Homework 2

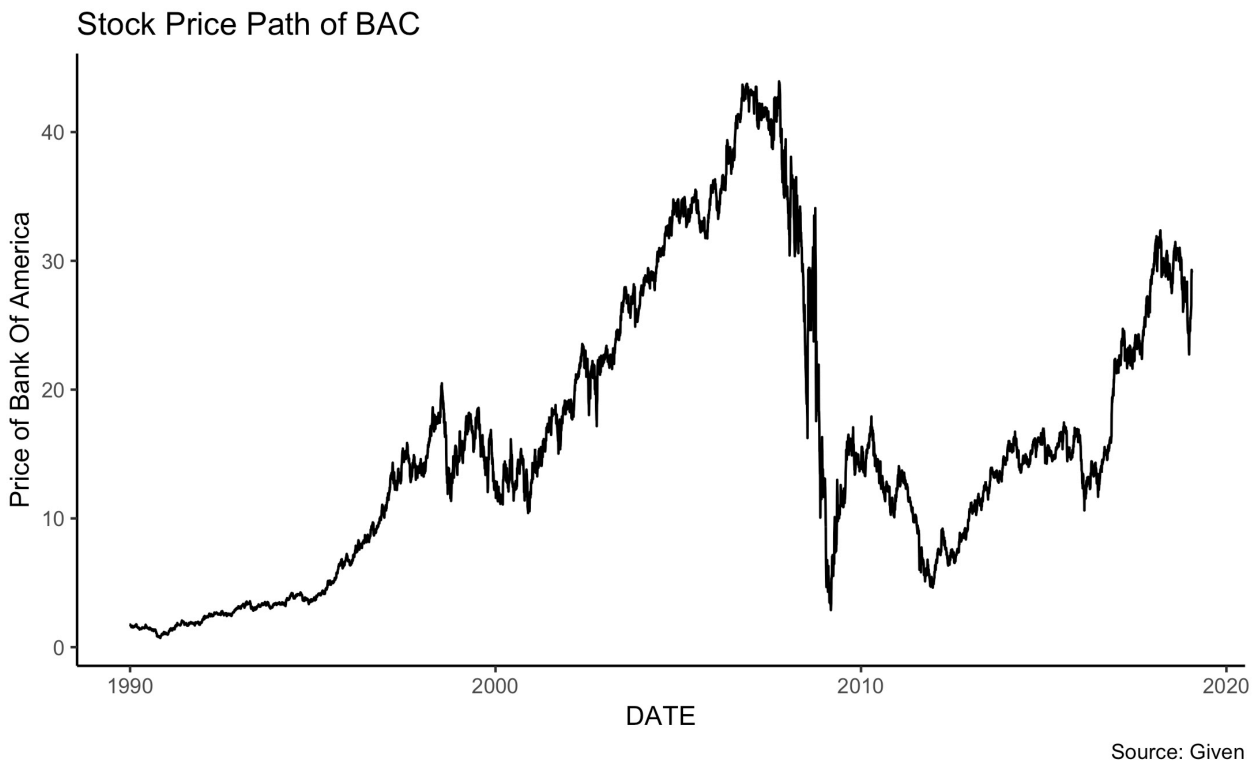
This R notebook is created by Kaiyi Li at March 1st, 2020 for the Homework 2 of volatility class.

Before we start everything, we read the data. (I convert the txt file to csv file for convienece.)

```r
bac_price <- read.csv('bac19.csv',head = T)
bac <- bac_price$bac
date <- bac_price$date
bac_price <- data.frame(
  DATE = as.Date.factor(date),
  price = bac
)
# head(bac_price)
```

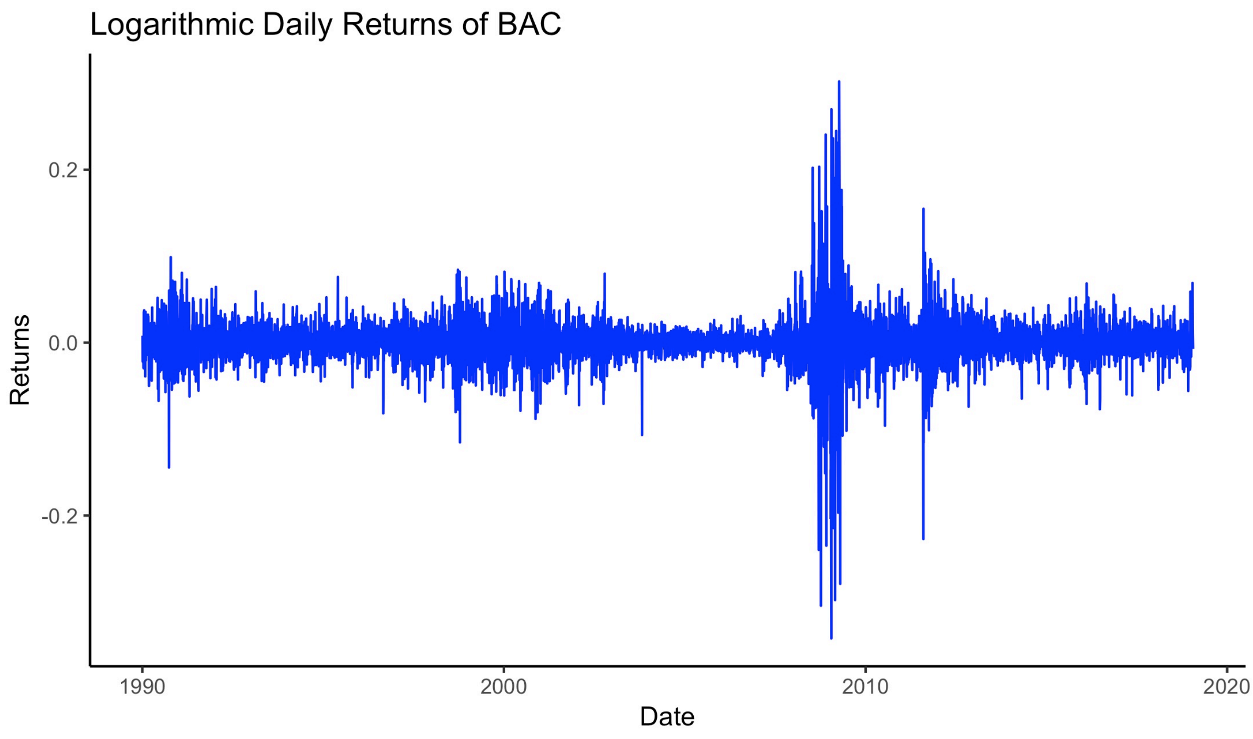
Plot the price path of BAC stock.

```r
library(ggplot2)
library(dplyr)
theme_set(theme_classic())
ggplot(bac_price, aes(x=DATE, y=price)) +
  geom_line() +
  labs(title="Stock Price Path of BAC",
       caption="Source: Given",
       y="Price of Bank Of America")
  # xlab("")
```



Plot the return path of bac.

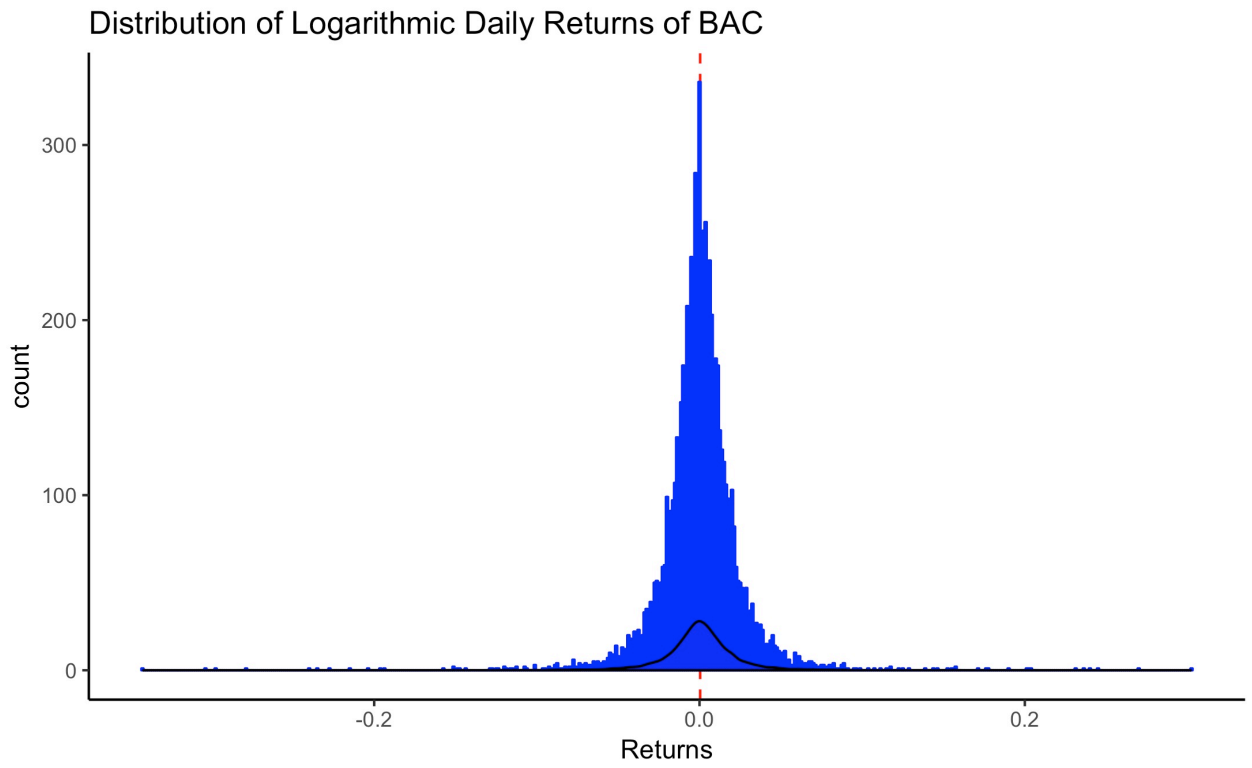
```
# Data frame of the returns
bac_daily_returns<-diff(log(bac))
bac_daily_returns <- data.frame(
  Date = as.Date.factor(date[-1]),
  Returns = bac_daily_returns
)
ggplot(bac_daily_returns, aes(x=Date)) +
  geom_line(aes(y=Returns),group = 1,colour = "blue") +
  labs(title="Logarithmic Daily Returns of BAC",
       caption="Source: Given ",
       y="Returns")
```



Source: Given

Plot the histogram of returns.

```
# ggplot(df, aes(x=weight)) + geom_histogram()
ggplot(bac_daily_returns,aes(x=Returns))+
  geom_vline(aes(xintercept=mean(Returns)),colour =
"red",linetype="dashed")+geom_histogram(binwidth = 0.00125,colour = "blue")+
  geom_density(alpha=0.05, fill="#FF6666") +
  labs(
  title = "Distribution of Logarithmic Daily Returns of BAC"
)
```



1 Use the attached data of daily equity prices of Bank of America (BAC) starting in 1990. In each case estimate the model with an intercept.

a)Estimate an ARCH(1) model and report the Schwarz information criterion

```
library(rugarch)
garchSpec <- ugarchspec(
  variance.model=list(model="sGARCH", garchOrder=c(1,0)),
  mean.model=list(armaOrder = c(0,0),include.mean=F))
fit = ugarchfit(spec = garchSpec, data = bac_daily_returns$Returns )
print(fit)
```

The SIC (BIC) for ARCH(1) is -4.8154

b) Estimate an ARCH(9) model and report the Schwarz information criterion

```
garchSpec <- ugarchspec(
  variance.model=list(model="sGARCH", garchOrder=c(9,0)),
  mean.model=list(armaOrder = c(0,0),include.mean=F))
fit = ugarchfit(spec = garchSpec, data = bac_daily_returns$Returns )
print(fit)
```

The SIC (BIC) for ARCH(9) model is -5.1150.

c) Estimate a GARCH(1,1) model and report the Schwarz information criterion

```
garchSpec <- ugarchspec(
  variance.model=list(model="sGARCH", garchOrder=c(1,1)),
  mean.model=list(armaOrder = c(0,0),include.mean=F))
fit = ugarchfit(spec = garchSpec, data = bac_daily_returns$Returns )
print(fit)
```

The SIC (BIC) for GARCH(1,1) model is -5.1535.

d) Estimate a GARCH(1,2) model and report the Schwarz information criterion

```
garchSpec <- ugarchspec(
  variance.model=list(model="sGARCH", garchOrder=c(2,1)),
  mean.model=list(armaOrder = c(0,0),include.mean=F))
fit = ugarchfit(spec = garchSpec, data = bac_daily_returns$Returns )
print(fit)
```

The SIC (BIC) for GARCH(1,2) model is -5.1523.

e) Pick another order GARCH model and report the Schwarz criterion

I picked GARCH(2,2) model.

```
garchSpec <- ugarchspec(
  variance.model=list(model="sGARCH", garchOrder=c(2,2)),
  mean.model=list(armaOrder = c(0,0),include.mean=F))
fit = ugarchfit(spec = garchSpec, data = bac_daily_returns$Returns )
print(fit)
```

The SIC (BIC) for GARCH(2,2) model is -5.1536.

f) Introduce another lagged or deterministic variable to see if it is significant.

```
garchSpec <- ugarchspec(
  variance.model=list(model="sGARCH", garchOrder=c(1,1)),
  mean.model=list(armaOrder = c(1,1),include.mean=F))
fit = ugarchfit(spec = garchSpec, data = bac_daily_returns$Returns )
print(fit)
```

Instead of not using any model, I used ARMA(1,1) as a mean model. It turns out that both the ar(1) term and ma(1) term are not significant.

Which is preferred? Do all of these models satisfy the basic criteria for a good model?

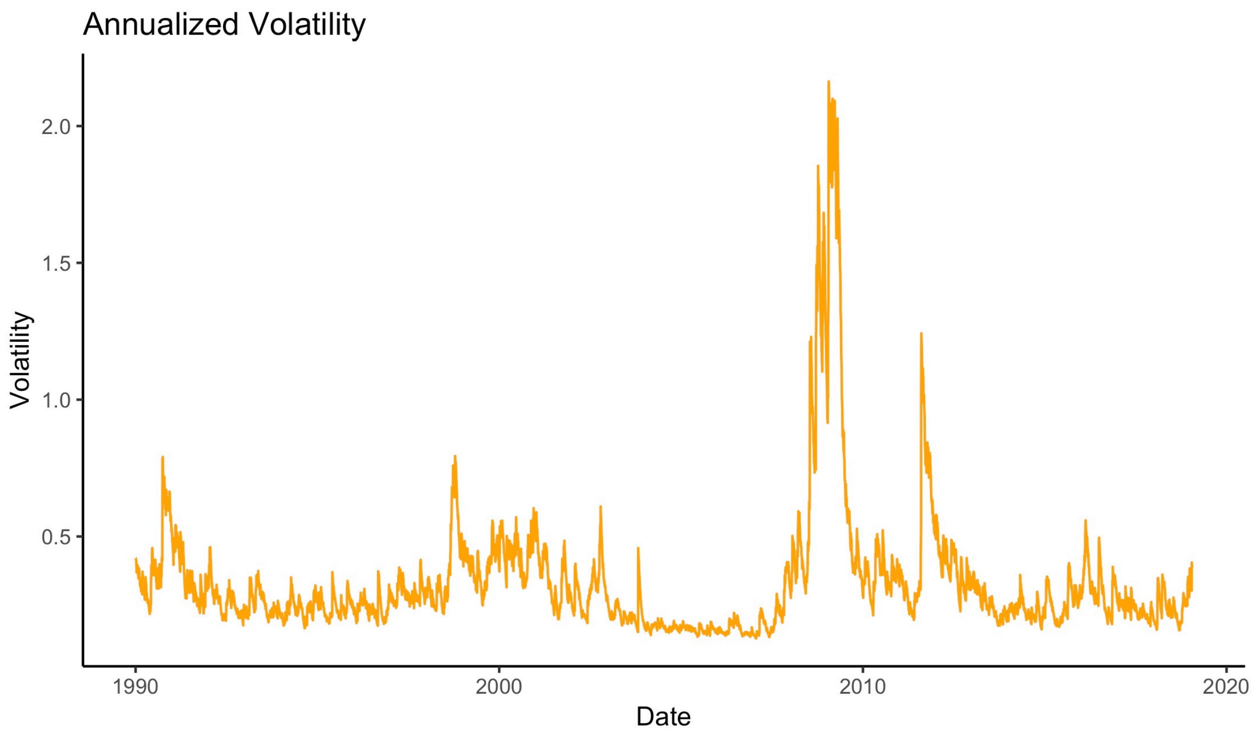
GARCH(1,1) is preferred given it has the most significant joint effect. Only GARCH(1,1) and GARCH(2,1) satisfy basic criteria as good models for their significance in the joint effect.

2. For the GARCH(1,2) model(one GARCH and two ARCH terms), calculate the time series of annualized volatilities. What was the maximum conditional volatility and when did this occur?

First we fit the GARCH(1,2) model.

```
#Fit the GARCH(1,2) model
garchSpec <- ugarchspec(
  variance.model=list(model="sGARCH", garchOrder=c(2,1)),
  mean.model=list(armaOrder = c(0,0),include.mean=F))
fit = ugarchfit(spec = garchSpec, data = bac_daily_returns$Returns )
```

Calculate and plot the annualized conditional volatility series



Source: Given

```
annualized_volatility <- fit@fit$sigma*sqrt(252)
bac_annualized_volatility <- data.frame(
  Date = as.Date.factor(date[-1]),
  Volatility = annualized_volatility
)
ggplot(bac_annualized_volatility, aes(x=Date)) +
  geom_line(aes(y=Volatility),group = 1,colour = "orange") +
  labs(title="Annualized Volatility",
       caption="Source: Given ",
       y="Volatility")
```

Find the date of the maximum conditional volatility.

```
bac_annualized_volatility[which.max(bac_annualized_volatility$Volatility),]
```

The maximum conditional volatility is 2.163118 (annualized), it occured on 2009-01-23.

3.Test the autocorrelation of the standardized residuals and the squared standardized residuals with 10 lags. Does this model pass both tests? Explain.

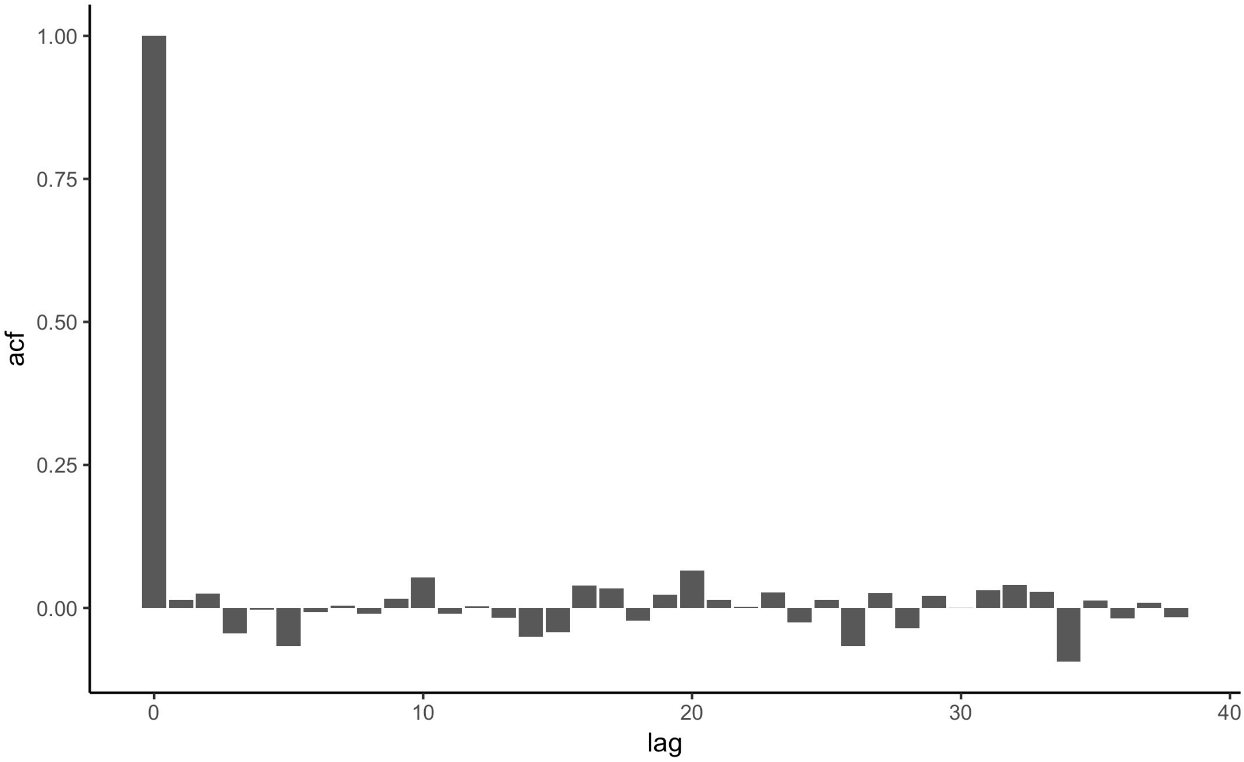
Calculate the standarized_residuals and its square.

```
standarized_residuals <- (fit@fit$residuals-
mean(fit@fit$residuals))/sqrt(var(fit@fit$residuals))
squared_standarized_residuals <- standarized_residuals**2
```

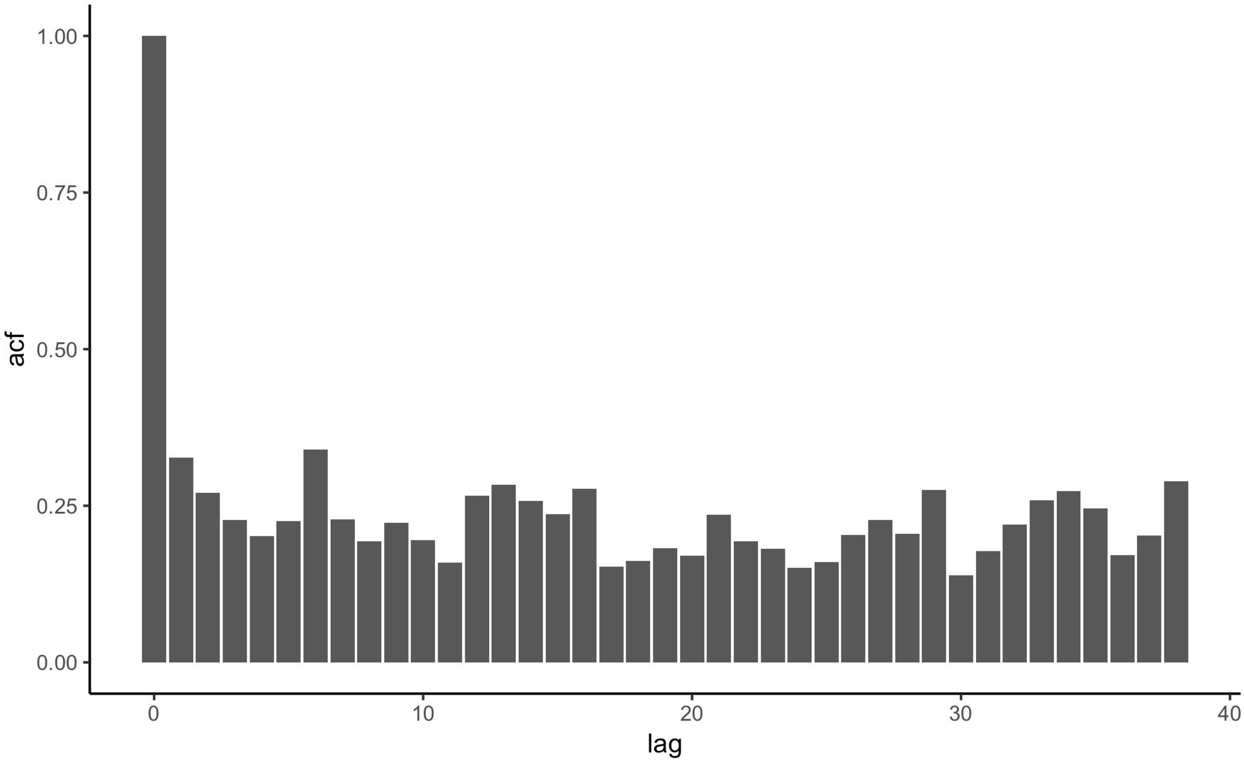
Plot their ACFs.

```
sracf <- acf(standarized_residuals,plot = F)
sracf <- with(sracf, data.frame(lag, acf))
ssracf <-acf(squared_standarized_residuals,plot = F)
ssracf <- with(ssracf, data.frame(lag, acf))
ggplot(data=sracf, mapping=aes(x=lag, y=acf))+
       geom_bar(stat = "identity", position = "identity")
ggplot(data=ssracf, mapping=aes(x=lag, y=acf))+
       geom_bar(stat = "identity", position = "identity")
```

The ACF plot of standarized residual:



The ACF plot of squared standarized Residuals



Test for Autocorrelation H0: The residuals are not autocorrelated. Ha: The residuals are autocorrelated. Significance: 0.05.

```
qchisq(0.95,30)
```

```
library(stats)
library(lawstat)
BT = Box.test(standarized_residuals, lag=10, type = "Ljung-Box", fitdf=0)
BT
```

Given that 76.866>50.89, we reject the null hypothesis at 95% confidence interval. Therefore, **The standarized residuals are autocorrelated.**

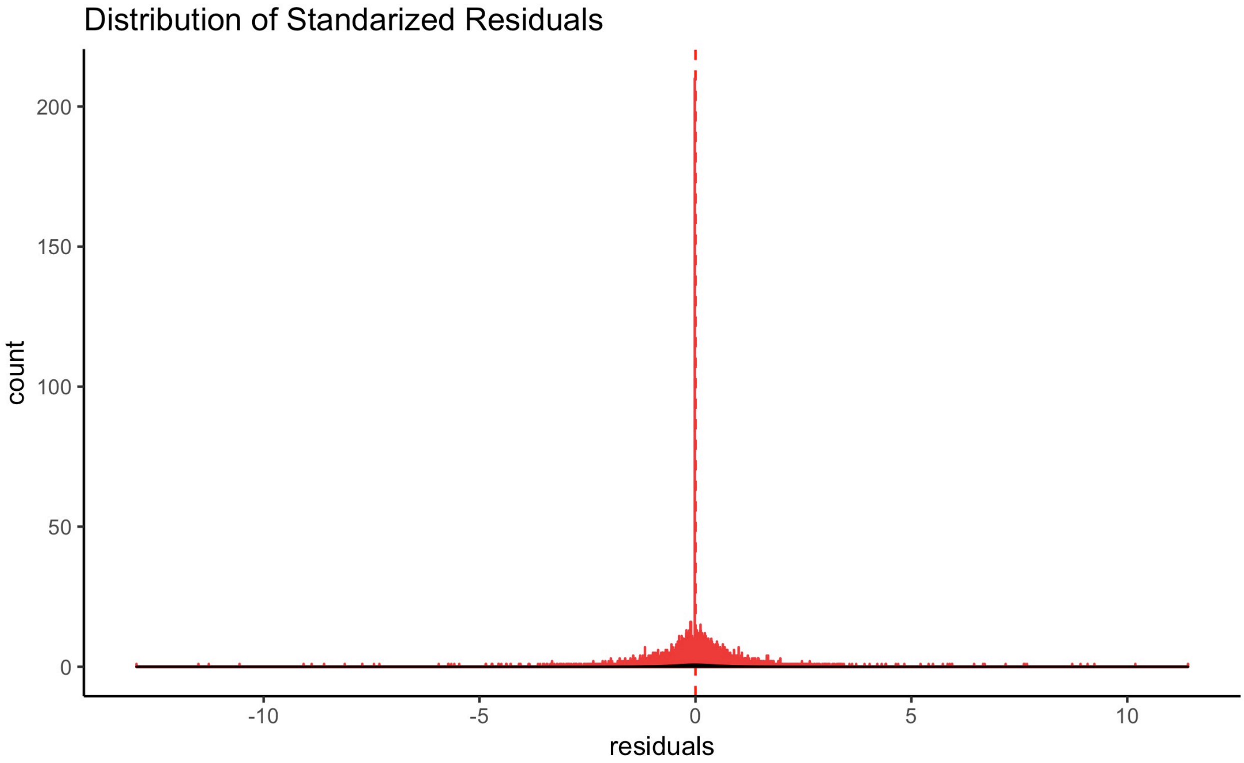
```
BT = Box.test(squared_standarized_residuals, lag=10, type = "Ljung-Box", fitdf=0)
BT
```

Given that 4511.1>50.89, we reject the null hypothesis at 95% confidence interval. Therefore, **The squared standarized residuals are autocorrelated.**

4. Report the skewness and kurtosis of the standardized residuals. Compare these with the BAC returns.

Plot the distribution of standarized residuals.

```
Residuals<-c()
Residuals <-data.frame(
  residuals = standarized_residuals,
  Date = date[-1]
)
ggplot(Residuals,aes(x=residuals))+
  geom_vline(aes(xintercept=mean(residuals)),colour =
"red",linetype="dashed")+geom_histogram(binwidth = 0.00125,colour = "brown2")+
  geom_density(alpha=0.5, fill="#FF6666") +
  labs(
  title = "Distribution of Standarized Residuals"
)
```



Calculate the kurtosis and skewness for standarized residual.

```
library(moments)
kurtosis(standarized_residuals)
skewness(standarized_residuals)
```

The kurtosis of standarized residuals is 29.97432, the skewness of standarized residuals is -0.3316823.

Calculate the kurtosis and skewness for daily returns of BAC.

```
kurtosis(bac_daily_returns$Returns)
skewness(bac_daily_returns$Returns)
```

The kurtosis of standarized residuals is 29.97432, the skewness of standarized residuals is -0.3316823.

The kurtosis and skewness of standarized residuals and daily returns of BAC are excatly the same!

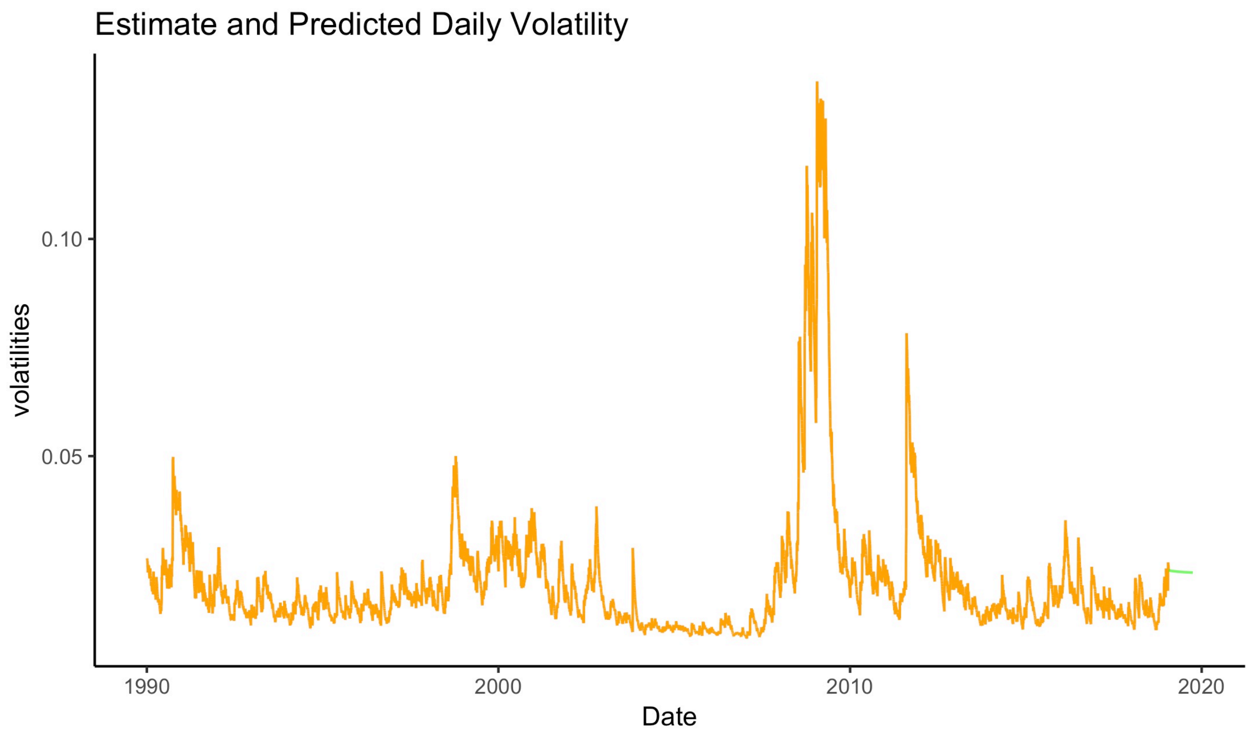
5. Forecast the next year of daily volatility for BAC and plot the result.

First, forecast daily volatility of next 252 days.

```r
spec <- ugarchspec(
  variance.model=list(model="sGARCH", garchOrder=c(2,1)),
  mean.model=list(armaOrder = c(0,0),include.mean=F))
fit = ugarchfit(data = bac_daily_returns$Returns, spec = spec)
forc = ugarchforecast(fit, n.ahead=252)
```

Second, plot the forecasted daily volatility.

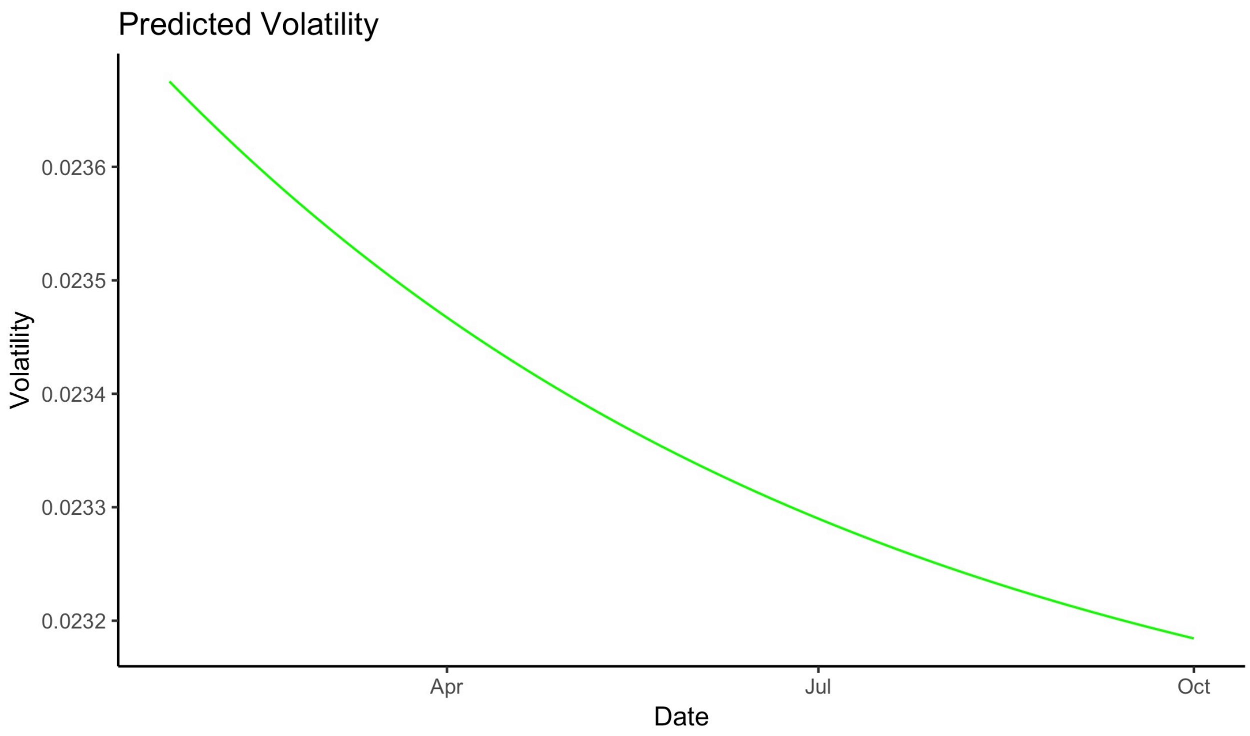
```r
x <- c(unclass(sigma(forc))[,"1990-01-16 08:00:00"][1:252])
x <- c(fit@fit$sigma,x)
horizon <- seq.Date(as.Date(date[length(date)])+1, length.out = 252, by = "day")
horizon <- c(as.Date(date)[-1],horizon)
combined_daily_data <- data.frame(
  Date = horizon,
  volatilities = x,
  col = c(rep("orange", 7320), rep("green", 252))
)
ggplot(combined_daily_data, aes(x=Date, y=volatilities)) +
  geom_line(aes(colour=col, group=1)) +
  scale_colour_identity()+
  labs(
    title = "Estimate and Predicted Daily Volatility",
    caption = "Yellow: Estimates, Green: Predicted"
  )
```



Yellow: Estimates, Green: Predicted

Or just see the predicted plot

```
predicted_sets <- data.frame(
  Date = seq.Date(as.Date(date[length(date)])+1, length.out = 252, by = "day"),
  volatilities = c(unclass(sigma(forc))[,"1990-01-16 08:00:00"][1:252])
)
ggplot(predicted_sets, aes(x=Date)) +
  geom_line(aes(y=volatilities),group = 1,colour = "green") +
  labs(title="Predicted Volatility",
       caption="Source: Predicted ",
       y="Volatility")
```



Source: Predicted

6. Reestimate the GARCH(1,2) with student-t distribution. Now what is the Schwarz criterion? Does it find this estimate preferable?

```
spec <- ugarchspec(
  variance.model=list(model="sGARCH", garchOrder=c(2,1)),
  mean.model=list(armaOrder = c(0,0),include.mean=F),distribution.model = "std")
fit = ugarchfit(data = bac_daily_returns$Returns, spec = spec)
print(fit)
```

The SIC (BIC) for GARCH(2,1) model is -5.2216, which is smaller than -5.1523(BIC for GARCH(2,1) with normal distribution). Therefore, GARCH(2,1) with student t distribution is preferrable.

7. Describe the volatility pattern of a GARCH(1,1):

Because we can rewrite the GARCH(1,1) in this fashion:

$$\epsilon_t^2 = \omega + (\alpha + \beta)\epsilon_{t-1}^2 + v_t - \beta v_{t-1}$$
$$v_t - \beta v_{t-1} = \epsilon_t^2 - (\alpha + \beta)\epsilon_{t-1}^2 - \omega$$

a. When the sum of alpha plus beta is small?

When the sum of alpha plus beta is small, (smaller than 1), it is still stationary. The effect of volatility shock will not be persistent for long.(Black line in the attached simulation plot)

b. When the sum of alpha plus beta is bigger than one?

When the sum of alpha plus beta is bigger than 1, it is not stationary. The effect of volatility shock will be persistent. Because we are unable to simulate this situation, we simulate a close one (alpha = 0.5,beta = 0.49, green line in the simulation plot)

c. When alpha is small and beta is big with a sum slightly less than one?

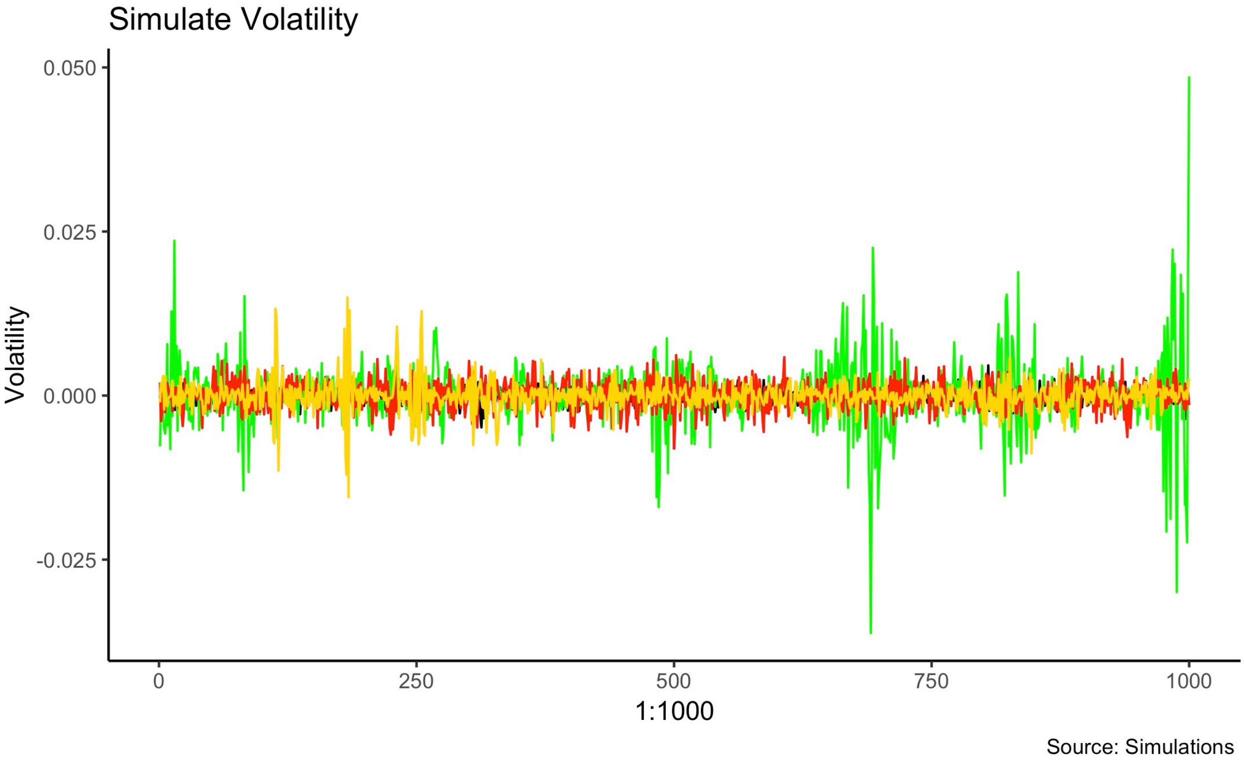
In the short run, the effect of volatility shock decays really quick, but it lasts for a long time.

d. When alpha is big and beta is small with a sum slightly less than one?

In the short run, the effect of volatility shock persist, but it only last for a very short period.

To further illustrate my points, here are some simulations.

```r
library(fGarch)
spec1 = garchSpec(model = list(alpha=0.1,beta=0.1))
sim1<- garchSim(spec1, n = 1000)
spec2 = garchSpec(model = list(alpha=0.5,beta=0.49))
sim2<- garchSim(spec2, n = 1000)
spec3 = garchSpec(model = list(alpha=0.1,beta=0.7))
sim3<- garchSim(spec3, n = 1000)
spec4 = garchSpec(model = list(alpha=0.7,beta=0.1))
sim4<- garchSim(spec4, n = 1000)
# spec4 = garchSim(model = list(alpha = 0.8,beta = 0.1))
# sim4 <- garchSim(spec4,n=1000)
sims <- cbind(sim1$garch,sim2$garch,sim3$garch,sim4$garch)
sims <- as.data.frame(sims)
ggplot(as.data.frame(sims),aes(x=1:1000)) +
  geom_line(aes(y=sims$V1),group = 1,colour = "black") +
  geom_line(aes(y=sims$V2),group = 1,colour = "green")+
  geom_line(aes(y=sims$V3),group = 1,colour = "red") +
  geom_line(aes(y=sims$V4),group = 1,colour = "gold") +
  labs(title="Simulate Volatility",
       caption="Source: Simulations ",
       y="Volatility")
```


Simulate Volatility
0.050
0.025
0.000
-0.025
Volatility
0
250
500
750
1000
1:1000
Source: Simulations