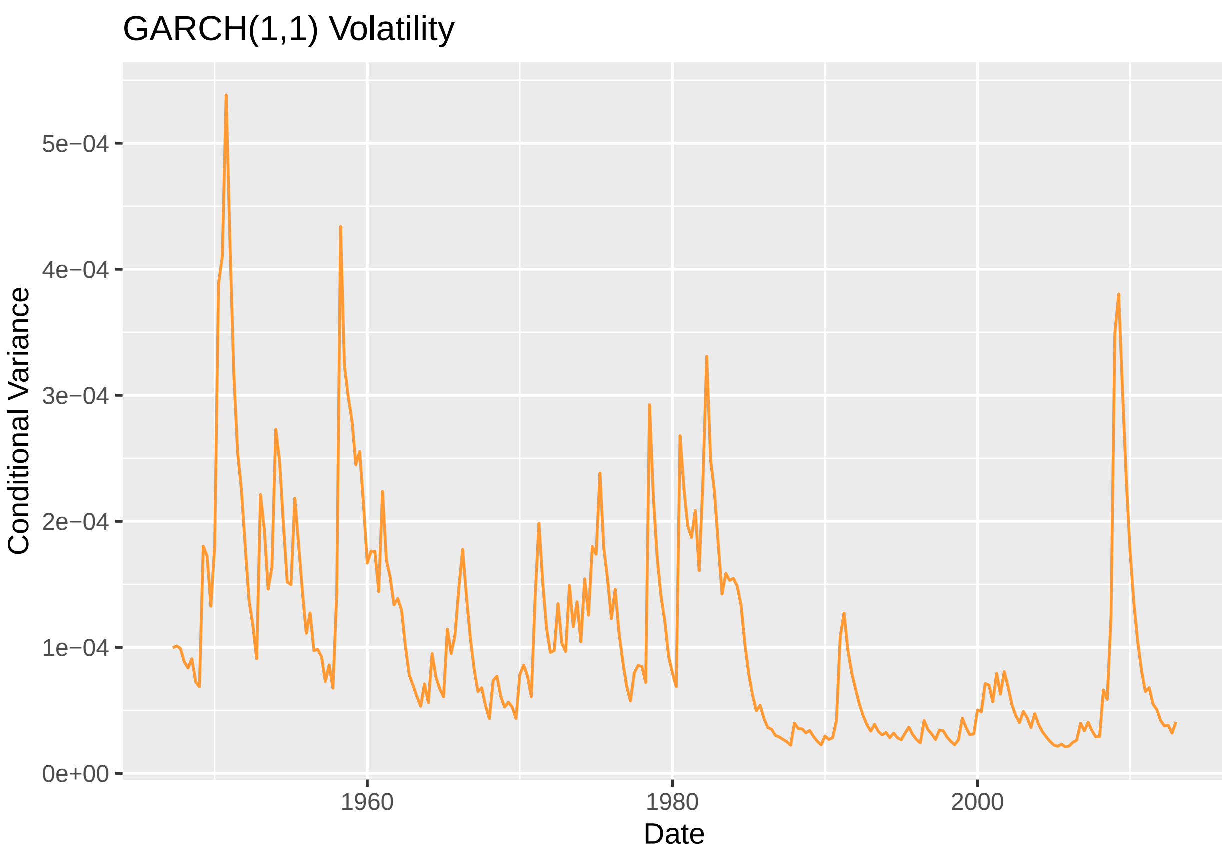
```
##
##  ARCH LM-test; Null hypothesis: no ARCH effects
##
## data:  gdp.growth
## Chi-squared = 262.8, df = 1, p-value < 2.2e-16
```

```r
# low p-value so conclude presence of ARCH 1 effects

### estimate ARCH model
garch.fit <- garchFit(~garch(1,1), data = gdp.growth$gdp, trace = FALSE)
garch.fit1 <- garchFit(~garch(1,1), data = gdp.growth$gdp[1:247], trace = FALSE)

summary(garch.fit)
```

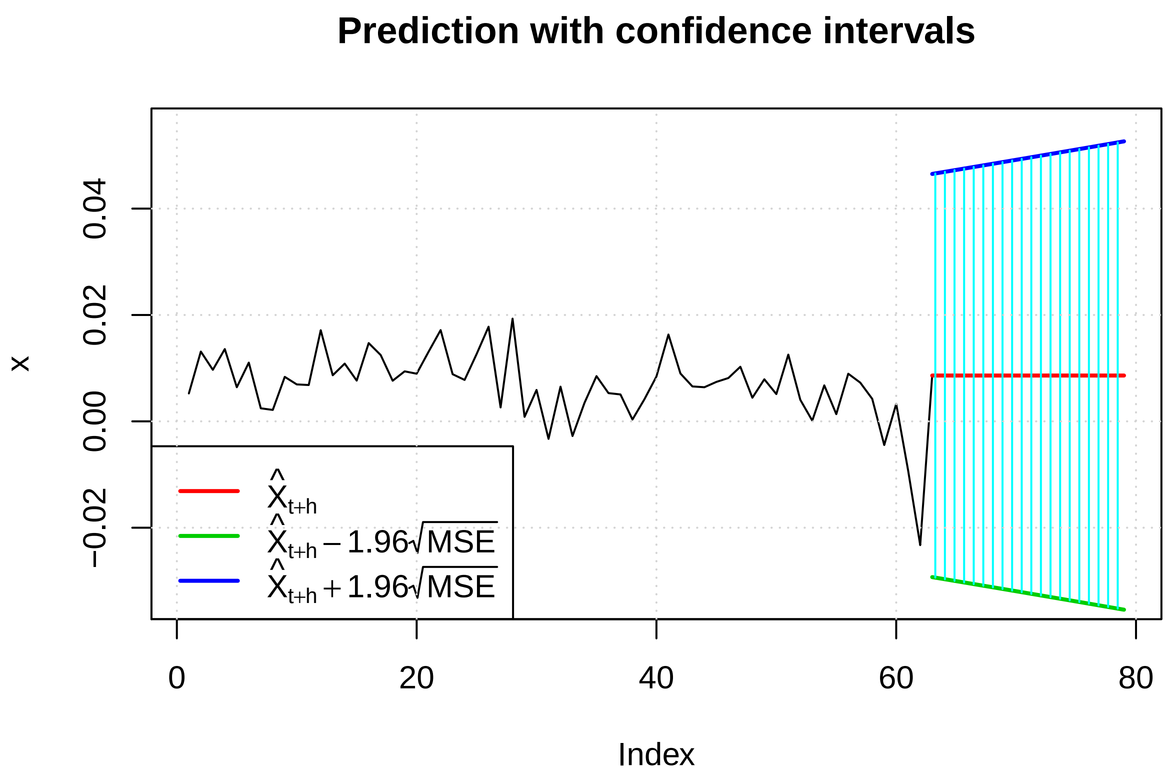
```
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~garch(1, 1), data = gdp.growth$gdp, trace = FALSE)
##
## Mean and Variance Equation:
##  data ~ garch(1, 1)
## <environment: 0x7fbf66008e40>
##  [data = gdp.growth$gdp]
##
## Conditional Distribution:
##  norm
##
## Coefficient(s):
##         mu       omega      alpha1       beta1
## 8.2055e-03  3.8571e-06  2.5533e-01  7.3471e-01
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu     8.206e-03   5.567e-04   14.739  < 2e-16 ***
## omega  3.857e-06   2.220e-06    1.737  0.08236 .
## alpha1 2.553e-01   7.497e-02    3.406  0.00066 ***
## beta1  7.347e-01   6.071e-02   12.103  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  868.9006    normalized:  3.29129
##
## Description:
##  Tue Jun  2 09:51:57 2020 by user:
##
##
## Standardised Residuals Tests:
##                                 Statistic p-Value
##  Jarque-Bera Test   R    Chi^2  12.06716  0.002396894
```

```
##   Shapiro-Wilk Test  R    W      0.9871315 0.01834881
##   Ljung-Box Test     R    Q(10)  38.45559  3.160947e-05
##   Ljung-Box Test     R    Q(15)  45.38053  6.667111e-05
##   Ljung-Box Test     R    Q(20)  52.31402  0.0001024523
##   Ljung-Box Test     R^2  Q(10)  12.54281  0.250368
##   Ljung-Box Test     R^2  Q(15)  15.56407  0.4115998
##   Ljung-Box Test     R^2  Q(20)  19.5429   0.4868298
##   LM Arch Test       R    TR^2   11.28139  0.5049703
##
## Information Criterion Statistics:
##       AIC       BIC       SIC      HQIC
## -6.552277 -6.498096 -6.552727 -6.530506
```

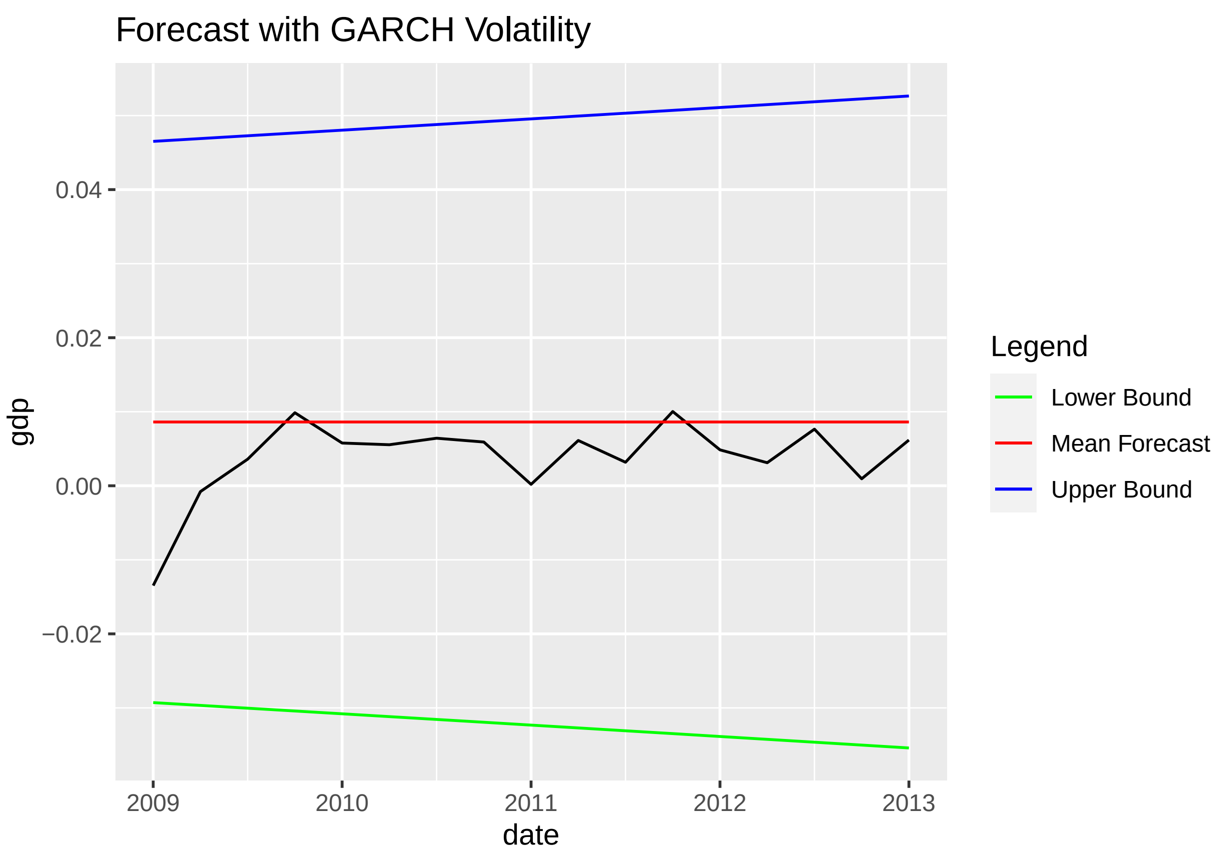
```r
### gdp ht = estimated variance
gdp.growth$ht <- garch.fit@h.t

ggplot(gdp.growth, aes(y = ht, x = date)) + geom_line(col = '#ff9933') + ylab('Conditional Variance') +
```



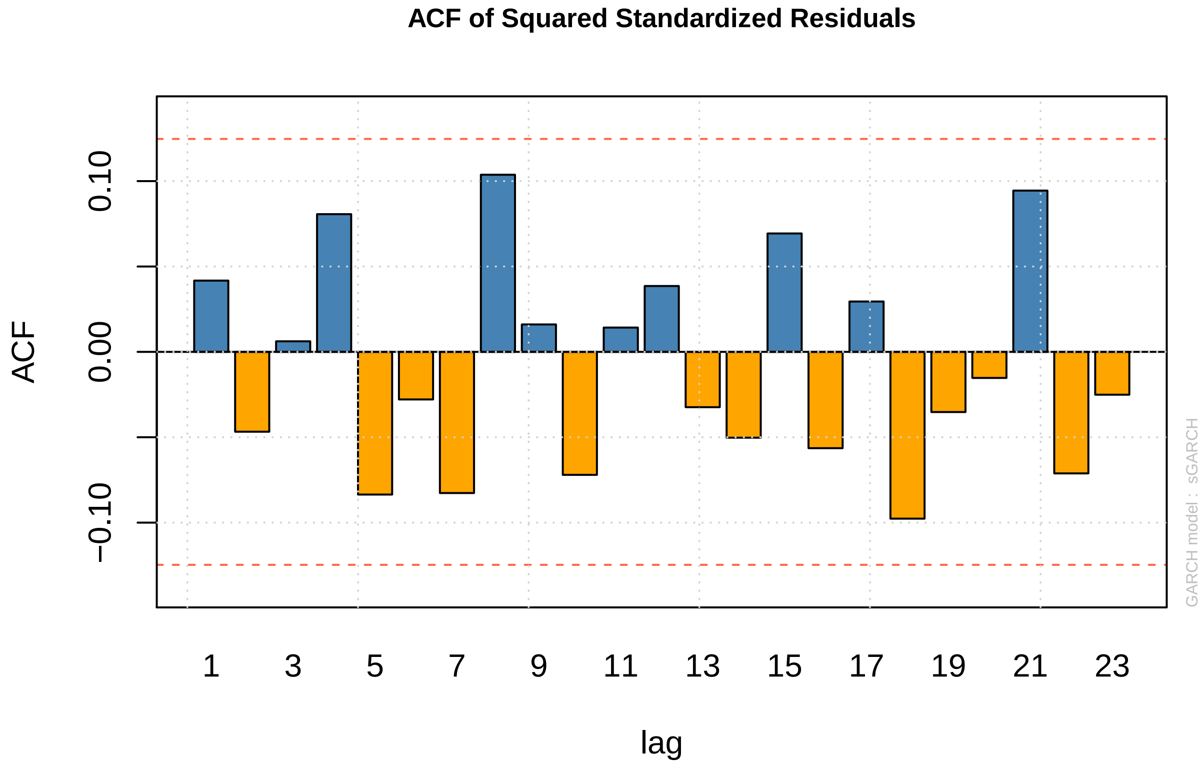

```r
predictsgdp <- predict(garch.fit1, n.ahead = 17, plot = TRUE)
```



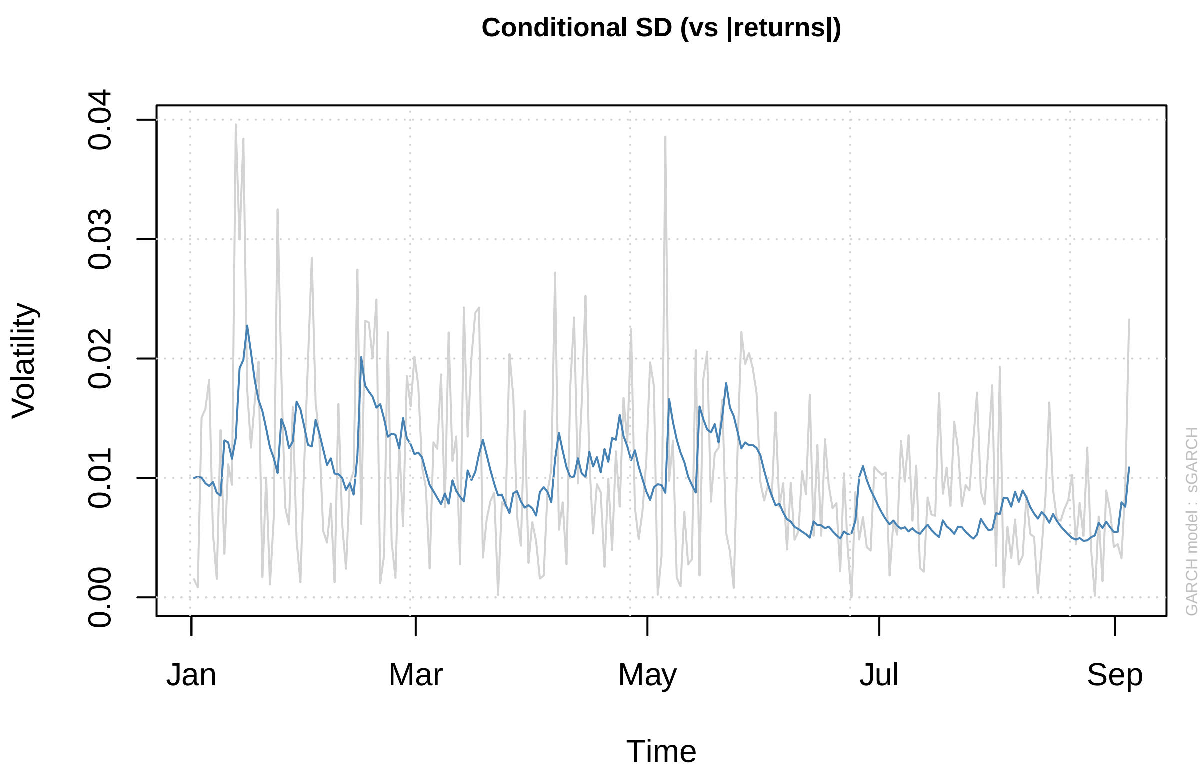

```
cols5 <- c("Mean Forecast" = "red", "Lower Bound" = "green", "Upper Bound" = "blue")
ggplot(gdp.growth[248:264,], aes(y = gdp, x = date)) +
  geom_line() +
  geom_line(aes(y=predictsgdp$meanForecast, color = "Mean Forecast")) +
   geom_line(aes(y=predictsgdp$upperInterval, color = "Upper Bound")) +
  geom_line(aes(y=predictsgdp$lowerInterval, color = "Lower Bound")) +
  ggtitle("Forecast with GARCH Volatility") +
  scale_color_manual("Legend", values = cols5)
```



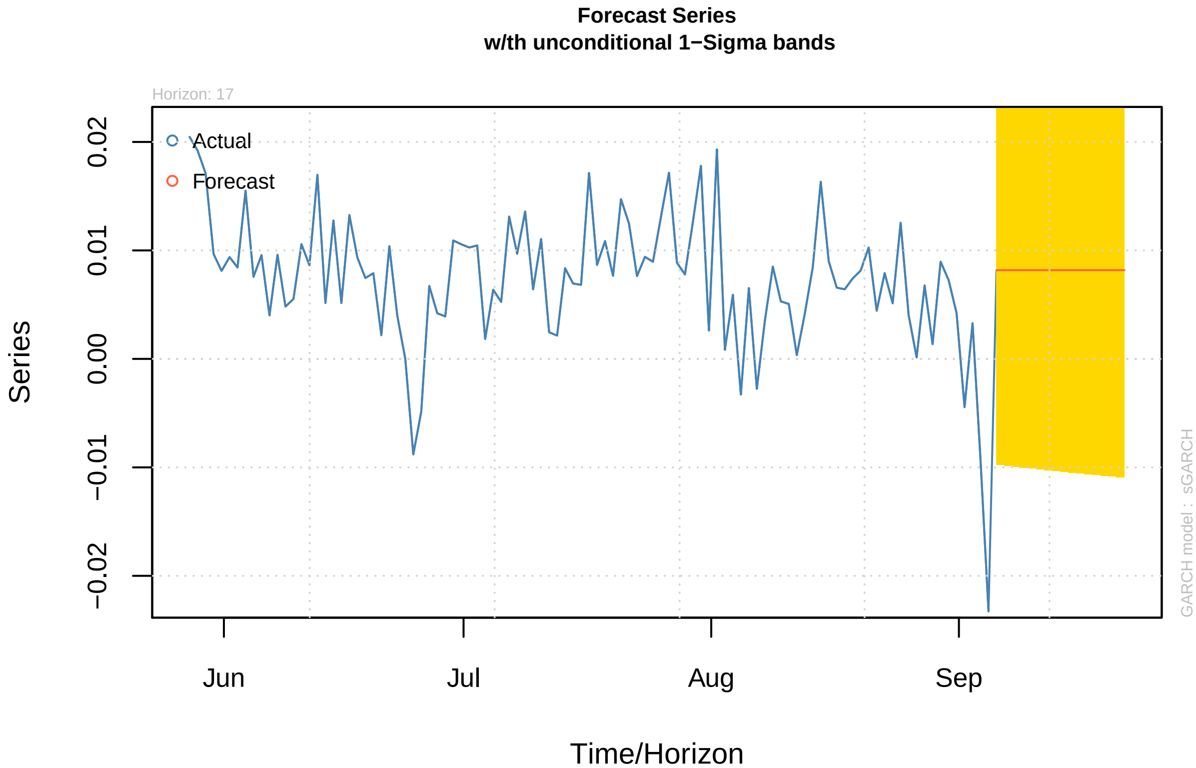

```
modelGDP144 = ugarchspec( variance.model = list(model = "sGARCH", garchOrder = c(1, 1)), mean.model = li

modelfitGDP144 = ugarchfit(spec = modelGDP144, data = gdp.growth$gdp[1:247])
#modelfitGDP
plot(modelfitGDP144, which = 11)
```



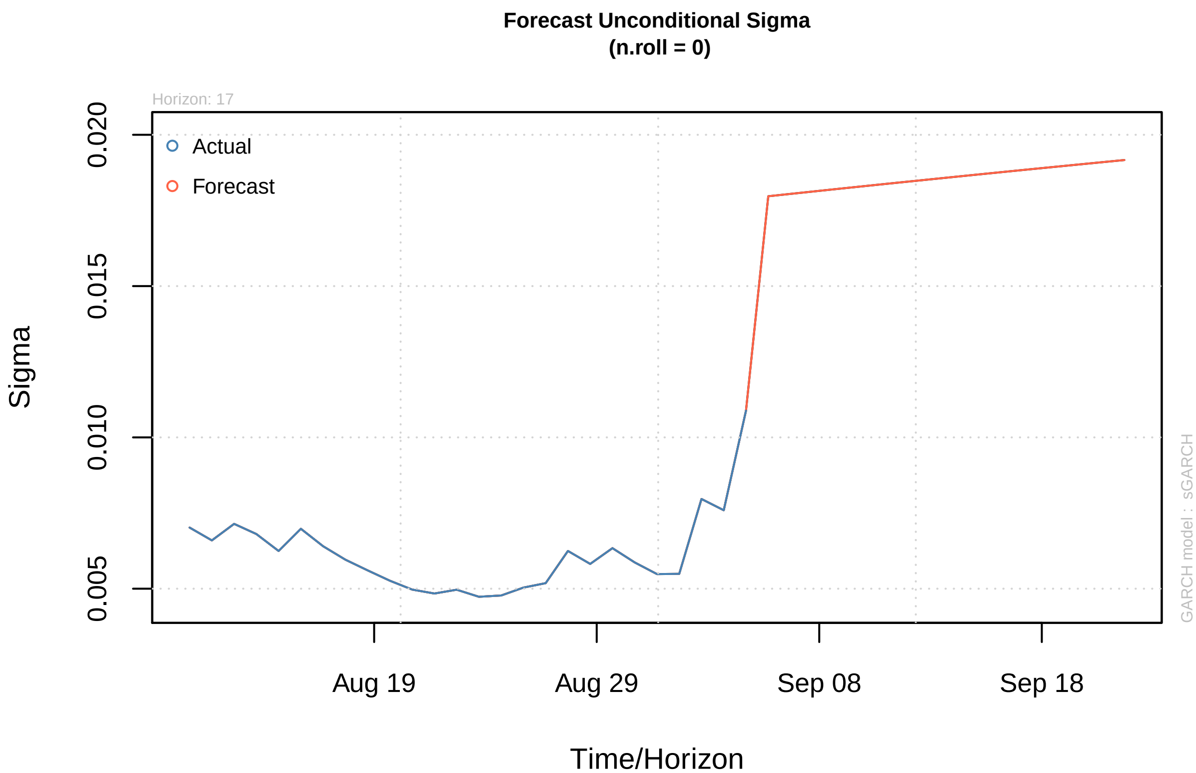

```
plot(modelfitGDP144, which = 3)
```




```
modelforGDP144 = ugarchforecast(modelfitGDP144, data = gdp.growth$gdp, n.ahead = 17, n.roll = 0, out.sa

plot(modelforGDP144,  which = 1 )
```



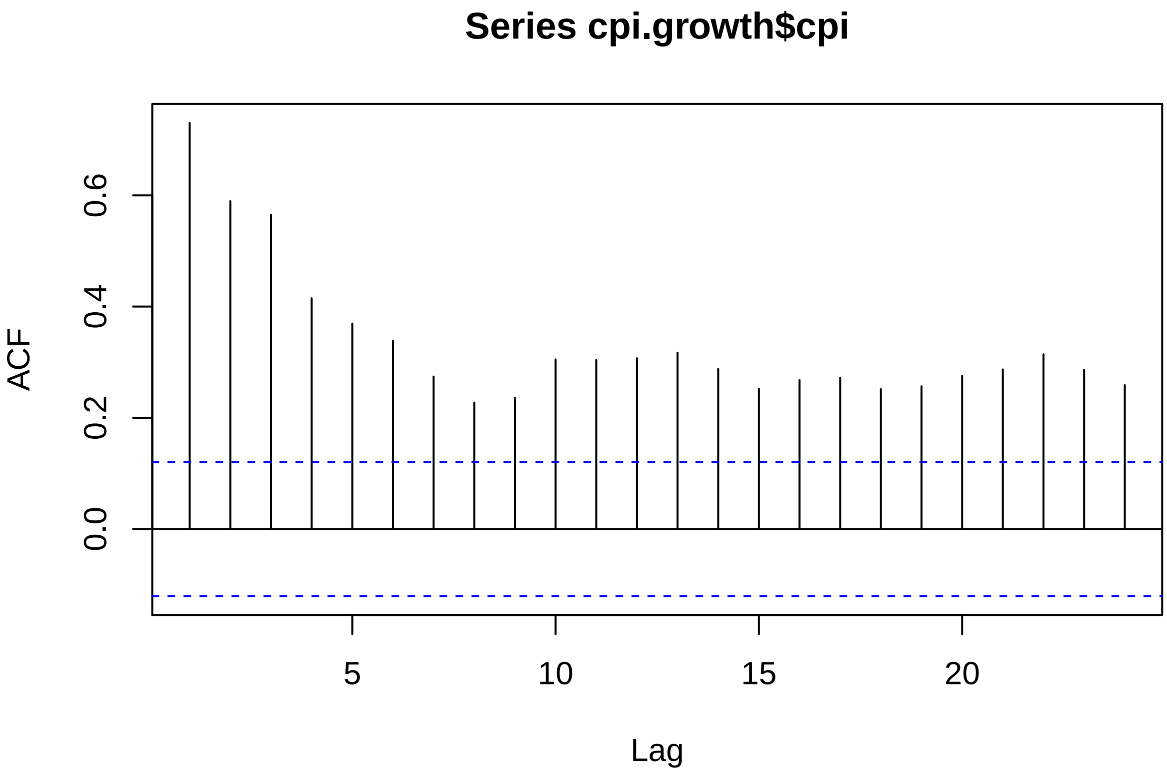

```r
plot(modelforGDP144,  which = 3 )
```



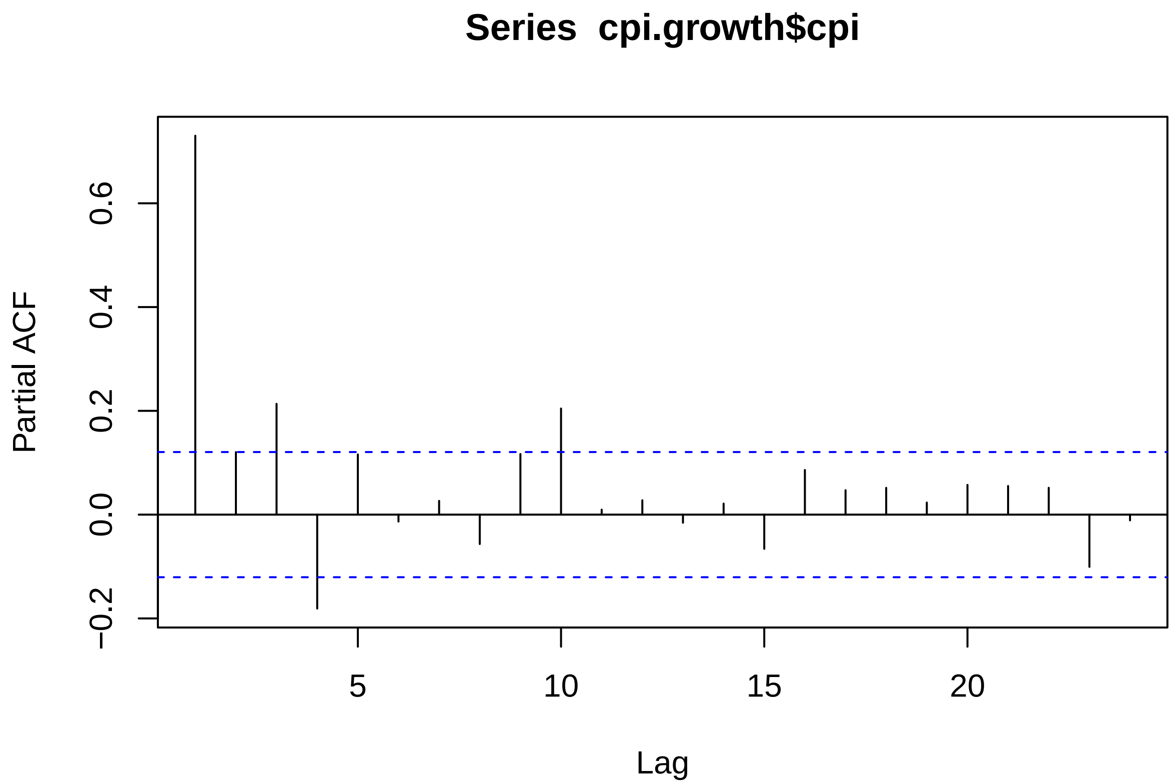
GDP fluctuates around its unconditional mean so this forecast does a good job of forecasting volatility. I used a GARCH(1,1) model.

Problem 14.6:

```
### CPI Conditional Mean and Conditional Variance

acf(cpi.growth$cpi)
```



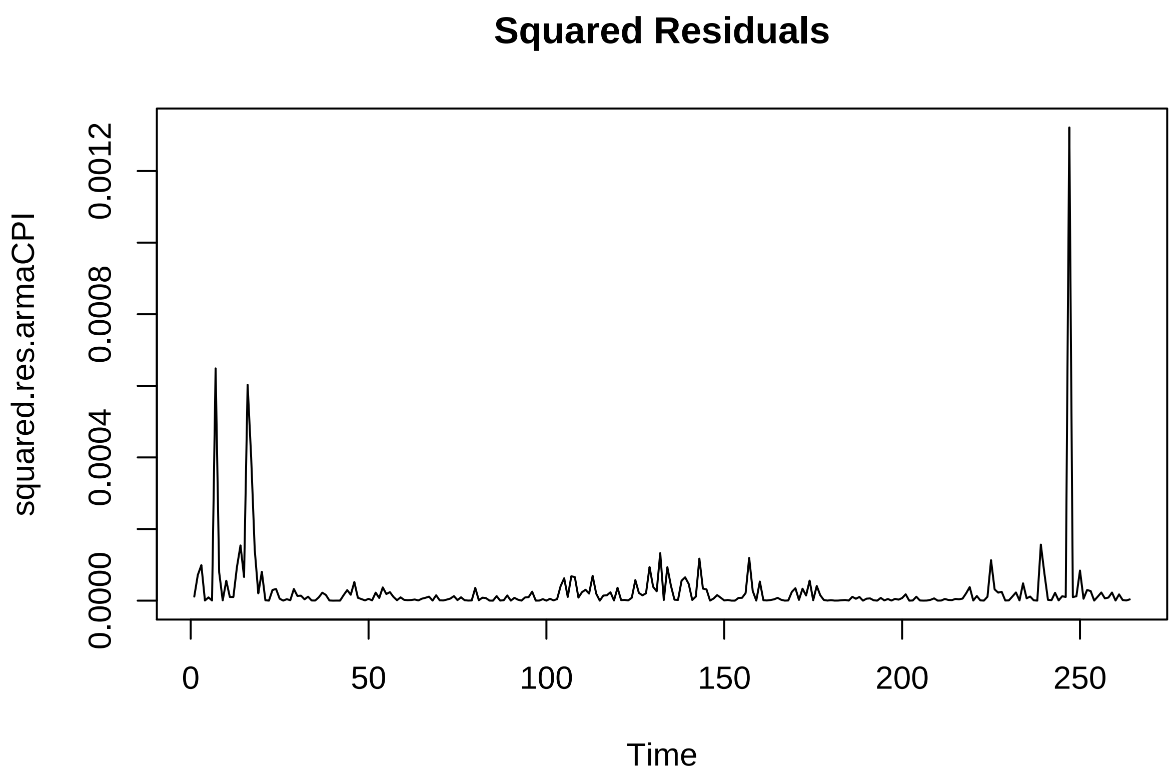

```
pacf(cpi.growth$cpi)
```



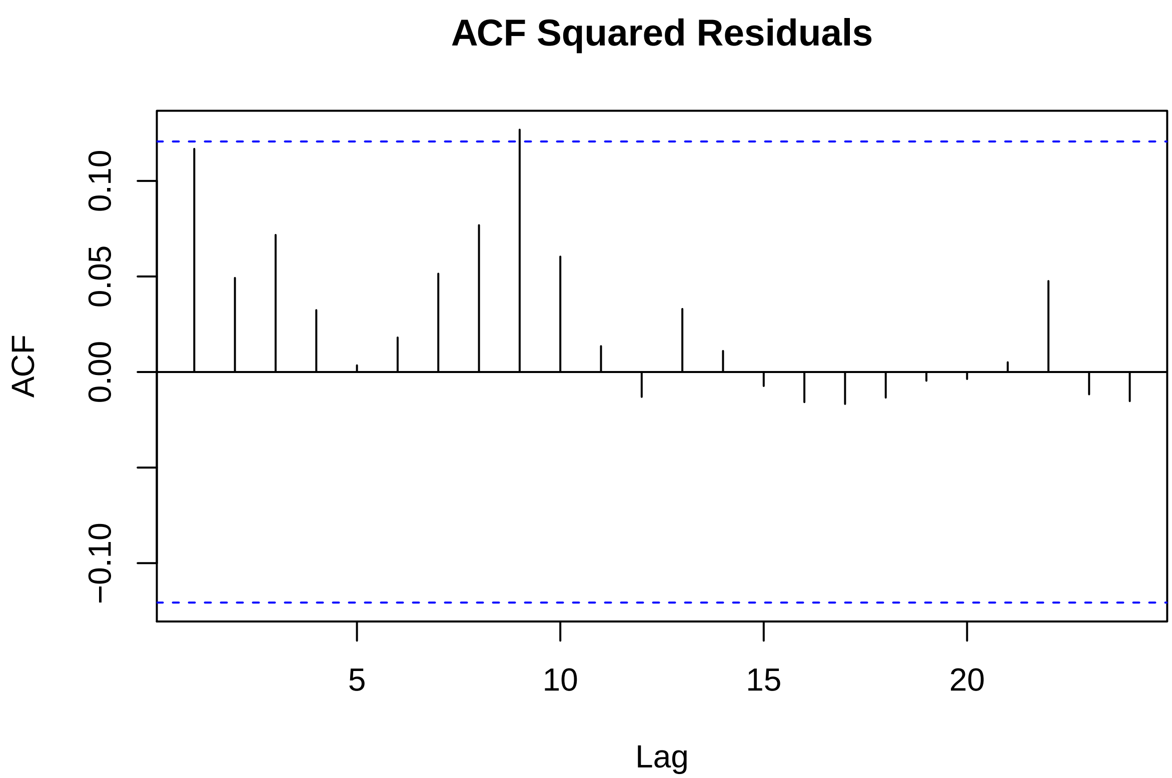

```r
auto.arima(cpi.growth$cpi)
```

```
## Series: cpi.growth$cpi
## ARIMA(4,1,1)
##
## Coefficients:
##          ar1      ar2     ar3      ar4      ma1
##       0.5771  -0.0559  0.2922  -0.2320  -0.9170
## s.e.  0.0684   0.0689  0.0690   0.0627   0.0388
##
## sigma^2 estimated as 2.817e-05:  log likelihood=1006.62
## AIC=-2001.23   AICc=-2000.9   BIC=-1979.8
```

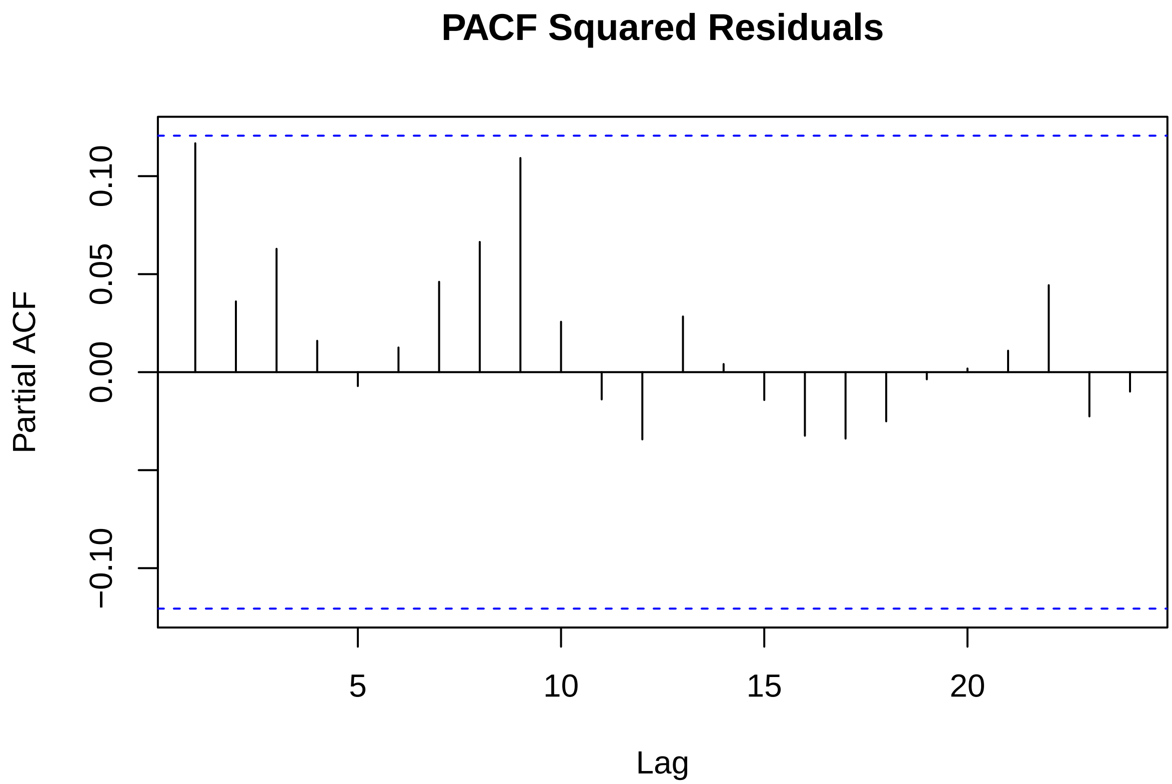
```r
armaCPI <- arima(cpi.growth$cpi, order = c(3,0,0)) # AR 3 model

res.armaCPI <- armaCPI$res

squared.res.armaCPI <- res.armaCPI^2


plot(squared.res.armaCPI, main = 'Squared Residuals')
```



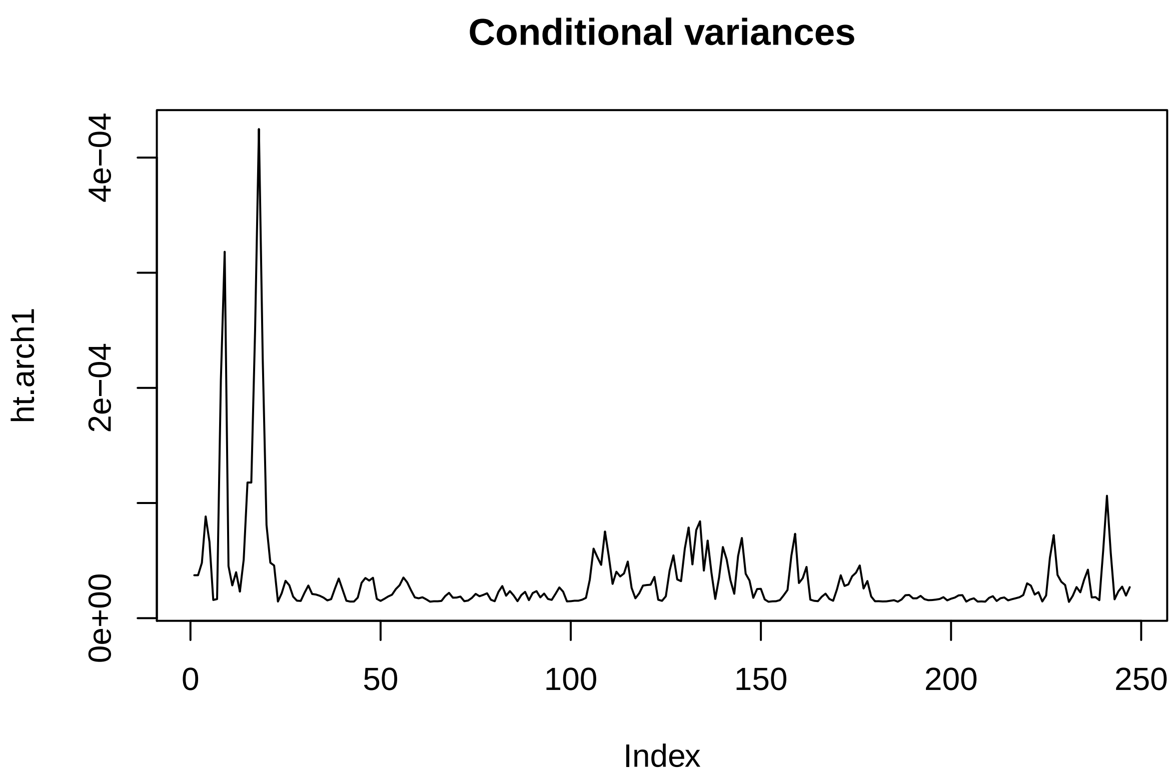

```
acf.squaredcpi <- acf(squared.res.armaCPI, main = 'ACF Squared Residuals')
```



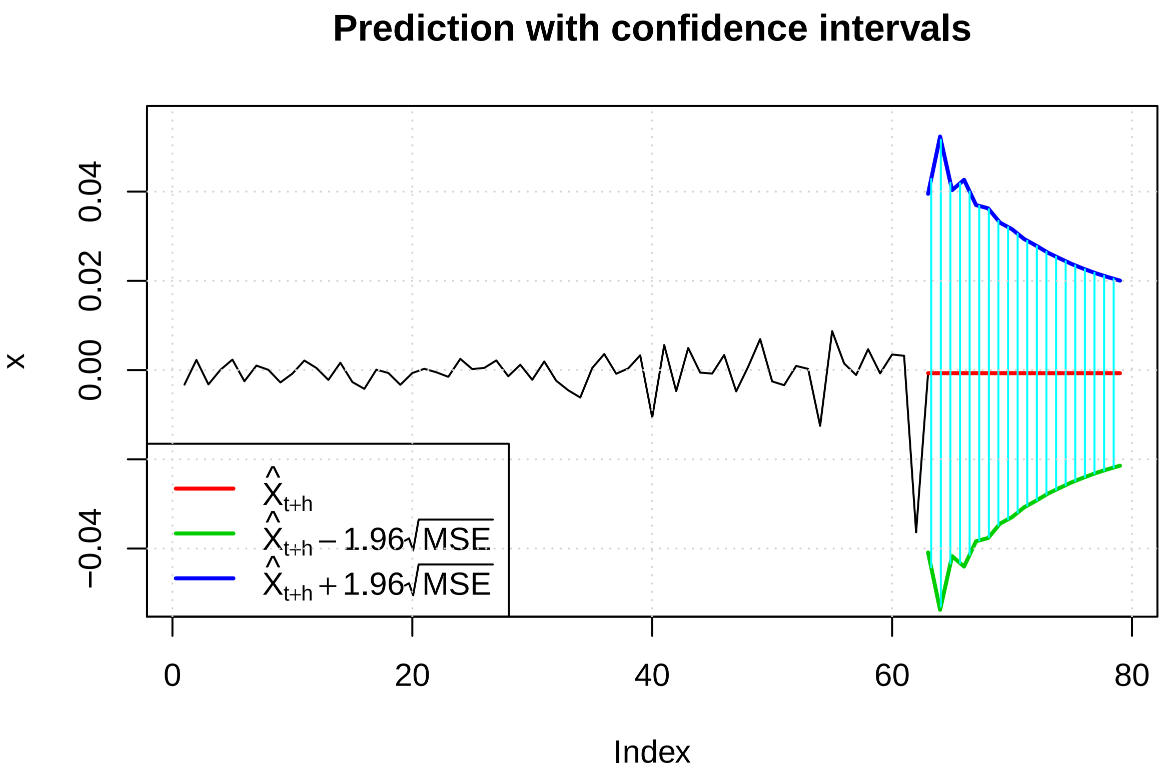

```
pacf.squaredcpi <- pacf(squared.res.armaCPI,main='PACF Squared Residuals')
```




```
arch1 <- garchFit(~garch(2,0), data = res.armaCPI[1:247], trace = FALSE)
#summary(arch1)
ht.arch1 <- arch1@h.t
plot(ht.arch1, type = "l", main='Conditional variances')
```

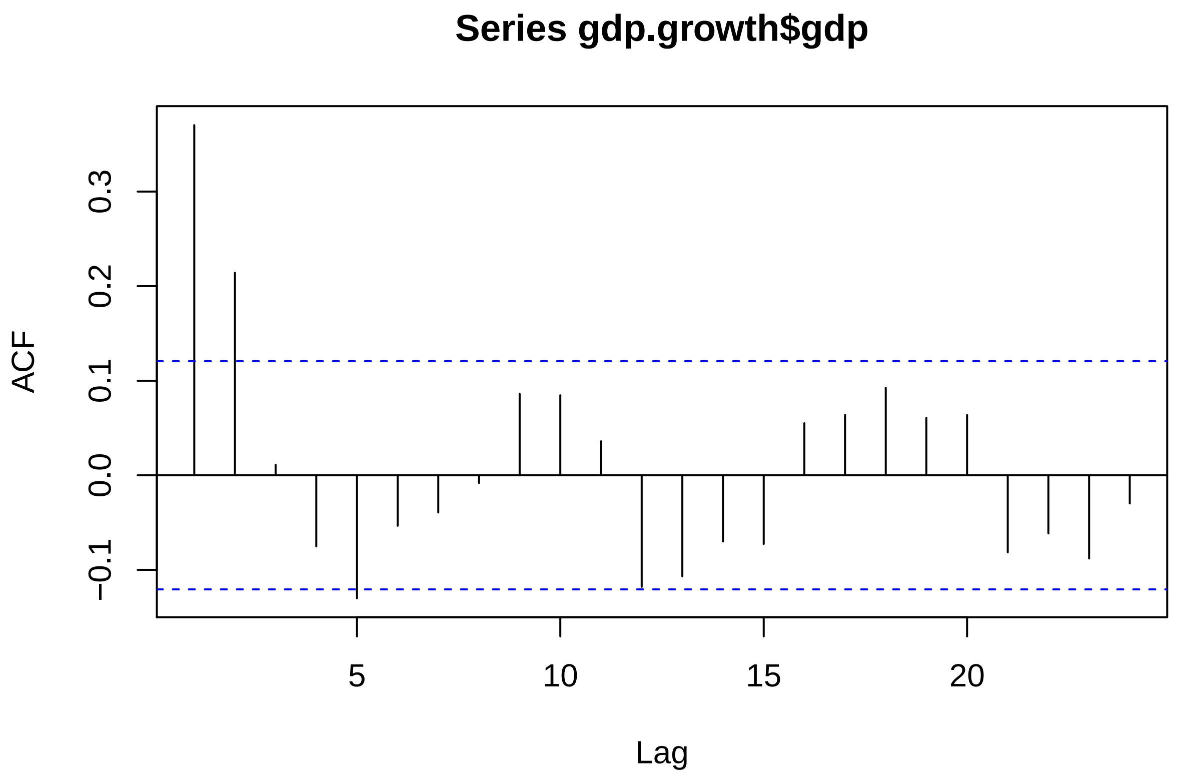


```r
predict.cpi.arma <- predict(arch1, n.ahead = 17, plot = TRUE)
```

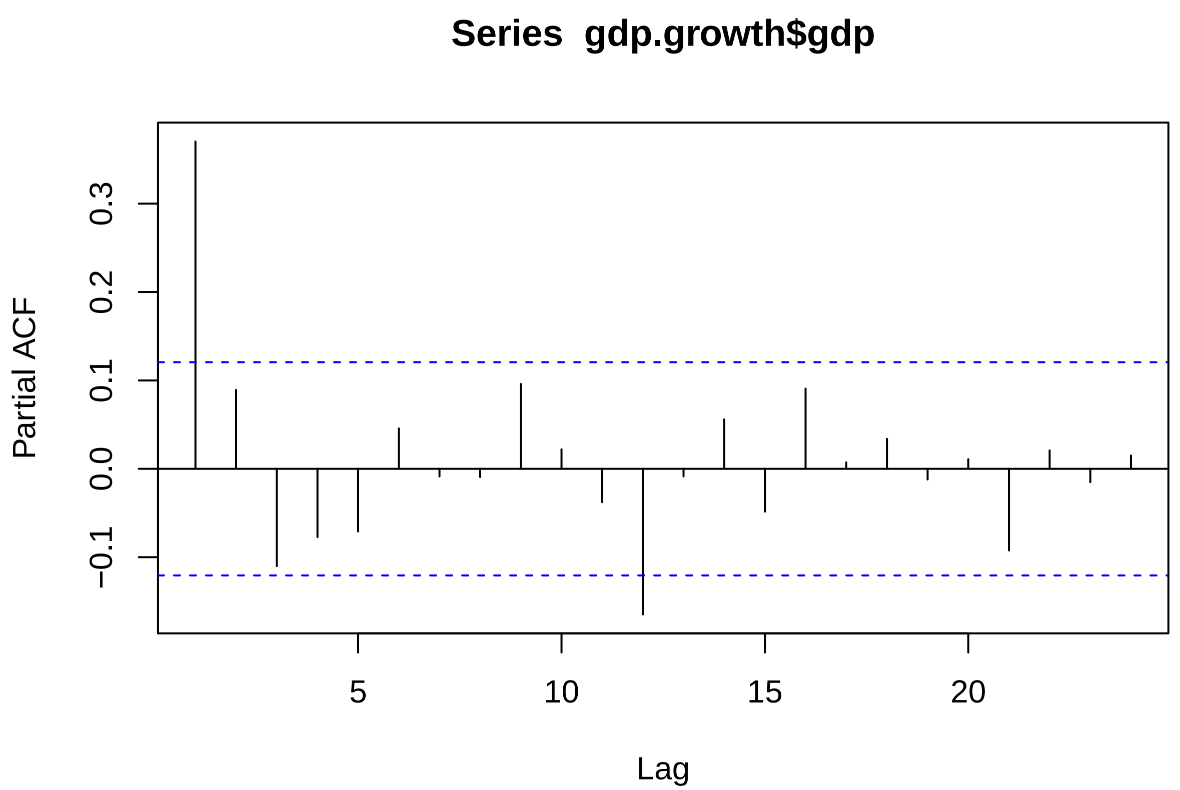


With a ARMA(2,0) and a ARCH(2) process, the residuals display white noise and with the forecast, and the 95% bands are narrower and the volatility is smaller than the forecast in 14.5.

```r
### GDP Conditional Mean and Conditional Variance

acf(gdp.growth$gdp)
```




```r
pacf(gdp.growth$gdp)
```

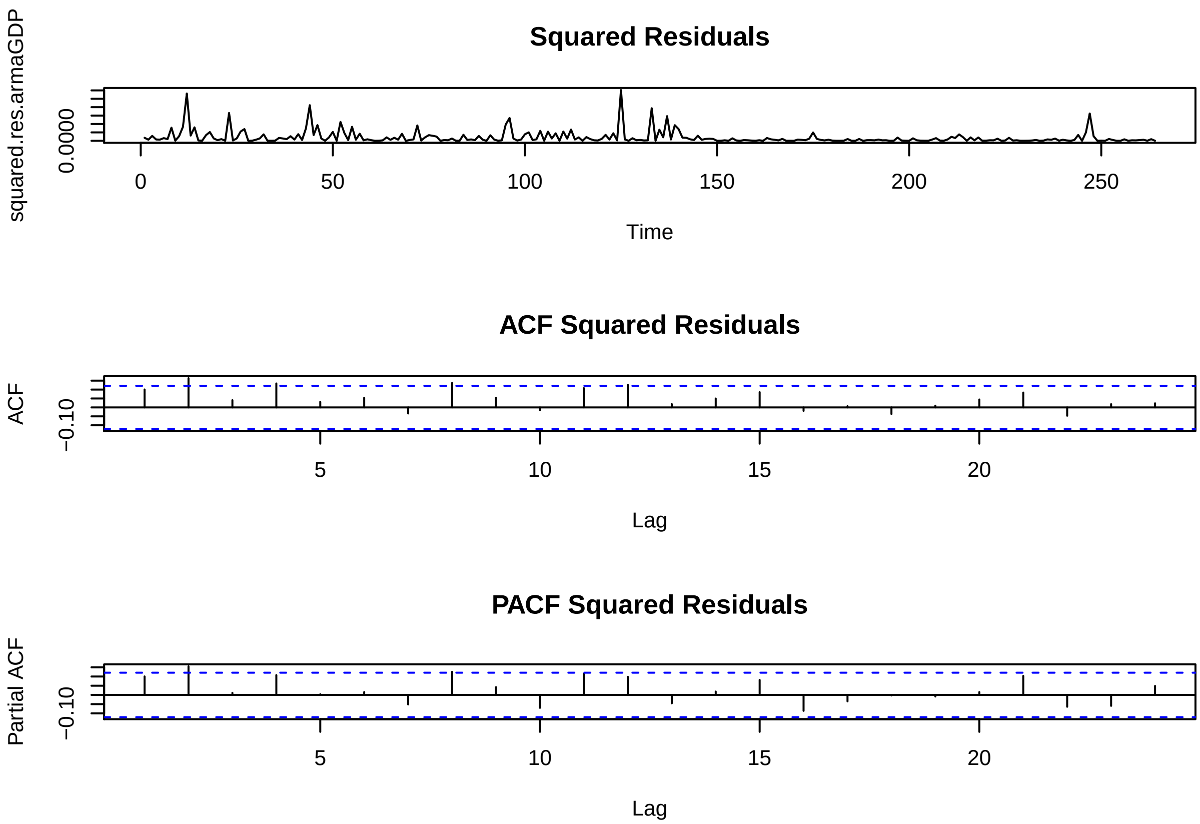



```r
auto.arima(gdp.growth$gdp)
```

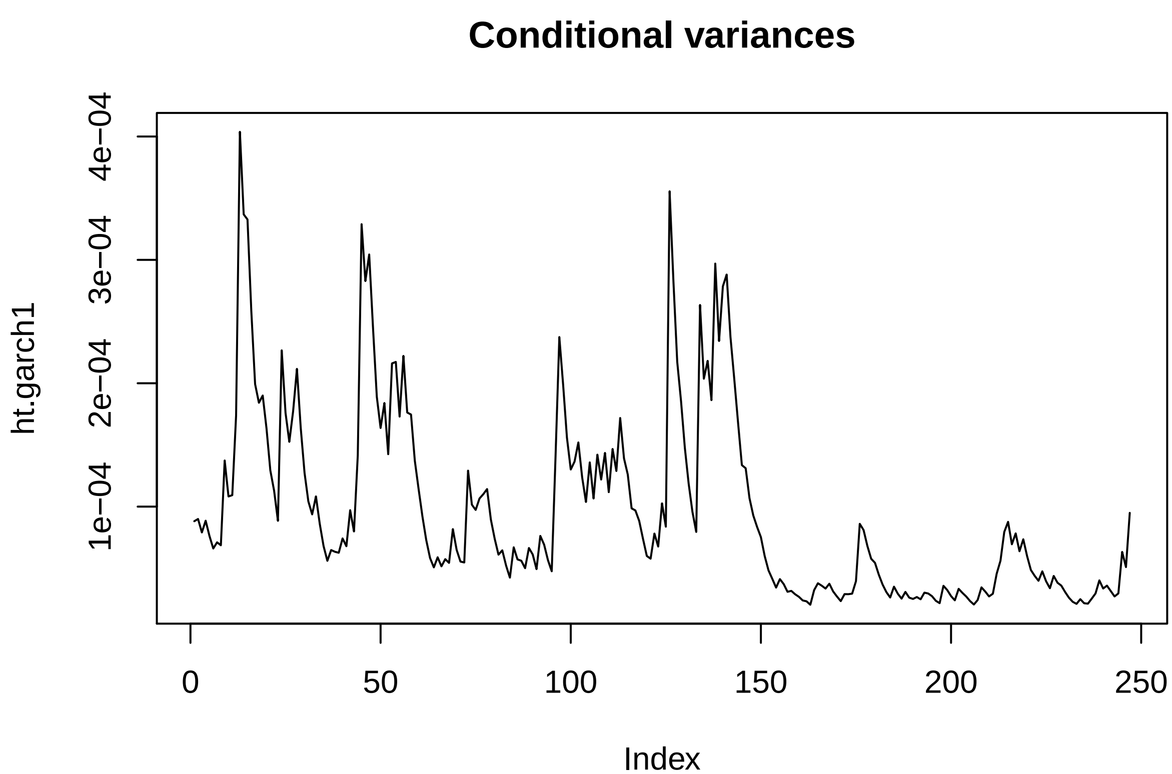
```
## Series: gdp.growth$gdp
```

```
## ARIMA(2,0,2) with non-zero mean
##
## Coefficients:
##          ar1      ar2      ma1     ma2    mean
##       1.2565  -0.6507  -0.9354  0.4765  0.0077
## s.e.  0.1748   0.1586   0.2074  0.1639  0.0008
##
## sigma^2 estimated as 8.201e-05:  log likelihood=869.73
## AIC=-1727.45   AICc=-1727.13   BIC=-1706
```

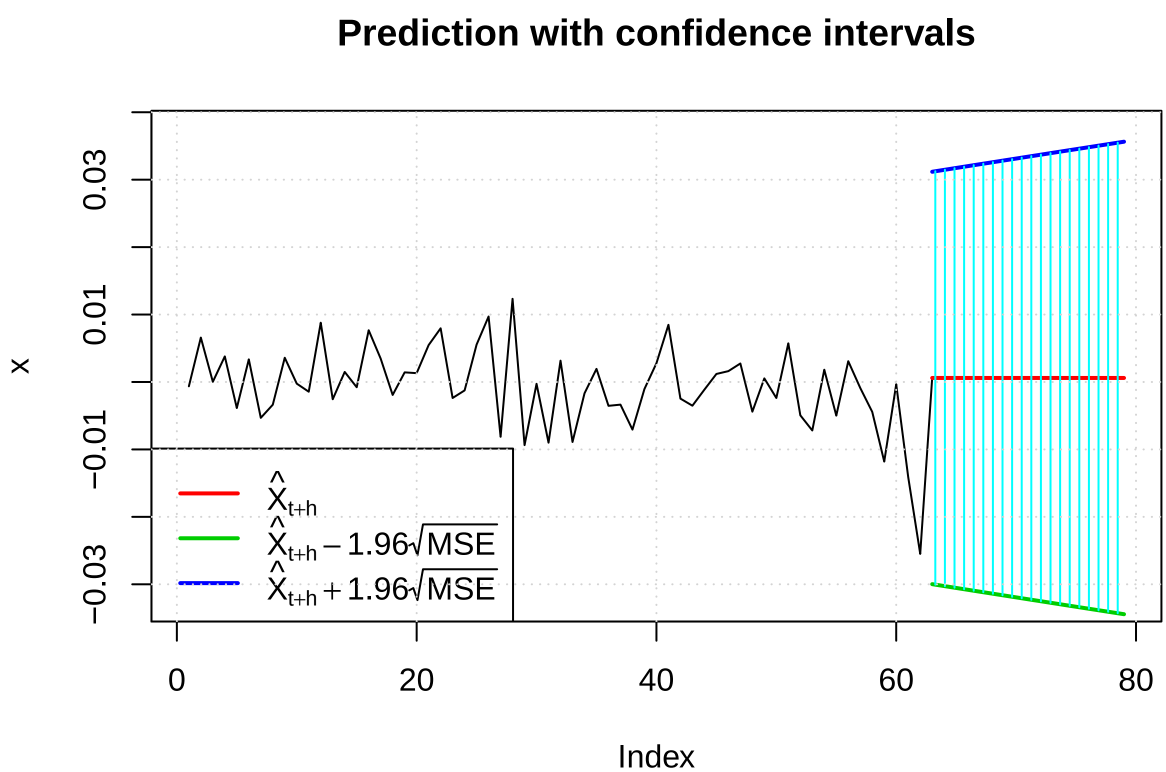
```r
armaGDP <- arima(gdp.growth$gdp, order = c(2,0,2)) # AR 3 model

res.armaGDP <- armaGDP$res
squared.res.armaGDP <- res.armaGDP^2

par(mfcol=c(3,1))
plot(squared.res.armaGDP, main = 'Squared Residuals')
acf.squaredgdp <- acf(squared.res.armaGDP, main = 'ACF Squared Residuals')
pacf.squaredgdp <- pacf(squared.res.armaGDP,main='PACF Squared Residuals')
```




```r
garch1 <- garchFit(~garch(1,1), data = res.armaGDP[1:247], trace = FALSE)
#summary(arch1)

ht.garch1 <- garch1@h.t
plot(ht.garch1, type = "l", main='Conditional variances')
```

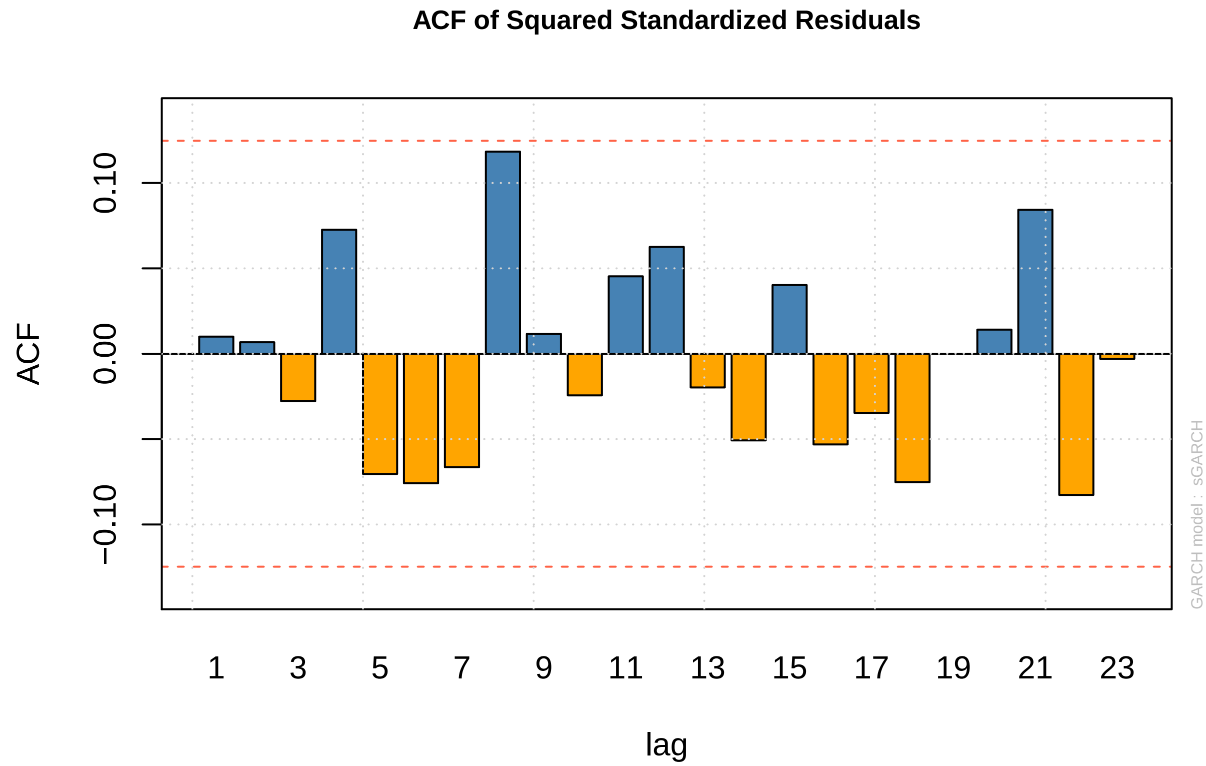



```
predict.gdp.arma <- predict(garch1, n.ahead = 17, plot = TRUE)
```

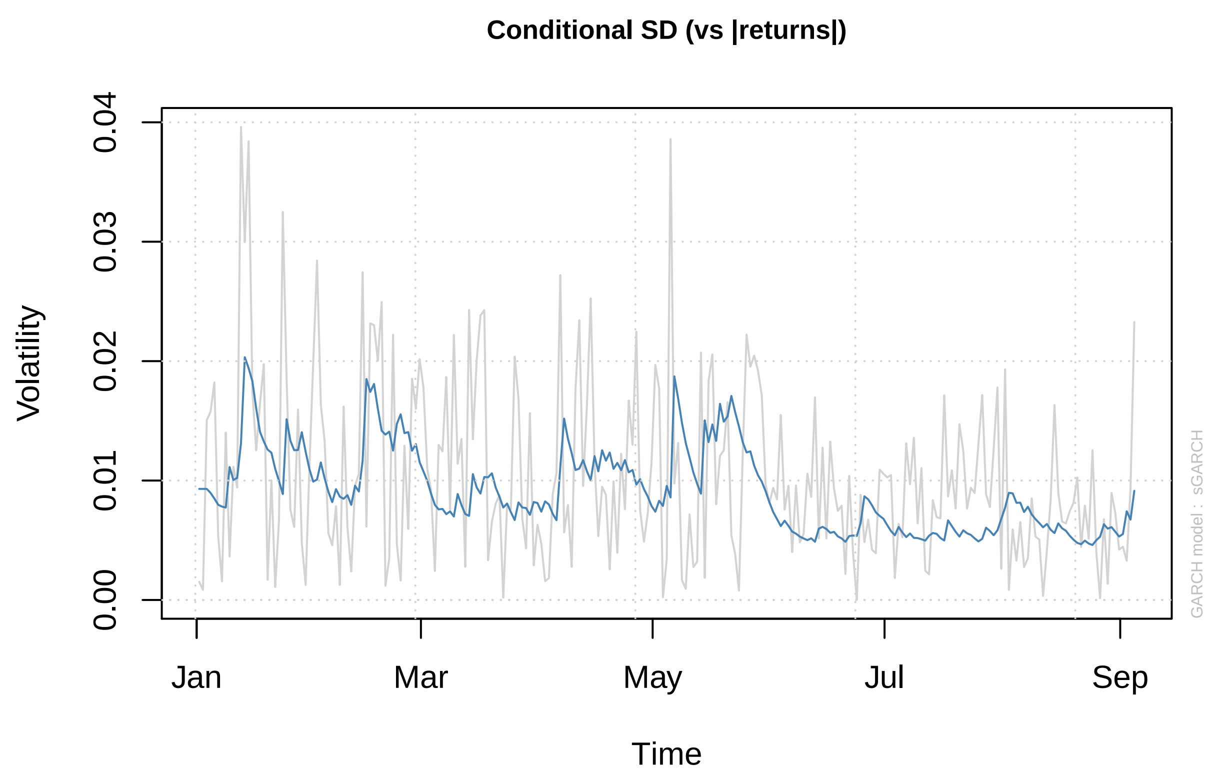



```
modelGDP = ugarchspec( variance.model = list(model = "sGARCH", garchOrder = c(1, 1)), mean.model = list

modelfitGDP = ugarchfit(spec = modelGDP, data = gdp.growth$gdp[1:247])
#modelfitGDP
```

```
plot(modelfitGDP, which = 11)
```

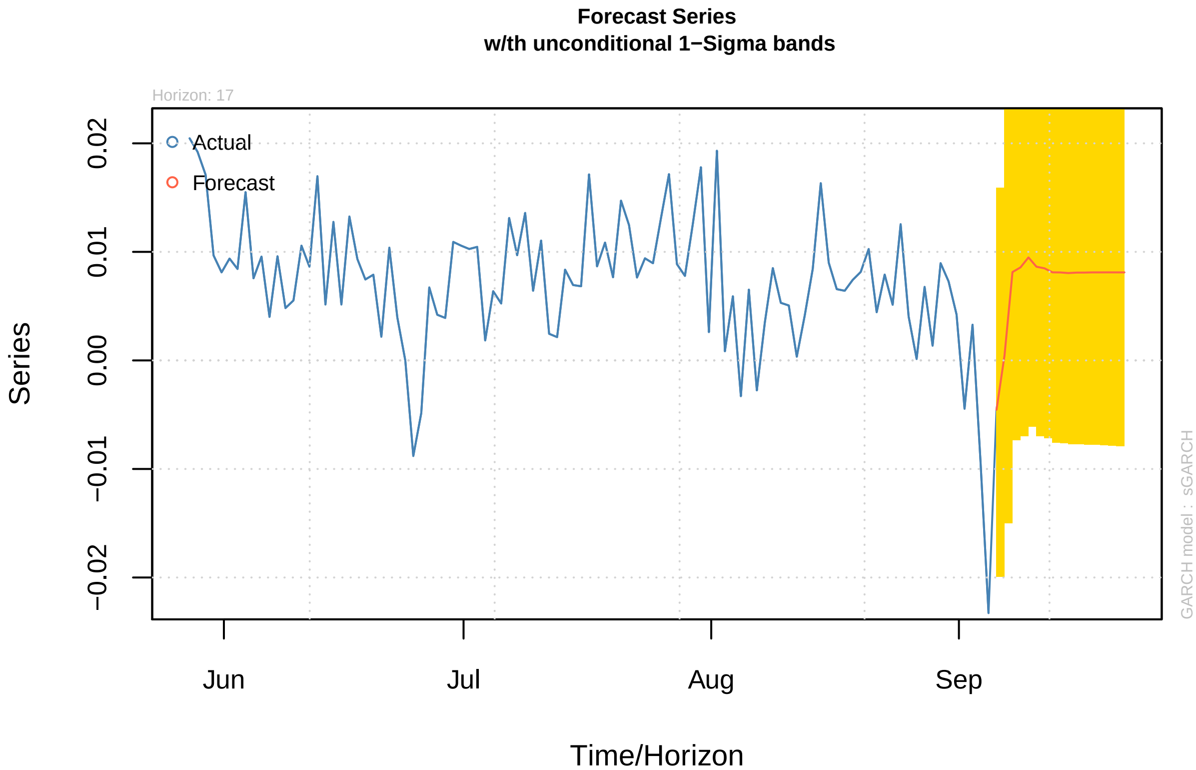



```
plot(modelfitGDP, which = 3)
```

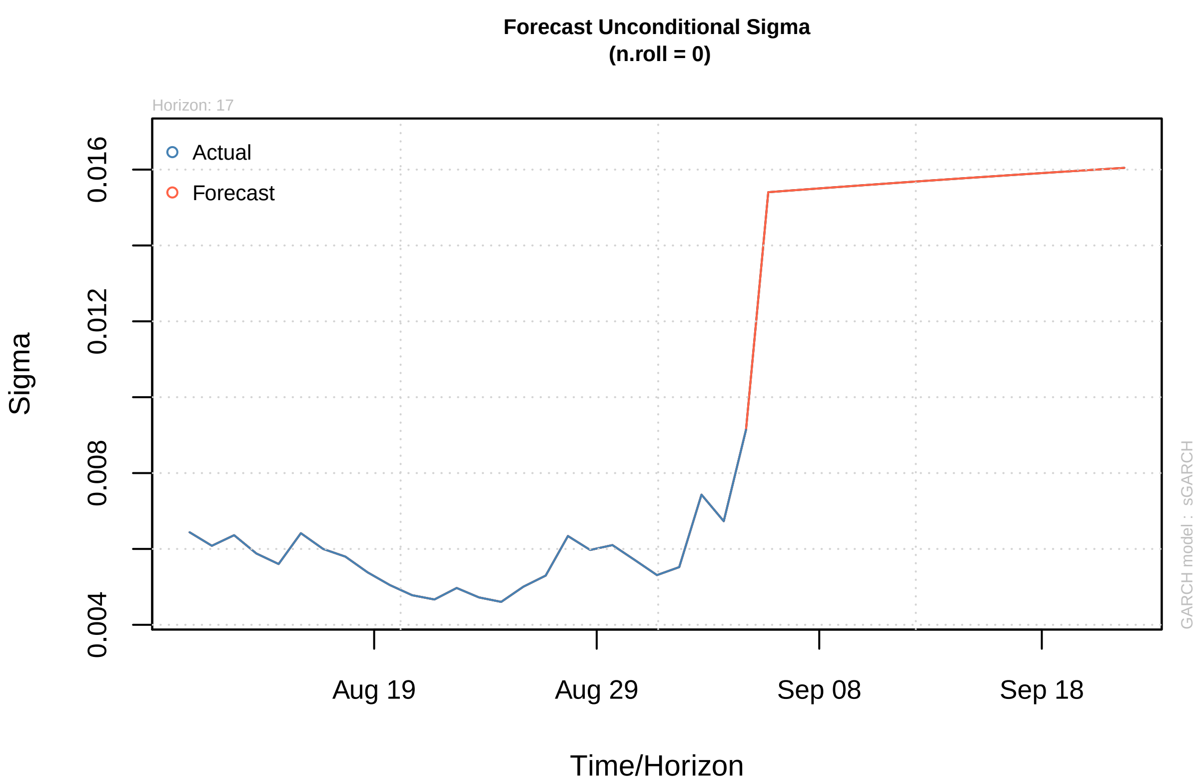



```
modelforGDP = ugarchforecast(modelfitGDP, data = gdp.growth$gdp, n.ahead = 17, n.roll = 0, out.sample =
```

```r
plot(modelforGDP,  which = 1 )
```




```r
plot(modelforGDP,  which = 3 )
```



With the forecast of GDP, I used a ARMA(3,0) and GARCH(1,1) order. The 95% bands are narrower and the volatility is smaller than the forecast in 14.5. For the sigma forecast, the forecast estimates it will be higher.