Exploratory Data Analysis

Loading libraries

The below code chunk loads the libraries we will be using in our analysis:

Reading and cleaning data

First, we input our stock data.

Our stock data consists of the following indices between 2000 and 2021:

- S&P500
- NASDAQ
- NYSE100

Important: Before running the code below, make sure your Knit directory is 'Document Directory'. This can be done by clicking the drop-down menu next to Knit, going to Knit directory and clicking on Document Directory.

```
setwd("..")
sp<-read.csv("Data/sp500.csv")
ny<-read.csv("Data/nyse.csv")
nas<-read.csv("Data/nasdaq.csv")
```

Now we will change the 'caldt' column to the Date format in order to plot the time series for each index:

```
sp$caldt<-as.Date(sp$caldt, format="%d/%m/%Y")
ny$caldt<-as.Date(ny$caldt, format="%d/%m/%Y")
nas$caldt<-as.Date(nas$caldt, format="%d/%m/%Y")

str(sp)
```

```
## 'data.frame':    5032 obs. of  2 variables:
##  $ caldt : Date, format: "2001-01-02" "2001-01-03" ...
##  $ spindx: num  1283 1348 1333 1298 1296 ...
```

```
str(ny)
```

```
## 'data.frame':    5284 obs. of  2 variables:
##  $ caldt : Date, format: "2000-01-03" "2000-01-04" ...
##  $ spindx: num  1455 1399 1402 1403 1441 ...
```

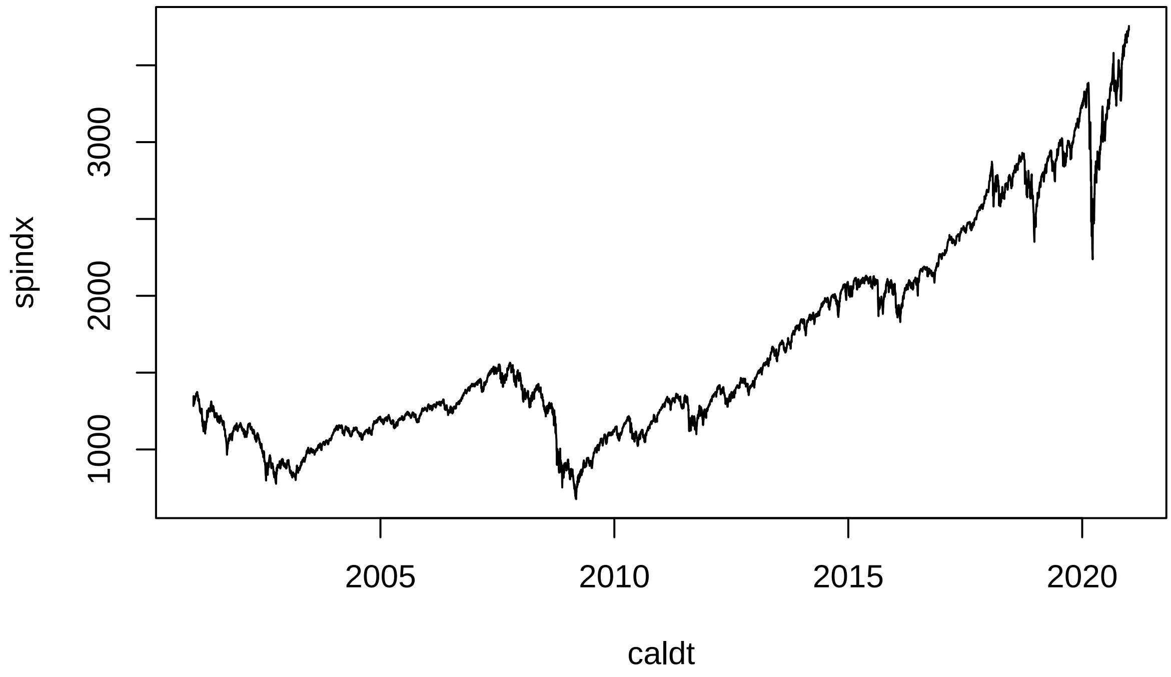
```
str(nas)
```

```
## 'data.frame':    5284 obs. of  2 variables:
##  $ caldt : Date, format: "2000-01-03" "2000-01-04" ...
##  $ ncindx: num  4131 3902 3878 3727 3883 ...
```

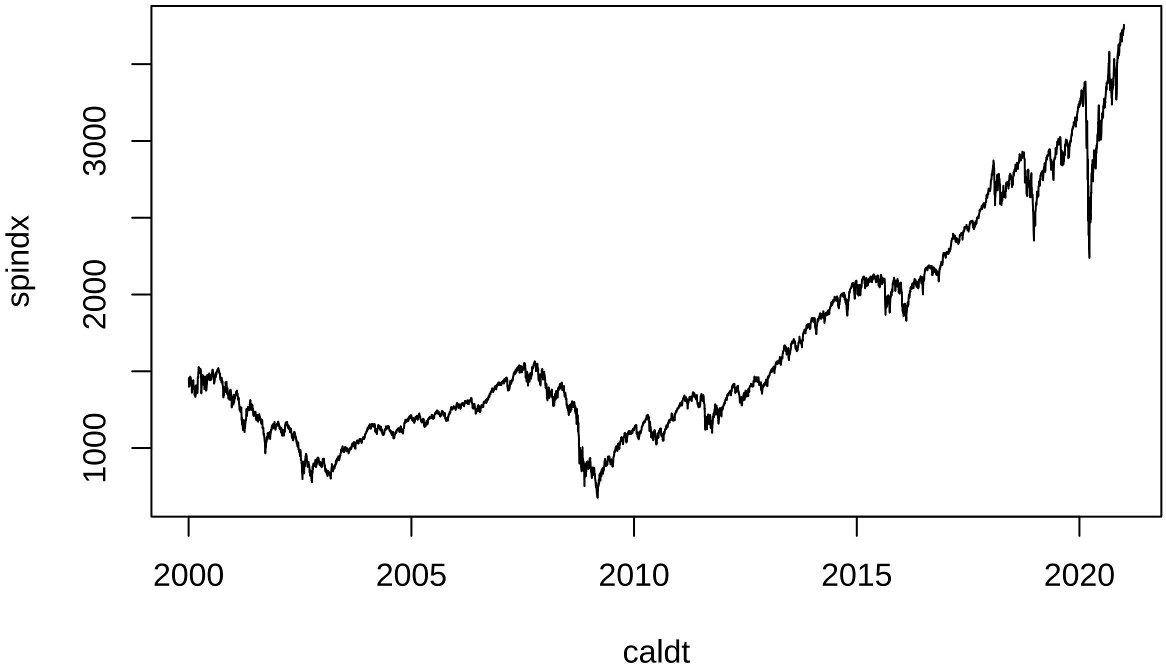
Initial Plots

We will start off by making a basic of stock price against time for each index, to get an idea of what our data looks like:

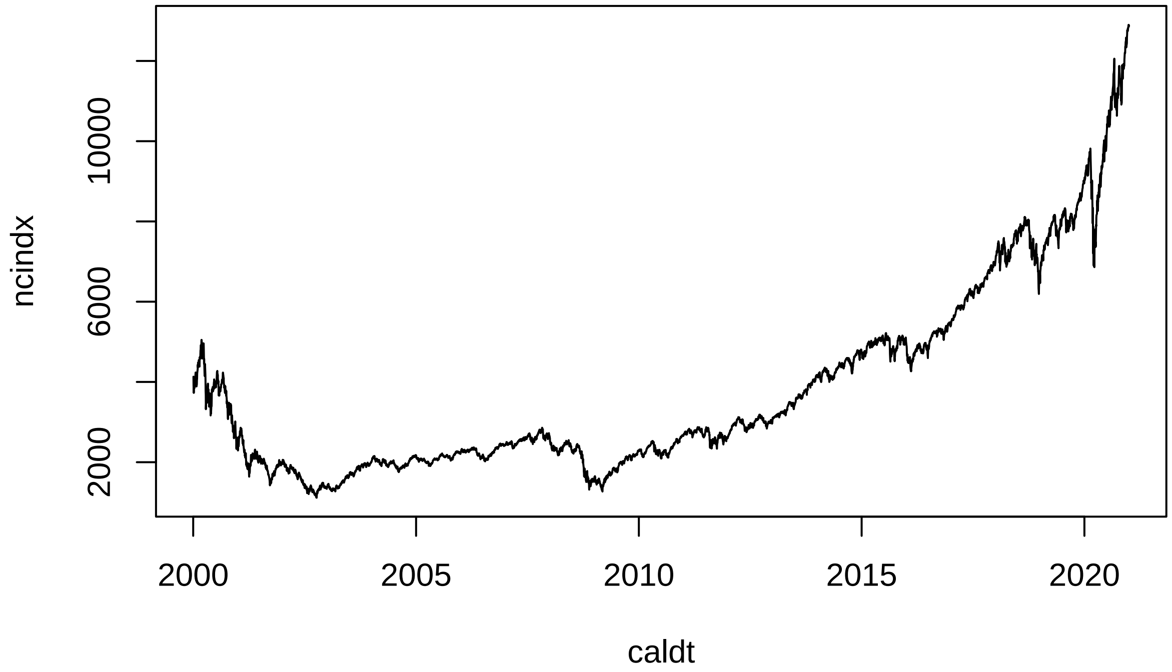
```
plot(sp, type='l')
```



```
plot(ny, type='l')
```



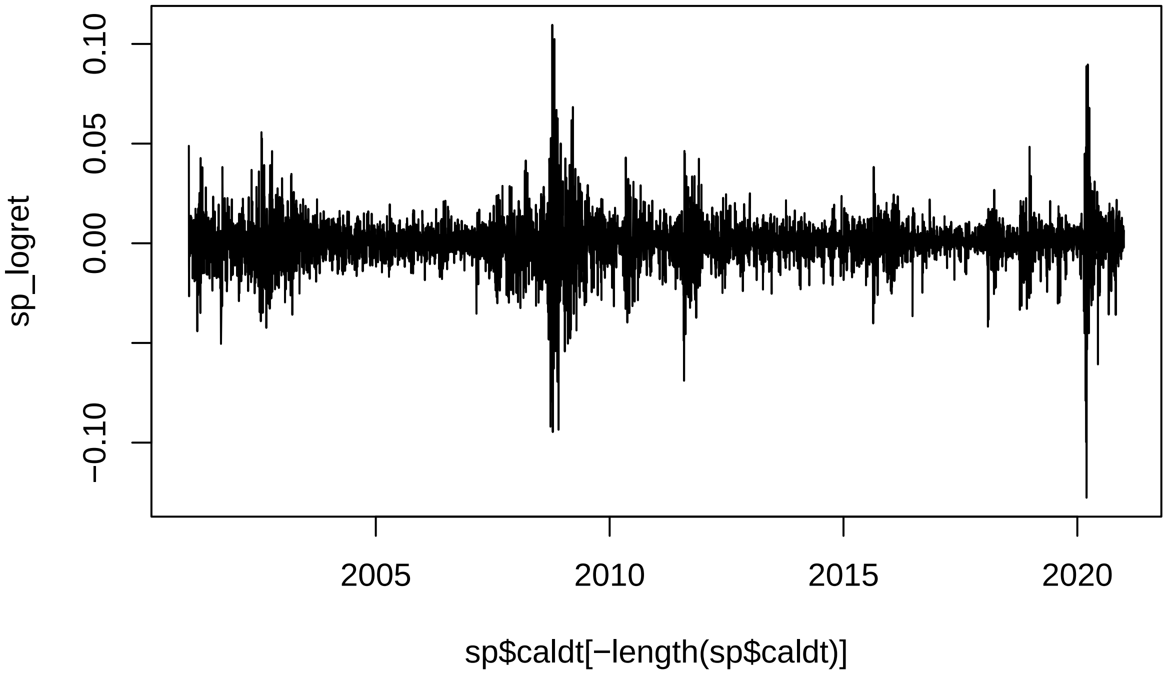
```
plot(nas, type='l')
```



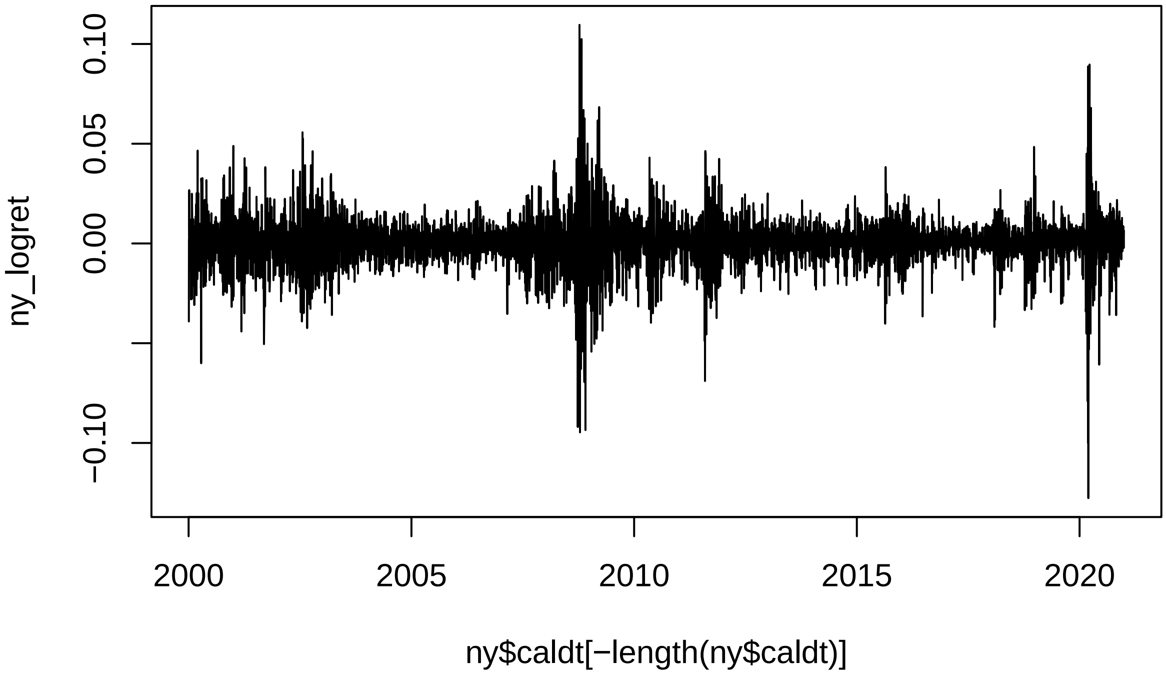
They all follow the same basic pattern, which is what we would expect, with the iconic fall in stock-price during the 2008-2009 period of the 'Great Recession'.

However, the stock price directly does not give us much information. Instead, we will take at the **daily log stock returns**.

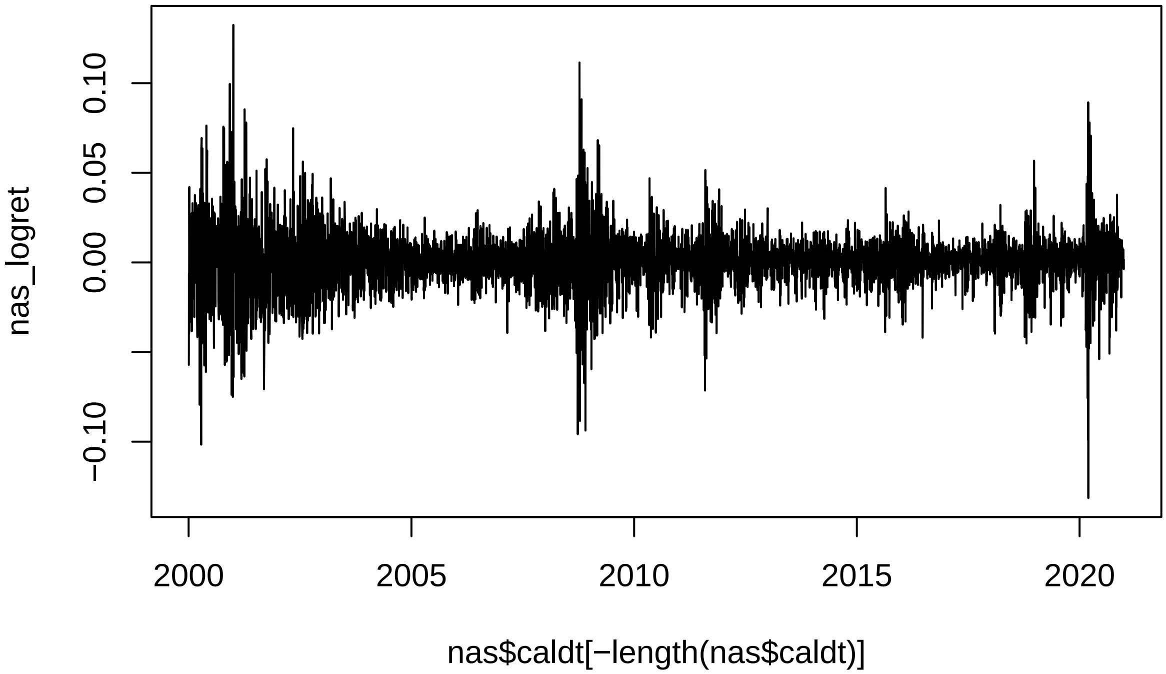
```
sp_logret <- diff(log(sp$spindx))
ny_logret <- diff(log(ny$spindx))
nas_logret <- diff(log(nas$ncindx))

plot(sp$caldt[-length(sp$caldt)],sp_logret,type='l')
```



```
plot(ny$caldt[-length(ny$caldt)],ny_logret, type='l')
```



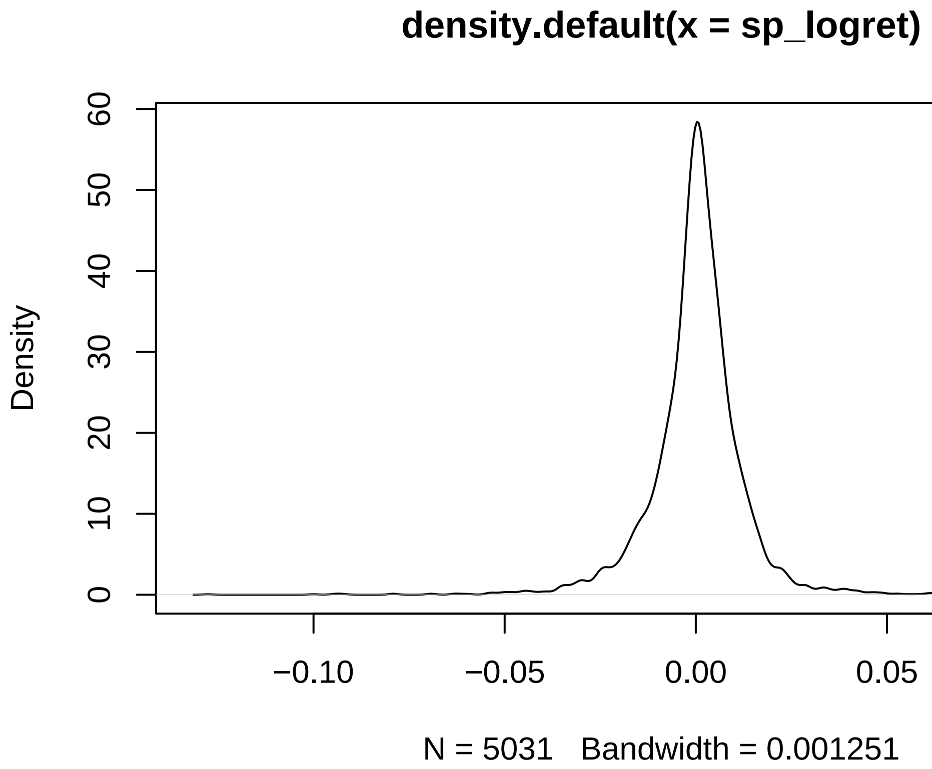
```r
plot(nas$caldt[-length(nas$caldt)],nas_logret, type='l')
```



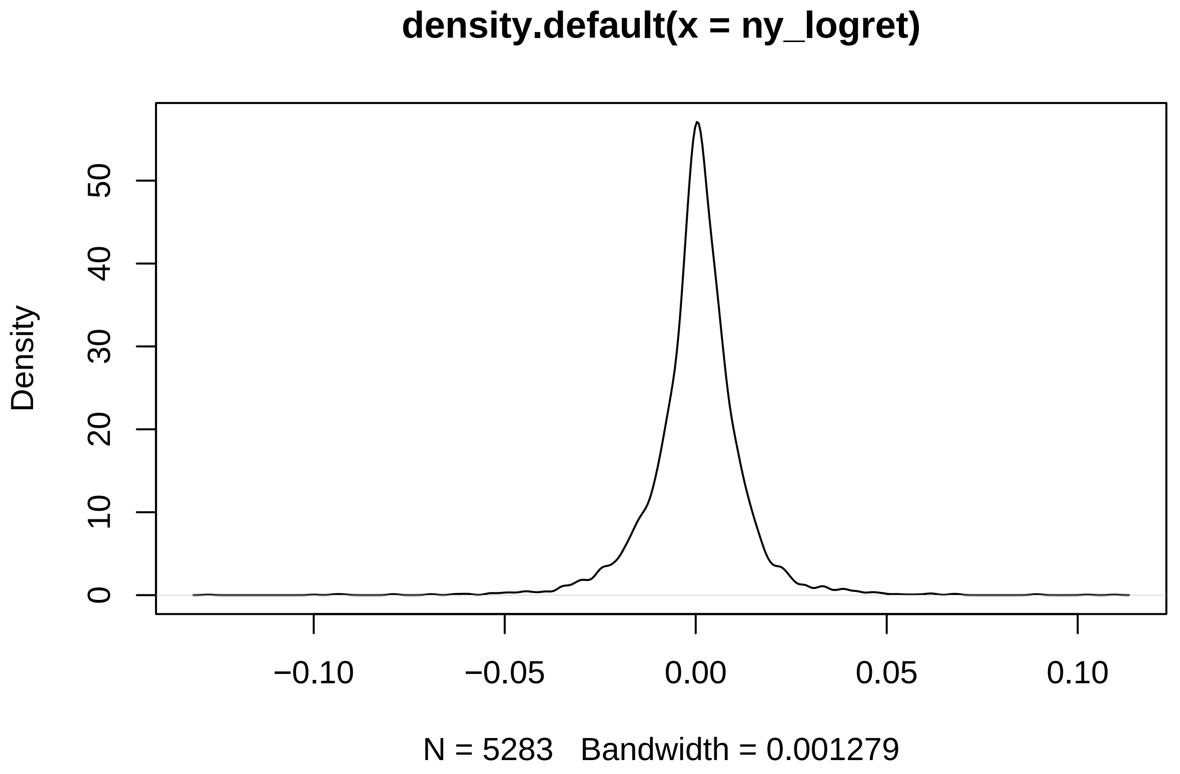
We can see that the returns average around 0% with very high variability during 2008-2009 (caused by the Great Recession) and during 2020 (caused by COVID-19).

Let us now plot the density of the returns to try to understand the distribution which will be helpful when we try to model the returns later one:

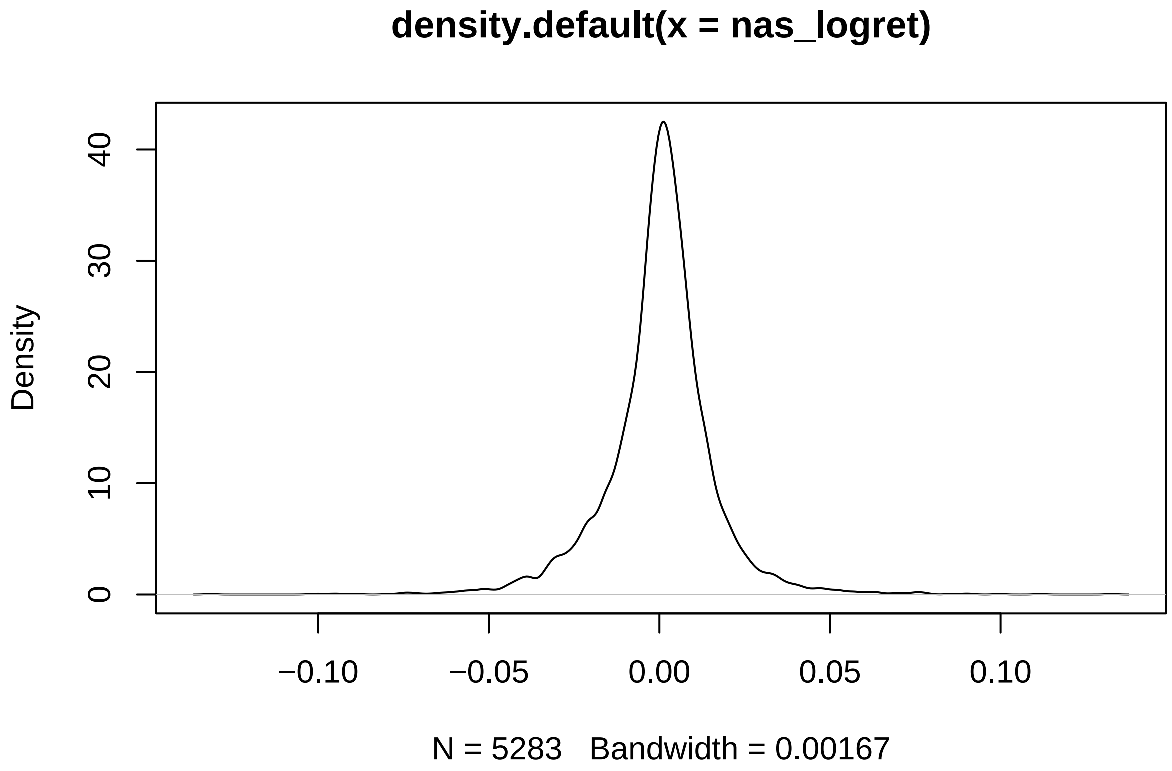
```
plot(density(sp_logret))
```



```r
plot(density(ny_logret))
```

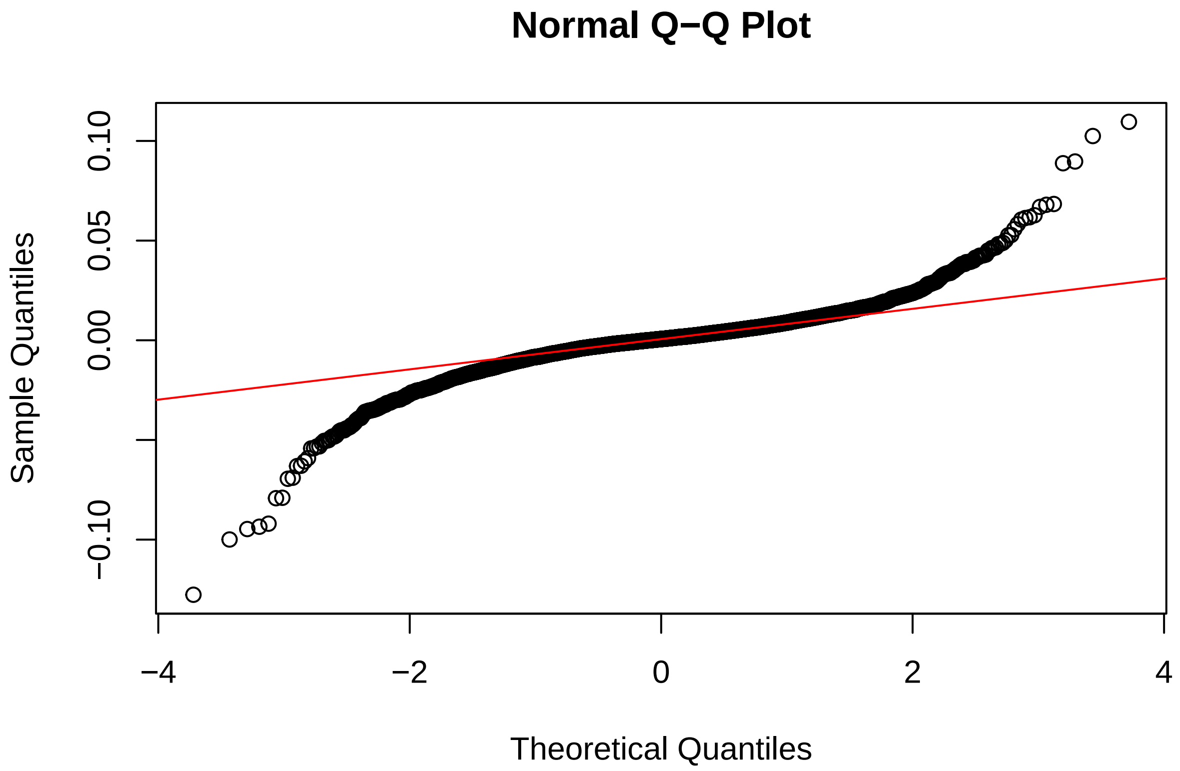


```
plot(density(nas_logret))
```

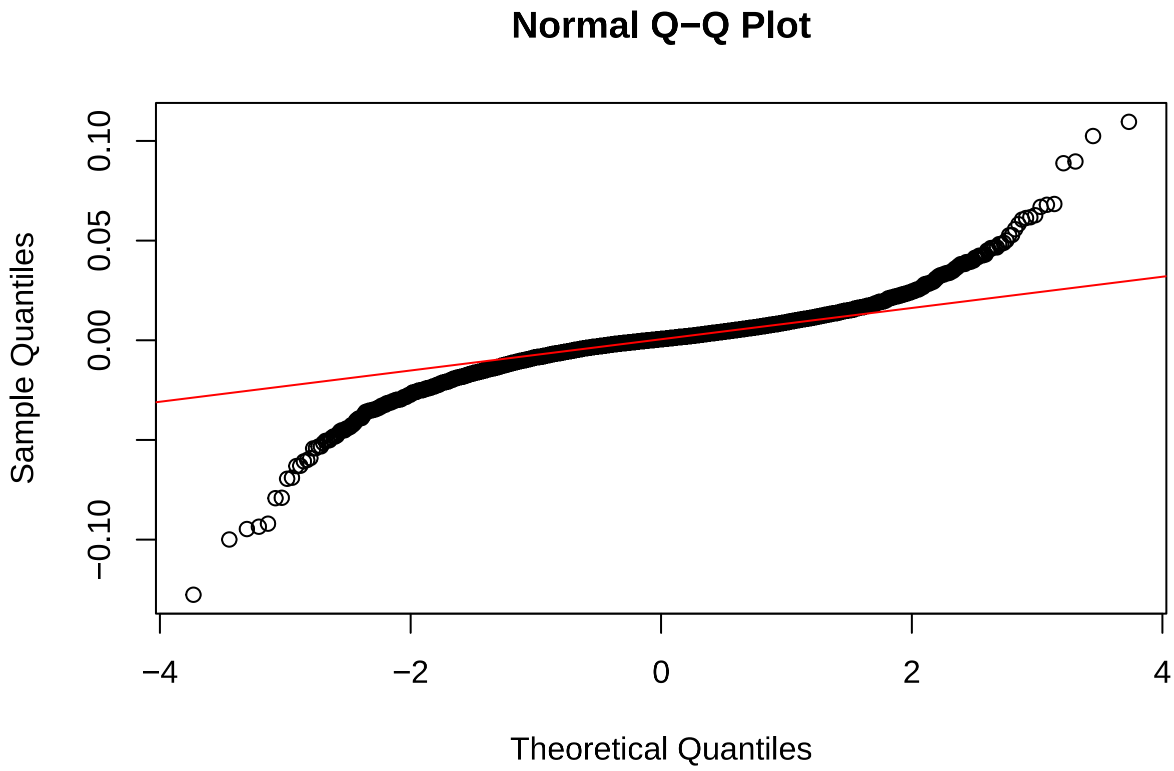


The returns look like they follow a normal distribution. So, we will make qq-plots to further confirm this:

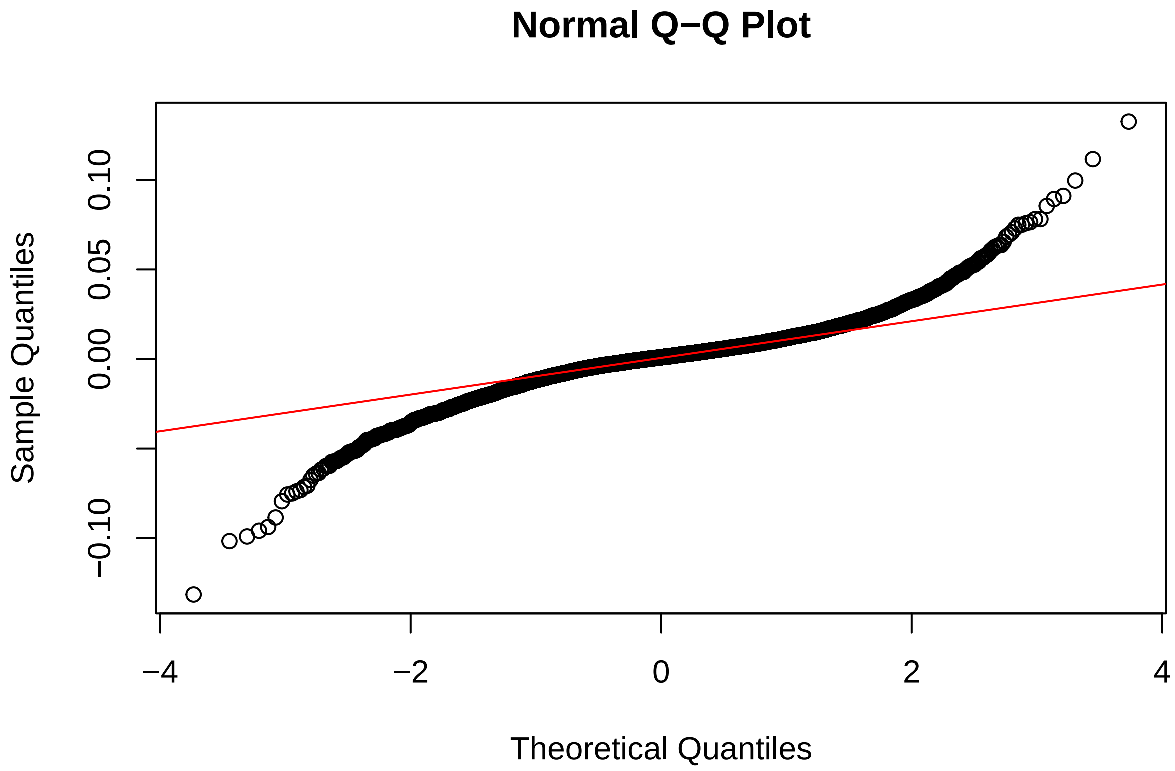
```
qqnorm(sp_logret)
qqline(sp_logret,col='red')
```



```
qqnorm(ny_logret)
qqline(ny_logret,col='red')
```



```
qqnorm(nas_logret)
qqline(nas_logret,col='red')
```



The log-returns have much heavier tails than the normal distribution, which suggests that it might follow a Student's t-distribution.

Calculating summary statistics

Let us now obtain some sample statistics of our data. We will first use summary():

```
summary(sp_logret)
```

```
##        Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.1276521 -0.0045195  0.0006476  0.0002135  0.0057220  0.1095720
```

```
summary(ny_logret)
```

```
##        Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.1276521 -0.0047659  0.0005935  0.0001795  0.0058089  0.1095720
```

```
summary(nas_logret)
```

```
##        Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.1314915 -0.0062893  0.0009564  0.0002154  0.0075183  0.1325465
```

Now we will calculate the skewness of our data:

```
skewness(sp_logret)
```

```
## [1] -0.4178326
```

```
skewness(ny_logret)
```

```
## [1] -0.393156
```

```
skewness(nas_logret)
```

```
## [1] -0.1333754
```

The skewness of our indexes are not equal to 0 which indicates that our log-returns might not be normally distributed. Let's also look at the tails of the distribution by calculating the sample kurtosis:

```
kurtosis(sp_logret)
```

```
## [1] 14.67896
```

```
kurtosis(ny_logret)
```

```
## [1] 13.94
```

```
kurtosis(nas_logret)
```

```
## [1] 9.621652
```

The sample kurtosis is much higher than 3 meaning our log-returns have much fatter tails than the normal distribution!

Doing basic time-series tests

We will carrying out tests to check if our series is stationary and auto-correlated.

We first test if our series is stationary:

```
lag.length = 25

Box.test(sp_logret, lag=lag.length, type="Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  sp_logret
## X-squared = 182.6, df = 25, p-value < 2.2e-16
```

```
Box.test(ny_logret, lag=lag.length, type="Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  ny_logret
## X-squared = 179.72, df = 25, p-value < 2.2e-16
```

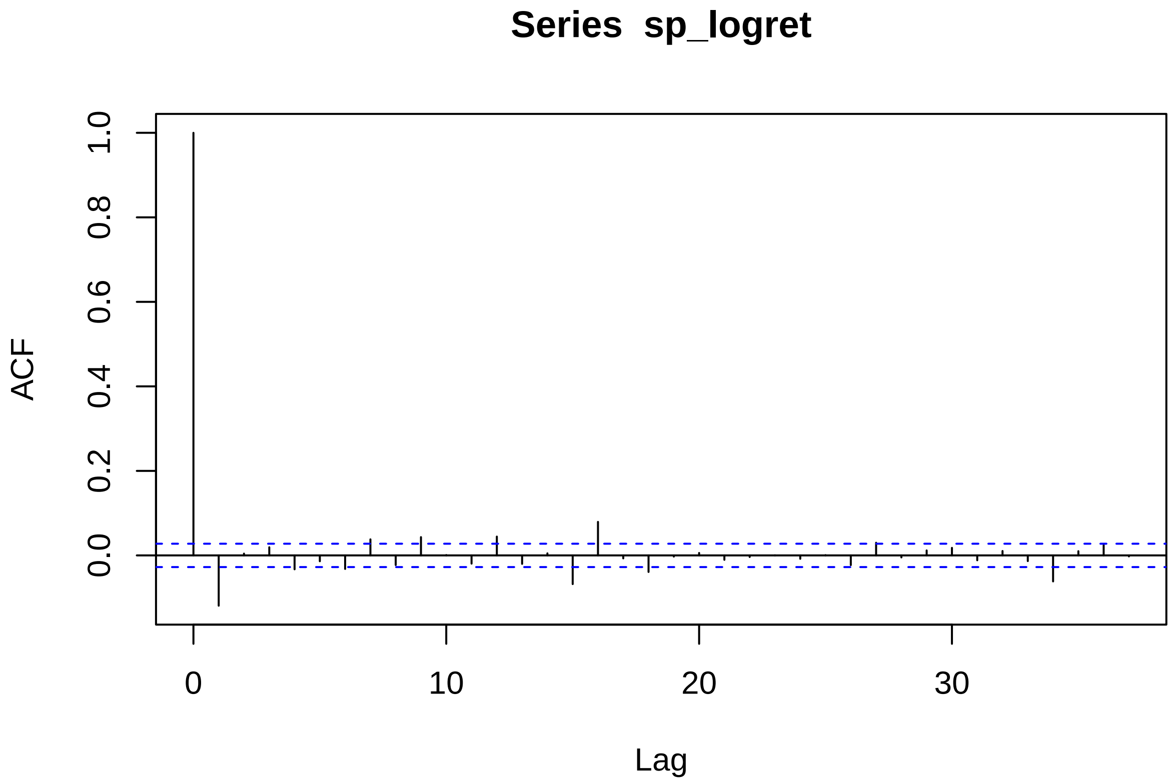
```
Box.test(nas_logret, lag=lag.length, type="Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  nas_logret
## X-squared = 133.38, df = 25, p-value < 2.2e-16
```

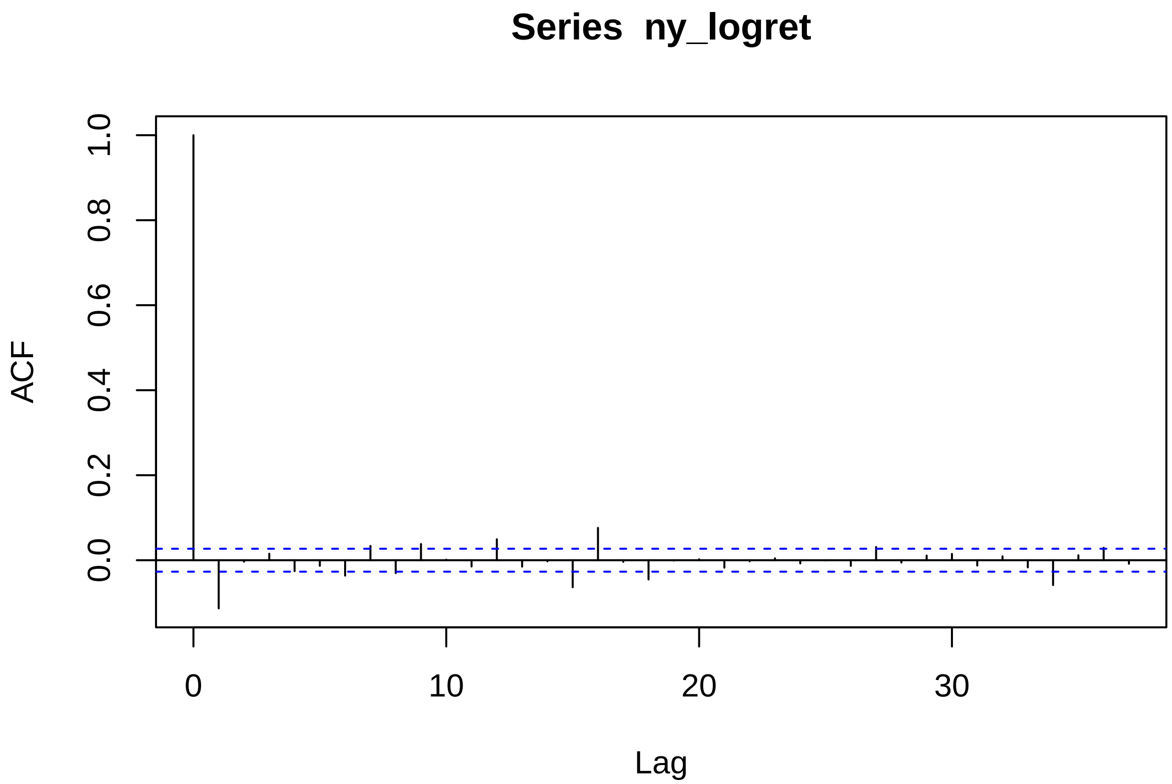
The p-value is very small which means we reject the null hypothesis that our correlations are 0. This means our data is not stationary and we might not use a GARCH model on log-returns directly.

We also plot the ACF of our indexes to see how our data is correlated:

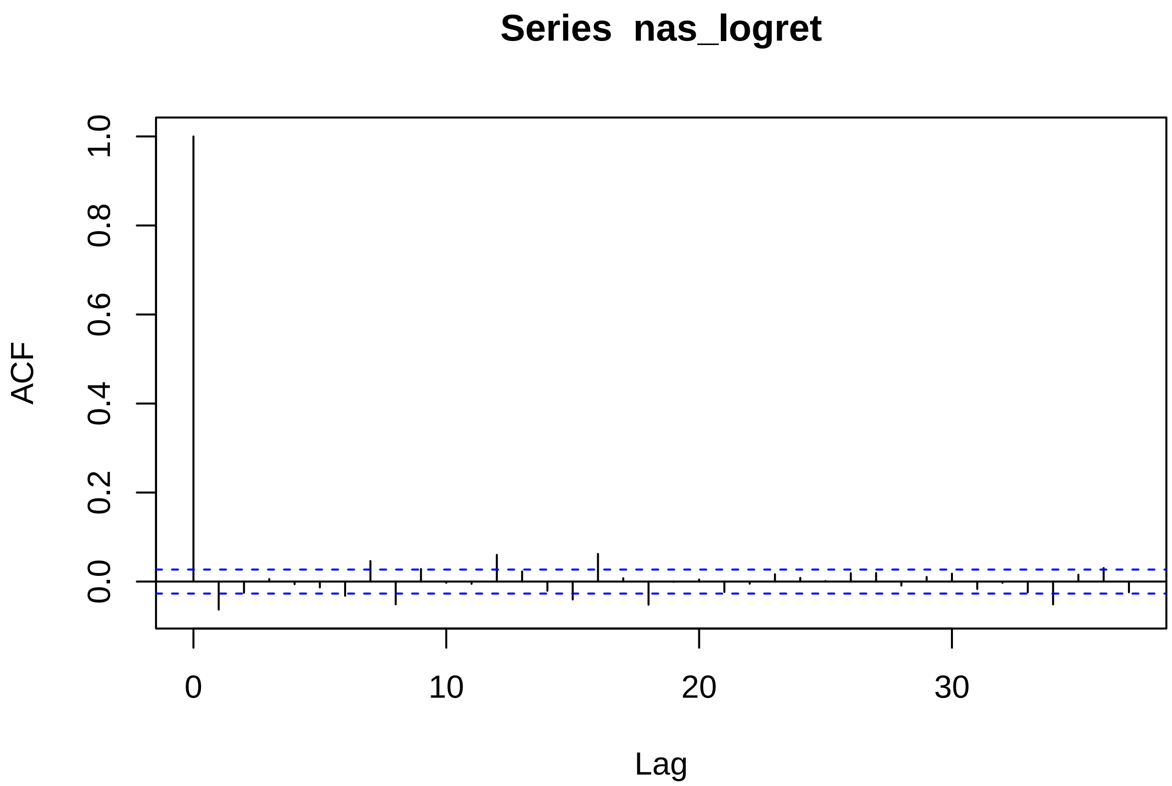
```
acf(sp_logret)
```



```
acf(ny_logret)
```



```
acf(nas_logret)
```



As you can see above there is serious correlation on the first lag, again confirm that our series is not stationary.

So instead, we will build a mean equation and try to convert our residuals into a stationary white noise.

Detecting change points in our log-returns and adjusting our data

Change points are intervals within our data in which the mean is different than the mean of the rest of the data. We have used H. Cho and P. Fryzlewicz (2021)'s algorithm to detect change points in our data (the code can be found here) and remove these change points from our data.

```r
source('main.R')

sp_changepoint<-wcm.gsa(sp_logret, double.cusum = TRUE)


mean_sp <- sp_logret * 0
position <- c(0, sp_changepoint$cp, length(sp_logret))
for(i in 1:(length(sp_changepoint$cp) + 1)){
  int <- (position[i] + 1):position[i + 1]
  mean_sp[int] <- mean(sp_logret[int])
}

sp_logret_changepoint <- sp_logret - mean_sp
```

```r
ny_changepoint<-wcm.gsa(ny_logret, double.cusum = TRUE)


mean_ny <- ny_logret * 0
position <- c(0, ny_changepoint$cp, length(ny_logret))
for(i in 1:(length(ny_changepoint$cp) + 1)){
  int <- (position[i] + 1):position[i + 1]
  mean_ny[int] <- mean(ny_logret[int])
}

ny_logret_changepoint <- ny_logret - mean_ny
```

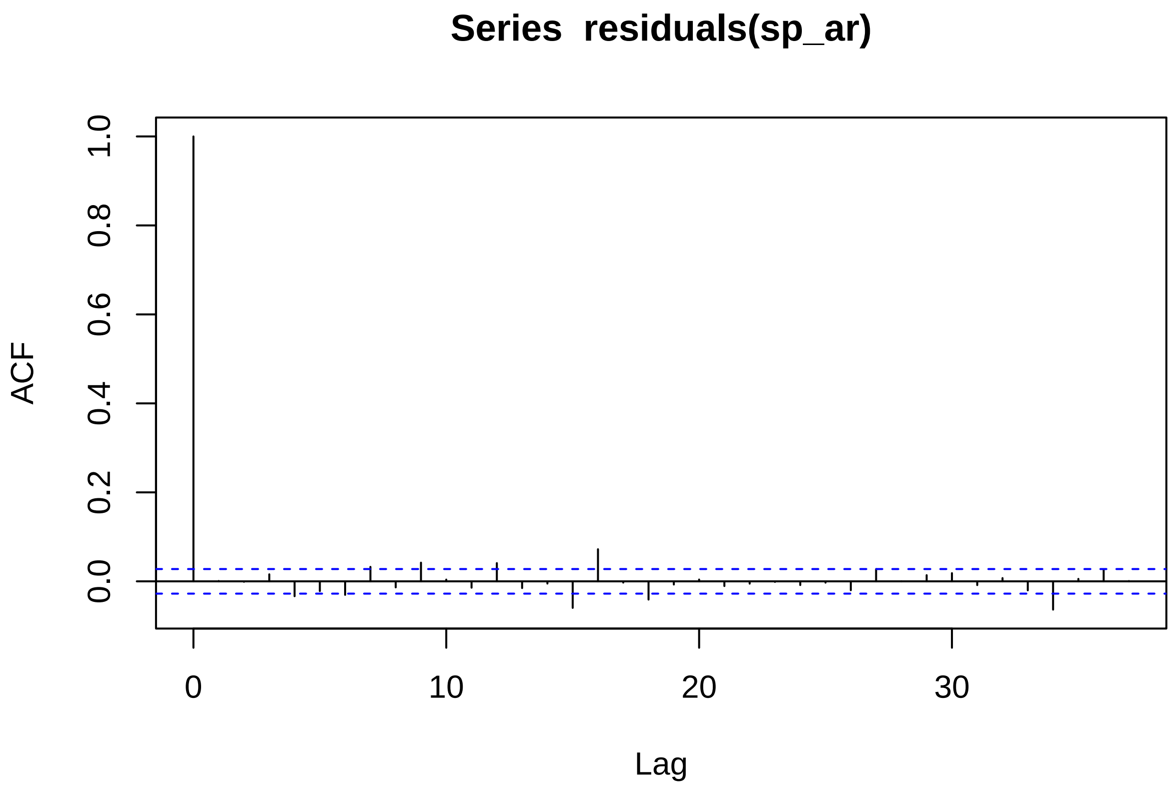
```r
nas_changepoint<-wcm.gsa(nas_logret, double.cusum = TRUE)


mean_nas <- nas_logret * 0
position <- c(0, nas_changepoint$cp, length(nas_logret))
for(i in 1:(length(nas_changepoint$cp) + 1)){
  int <- (position[i] + 1):position[i + 1]
  mean_nas[int] <- mean(nas_logret[int])
}

nas_logret_changepoint <- nas_logret - mean_nas
```

Building a mean-equation

```r
sp_ar <- arima(sp_logret, order = c(1, 0, 1))
sp_ar
```

```
##
## Call:
## arima(x = sp_logret, order = c(1, 0, 1))
##
## Coefficients:
##           ar1      ma1  intercept
##       -0.0631  -0.0578      2e-04
## s.e.   0.1056   0.1056      2e-04
##
## sigma^2 estimated as 0.0001533:  log likelihood = 14955.89,  aic = -29903.78
```

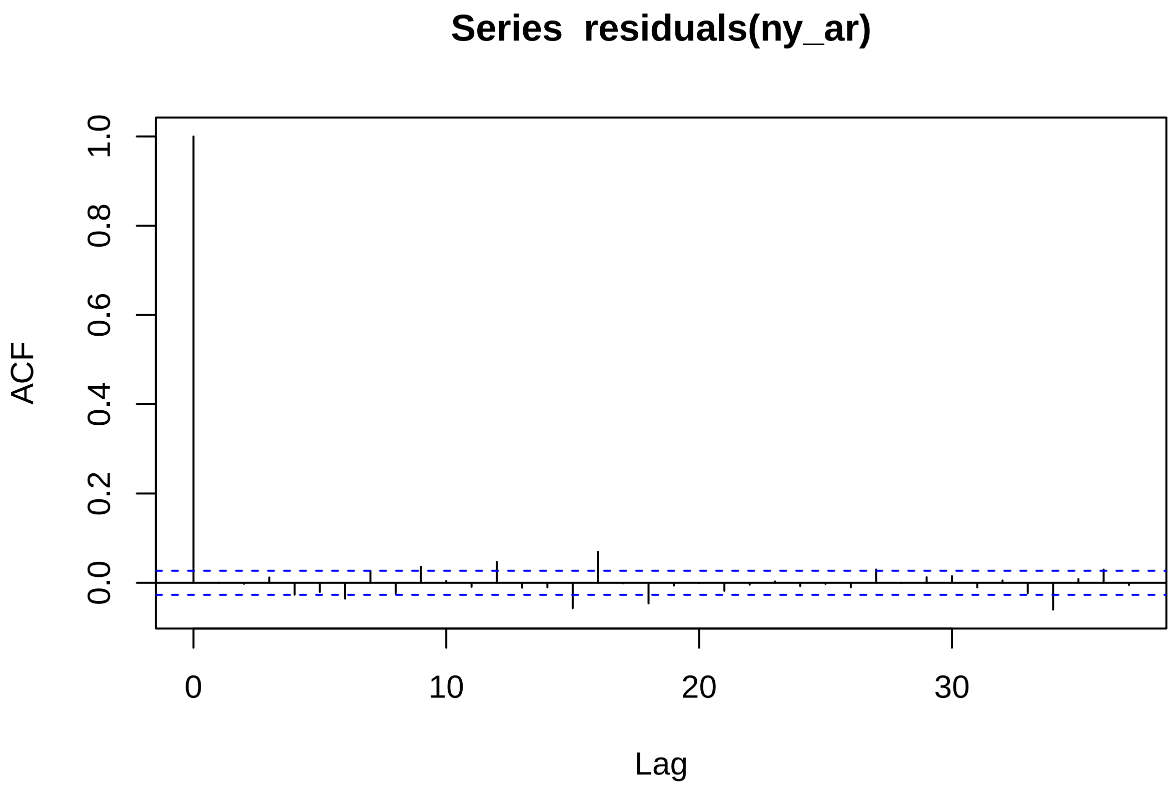
```r
acf(residuals(sp_ar))
```



```
ny_ar <- arima(ny_logret, order = c(1, 0, 1))
ny_ar
```

```
##
## Call:
## arima(x = ny_logret, order = c(1, 0, 1))
##
## Coefficients:
##           ar1      ma1  intercept
##       -0.0020  -0.1135      2e-04
## s.e.   0.1143   0.1136      2e-04
##
## sigma^2 estimated as 0.0001555:  log likelihood = 15667.19,  aic = -31326.38
```

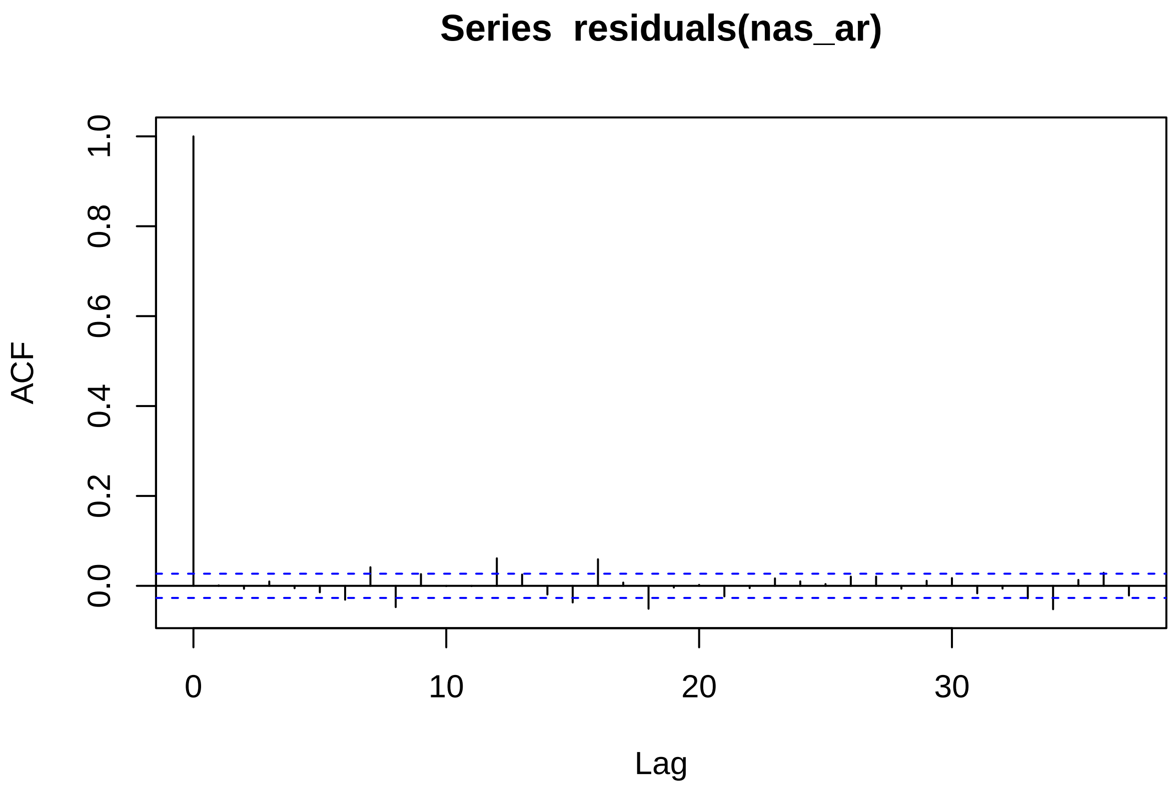
```
acf(residuals(ny_ar))
```



```
nas_ar <- arima(nas_logret, order = c(1, 0, 1))
nas_ar
```

```
##
## Call:
## arima(x = nas_logret, order = c(1, 0, 1))
##
## Coefficients:
##          ar1      ma1  intercept
##       0.2811  -0.3470      2e-04
## s.e.  0.2102   0.2065      2e-04
##
## sigma^2 estimated as 0.000255:  log likelihood = 14360.11,  aic = -28712.21
```

```
acf(residuals(nas_ar))
```



```
lag.length = 25

Box.test(residuals(sp_ar), lag=lag.length, type="Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  residuals(sp_ar)
## X-squared = 94.264, df = 25, p-value = 5.697e-10
```

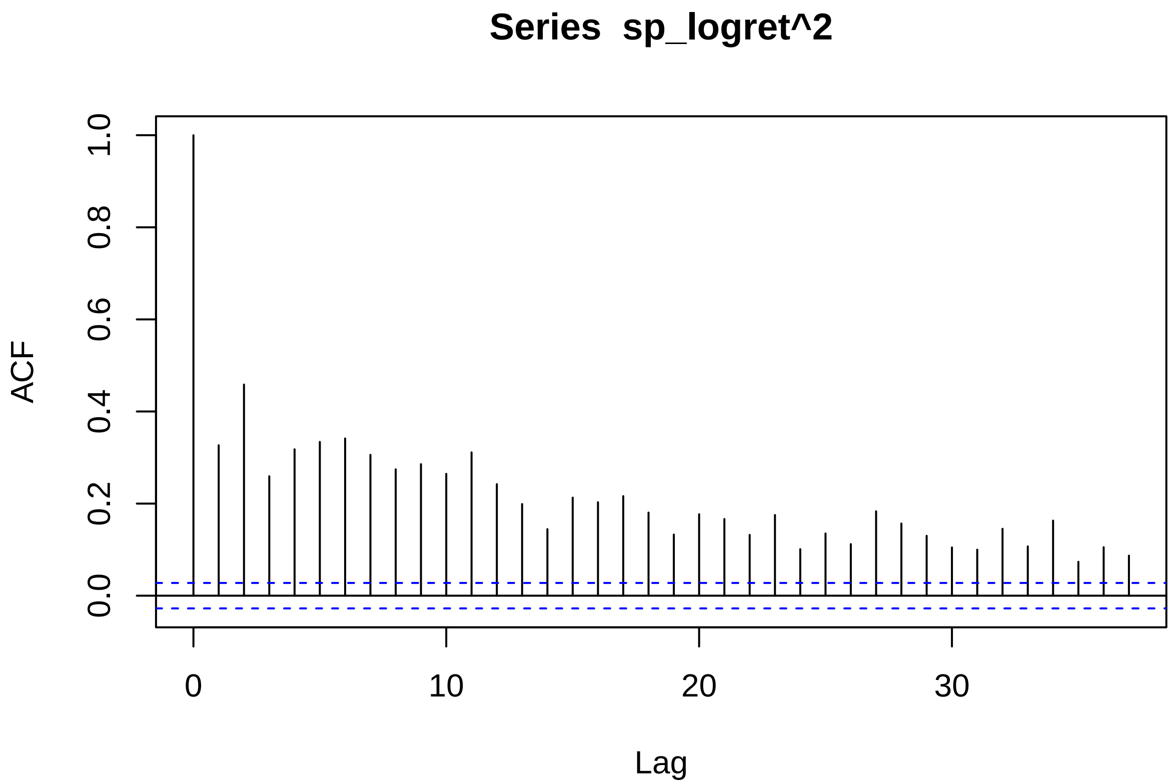
```
Box.test(residuals(ny_ar), lag=lag.length, type="Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  residuals(ny_ar)
## X-squared = 96.877, df = 25, p-value = 2.095e-10
```

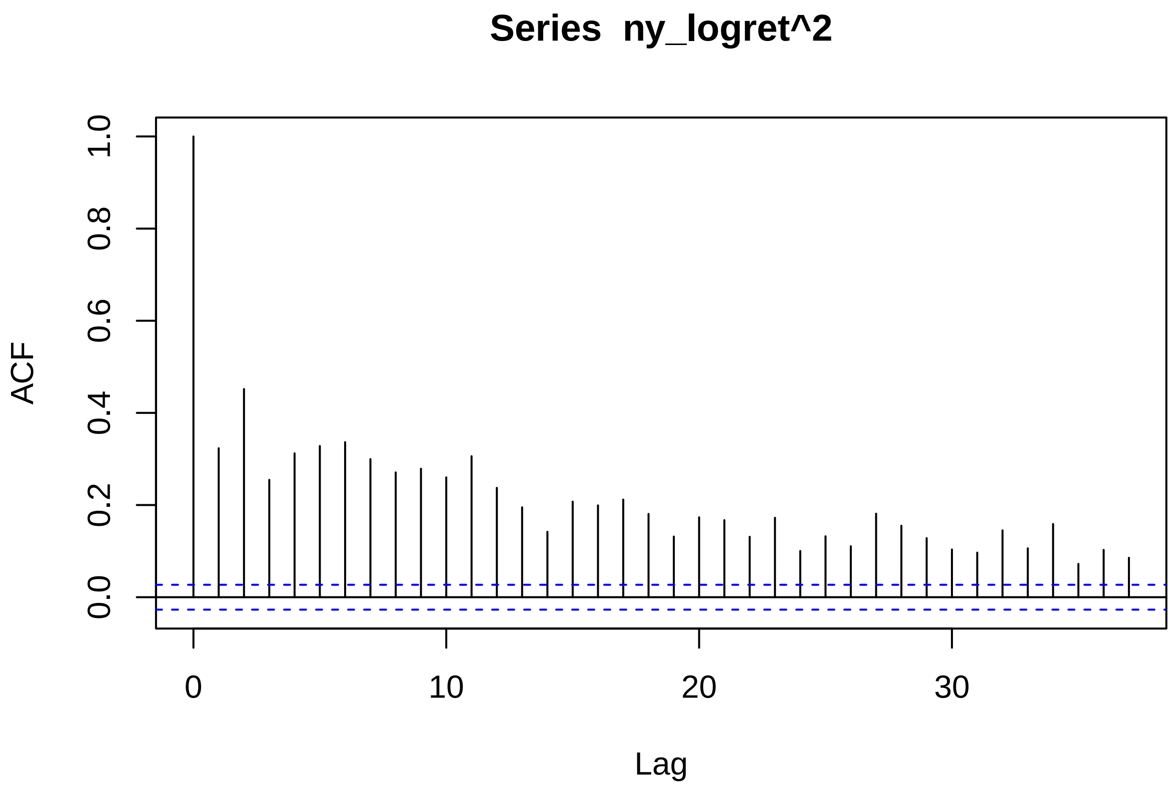
```
Box.test(residuals(nas_ar), lag=lag.length, type="Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  residuals(nas_ar)
## X-squared = 101.92, df = 25, p-value = 2.974e-11
```

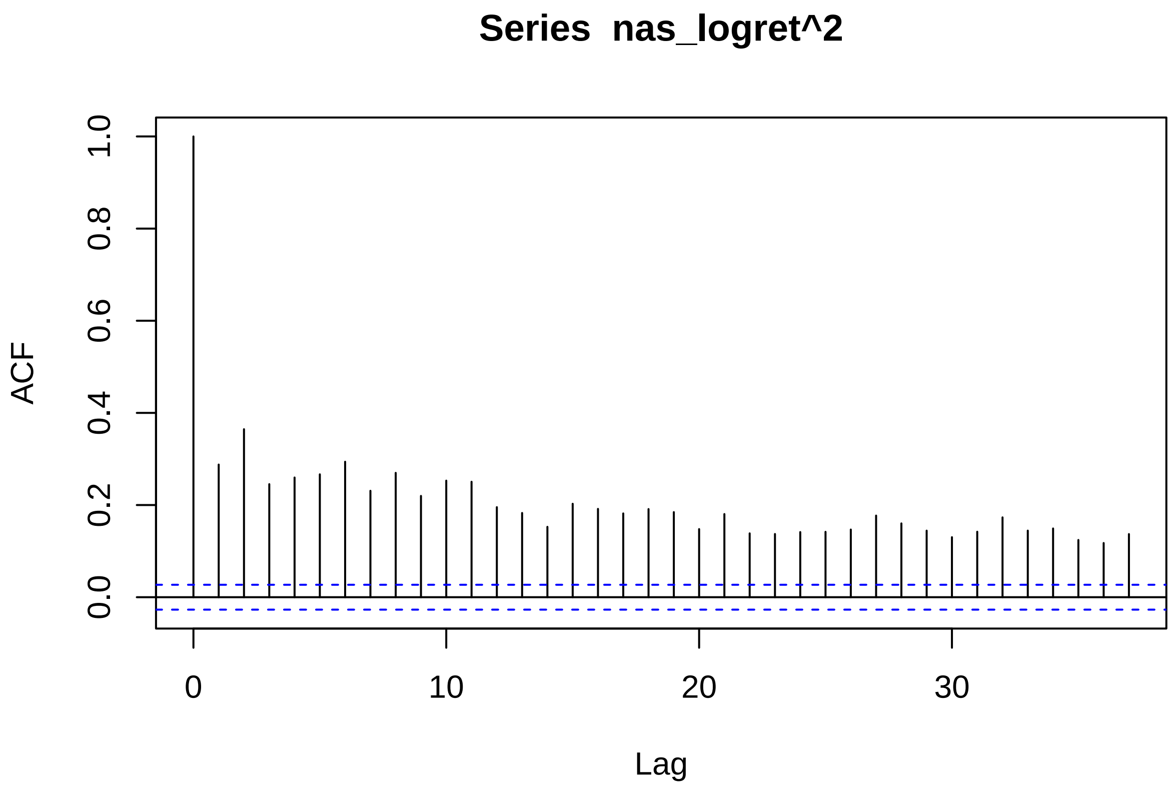
```
acf(sp_logret^2)
```



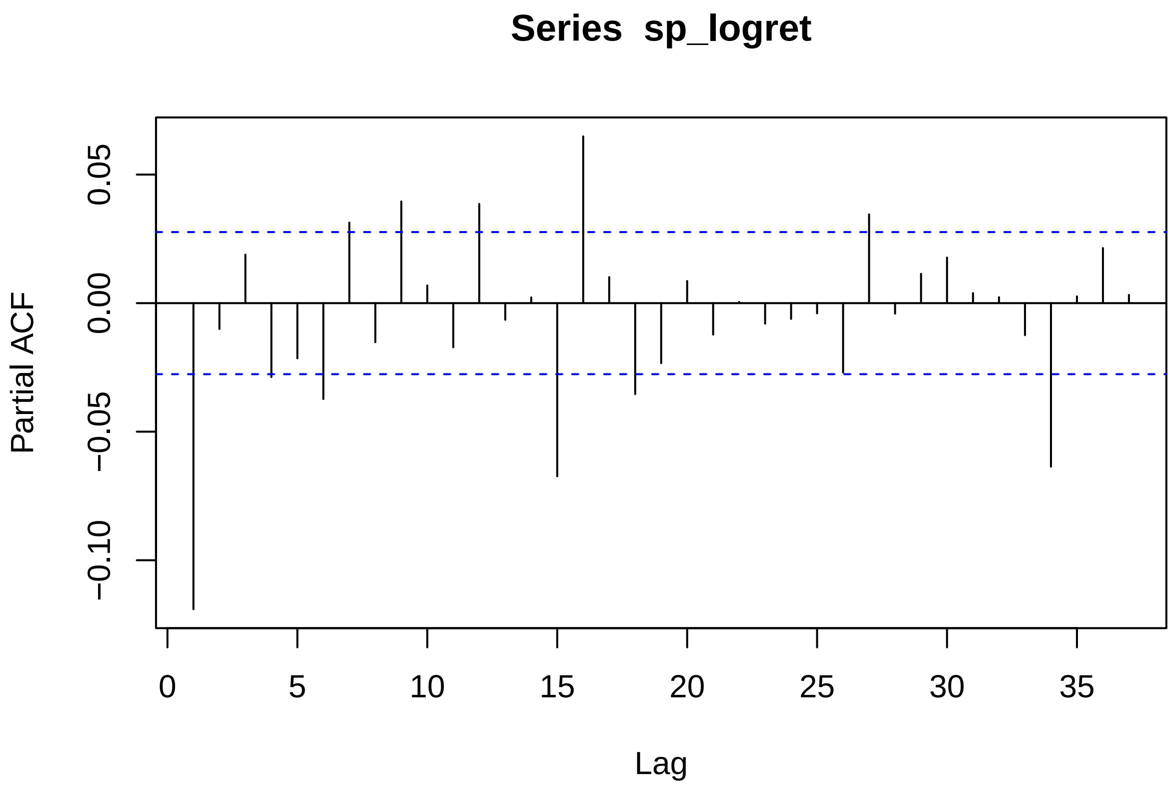
```
acf(ny_logret^2)
```



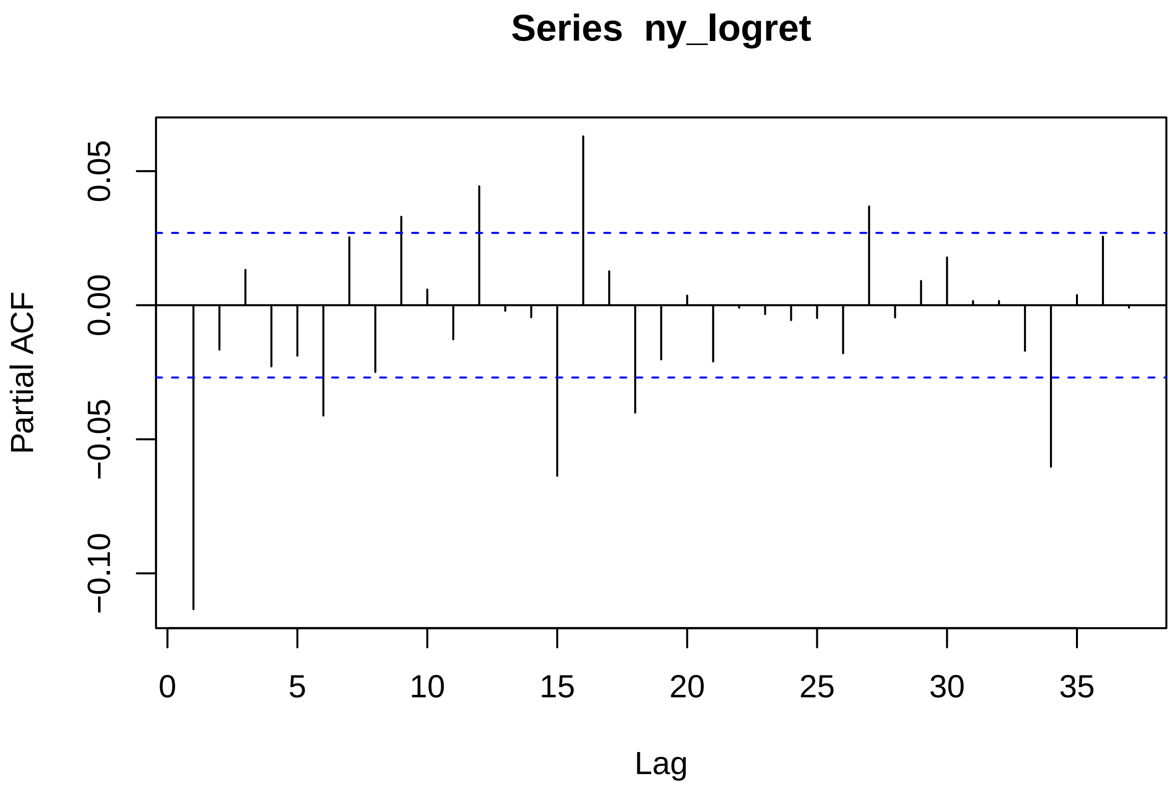
```
acf(nas_logret^2)
```



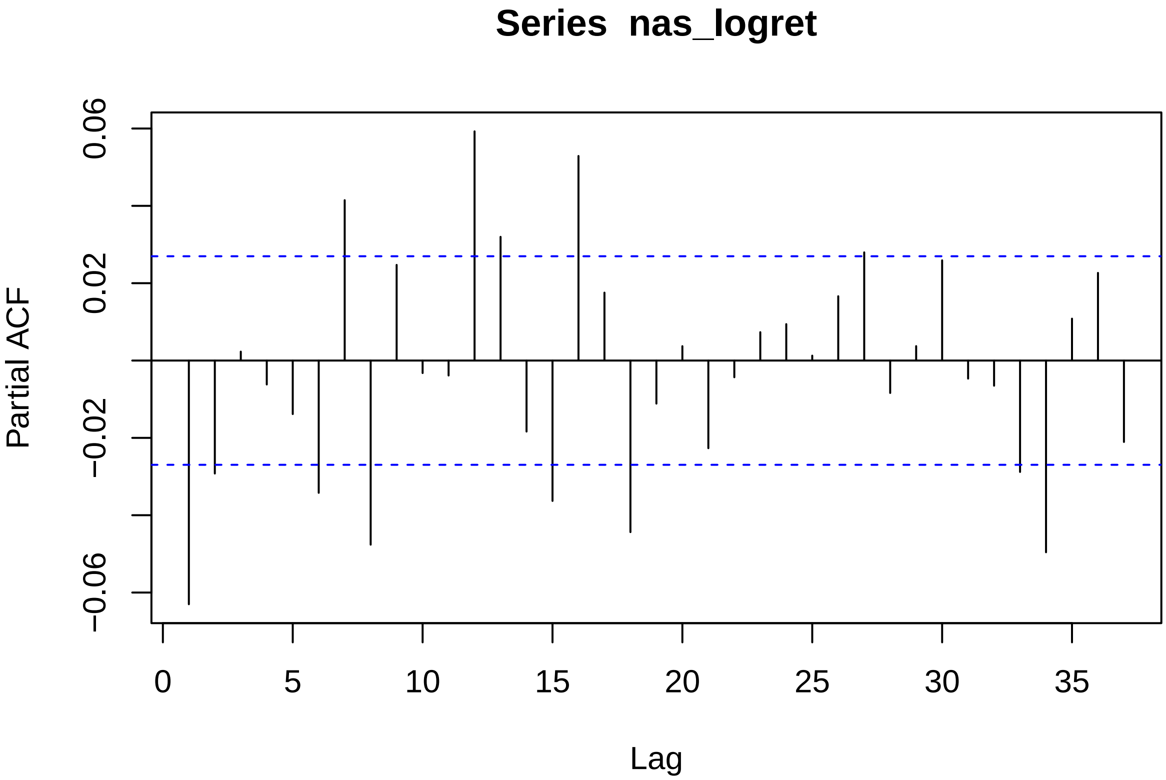
```
pacf(sp_logret)
```



```
pacf(ny_logret)
```



```
pacf(nas_logret)
```



Outputting Files

```
setwd('..')
write.csv(sp_logret_changepoint, 'Data/Processed/sp_logret_changepoint.csv', row.names=T)
write.csv(ny_logret_changepoint, 'Data/Processed/ny_logret_changepoint.csv', row.names=T)
write.csv(nas_logret_changepoint, 'Data/Processed/nas_logret_changepoint.csv', row.names=T)
```