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 2020:

- S&P500
- DOWJONES
- 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')
dow<-read.csv("Data/dowjones.csv")
nas<-read.csv("Data/nasdaq.csv")
```

Now we will rename some columns and fixing Date and number formats:

```
#Renaming columns
colnames(sp) <- c("Date","Price")
colnames(dow) <- c("Date","Price")
colnames(nas) <- c("Date","Price")

#Fixing the Date format
sp$Date<-as.Date(sp$Date, format="%d/%m/%Y")
dow$Date<-as.Date(dow$Date, format="%d/%m/%Y")
nas$Date<-as.Date(nas$Date, format="%d/%m/%Y")

#Fixing numeric format
dow$Price <- as.numeric(gsub(",", "", dow$Price))

#Getting an overall idea of our datasets
str(sp)
```

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

```r
str(dow)
```

```
## 'data.frame':	5286 obs. of  2 variables:
##  $ Date : Date, format: "2000-01-03" "2000-01-04" ...
##  $ Price: num  11358 10998 11123 11253 11523 ...
```

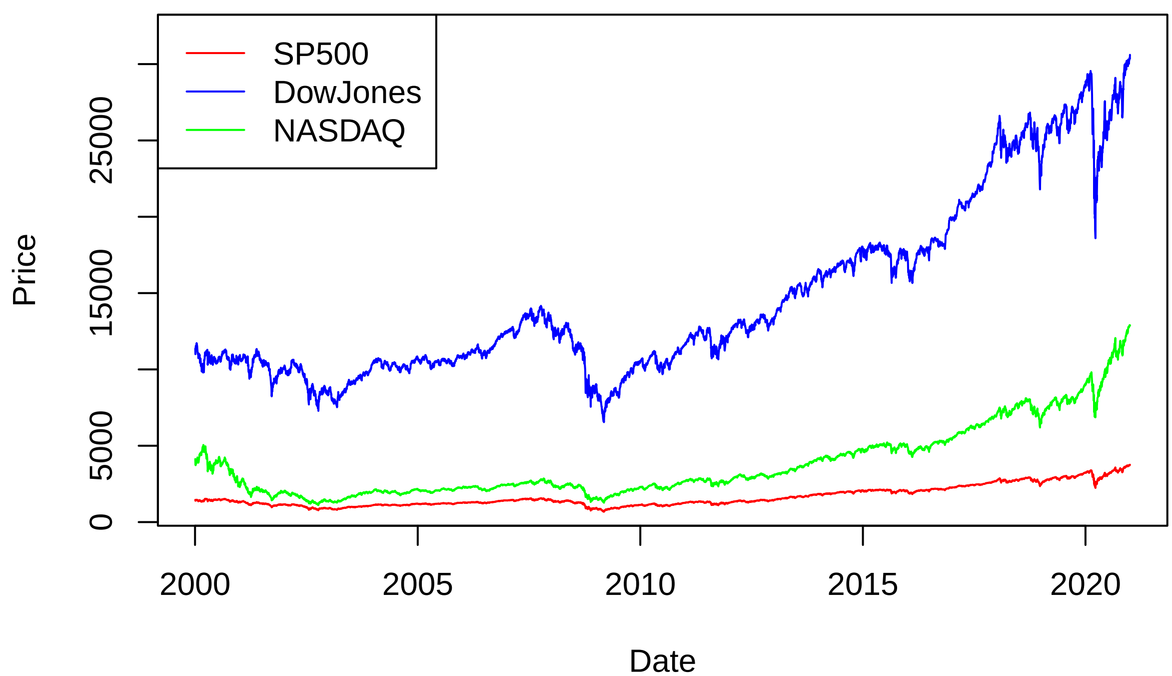
```r
str(nas)
```

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

Initial Plots

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

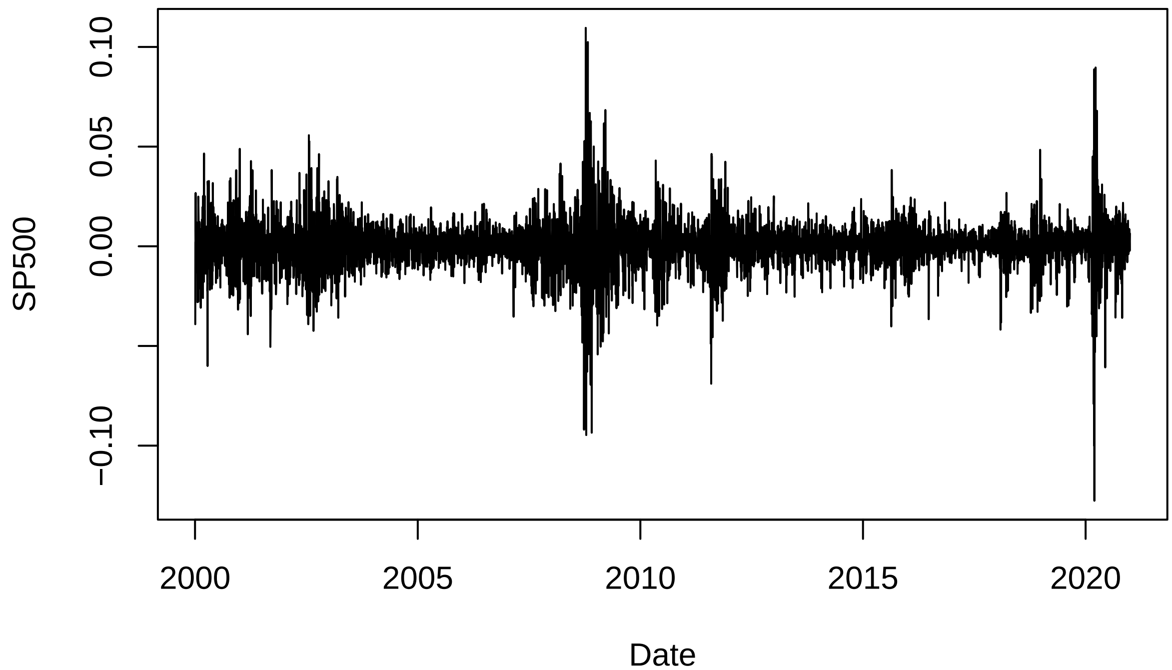
```r
#Plotting Price against Date for each index
plot(sp, col='red',type='l',ylim=c(1000,32000))
lines(dow, col='blue')
lines(nas, col='green')
legend("topleft", legend=c("SP500", "DowJones", "NASDAQ"),
       col=c("red", "blue", "green"), lty=c(1,1,1))
```



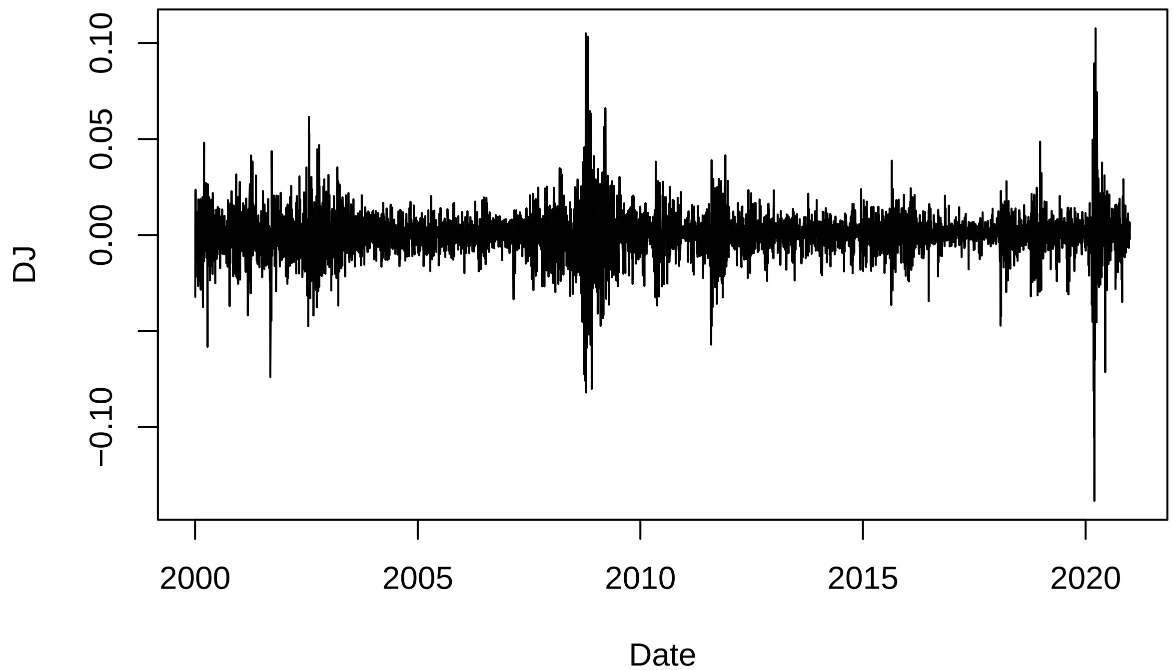
They all follow the same basic pattern, which is what we would expect, with the iconic fall in stock-price during the 2008-2009 period of the 'Great Recession'.

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

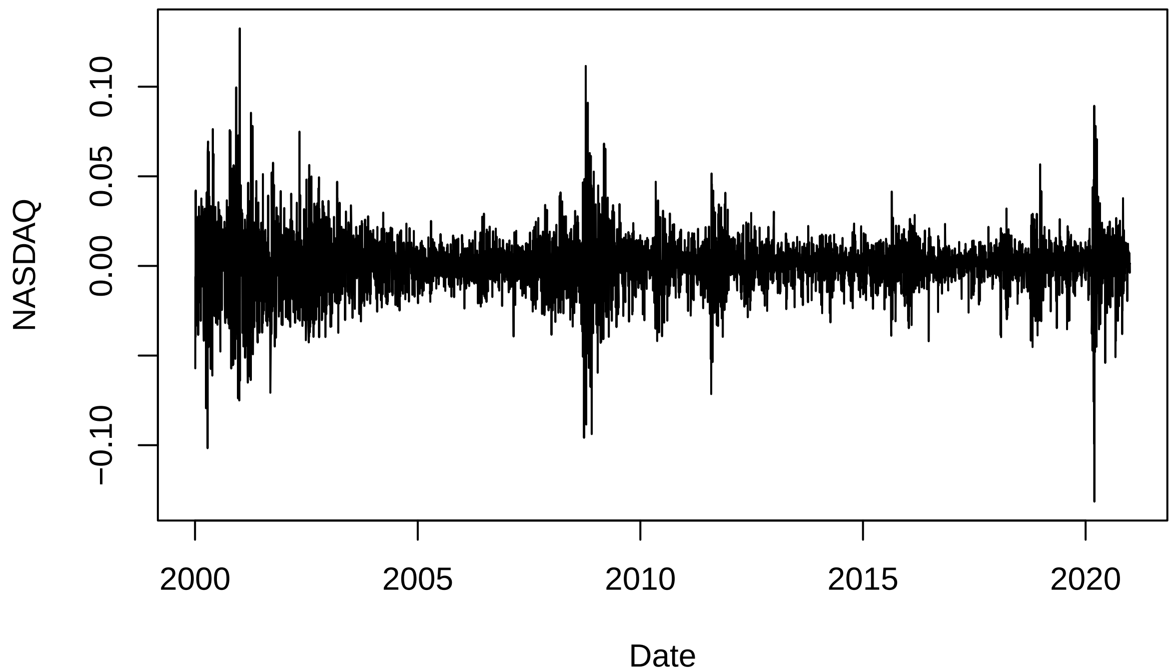
```
sp_logret <- diff(log(sp$Price))
dow_logret <- diff(log(dow$Price))
nas_logret <- diff(log(nas$Price))

plot(sp$Date[-length(sp$Date)],sp_logret,type='l',xlab="Date",ylab="SP500")
```




```
#title(main="LOG RETURN",cex.main=2.2)
plot(dow$Date[-length(dow$Date)],dow_logret, type='l',xlab="Date",ylab="DJ")
```



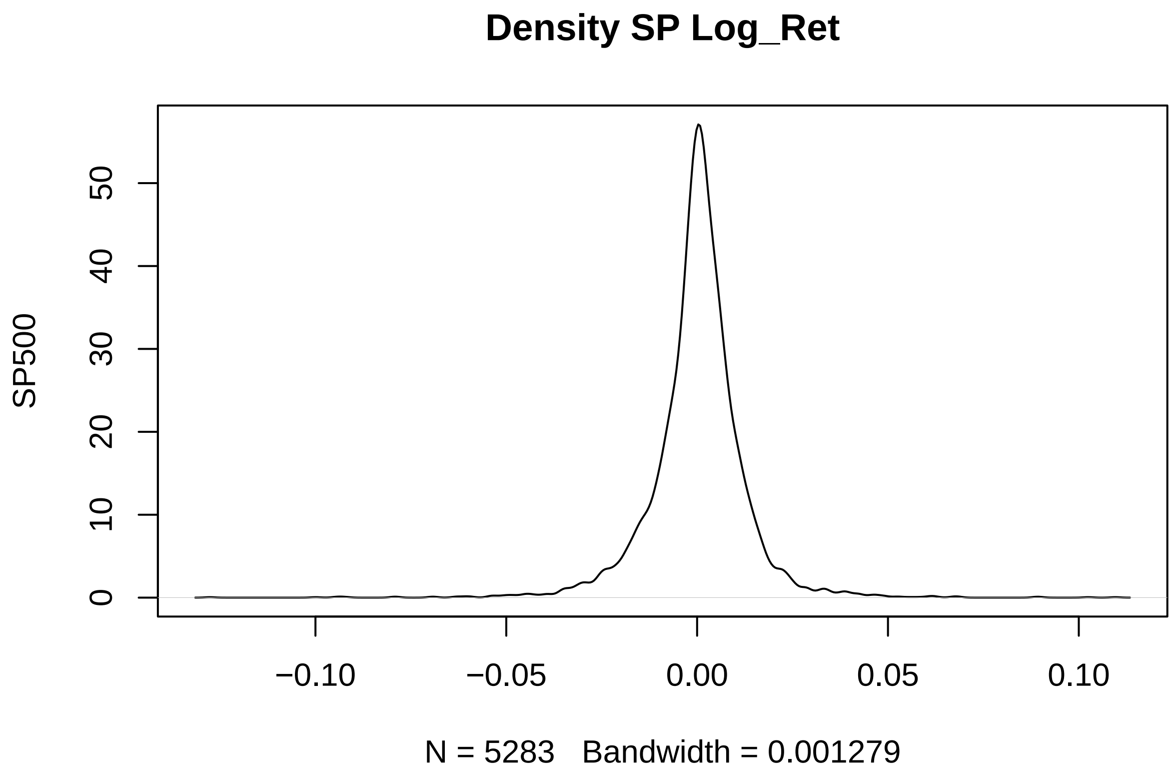

```
plot(nas$Date[-length(nas$Date)],nas_logret, type='l',xlab="Date",ylab="NASDAQ")
```



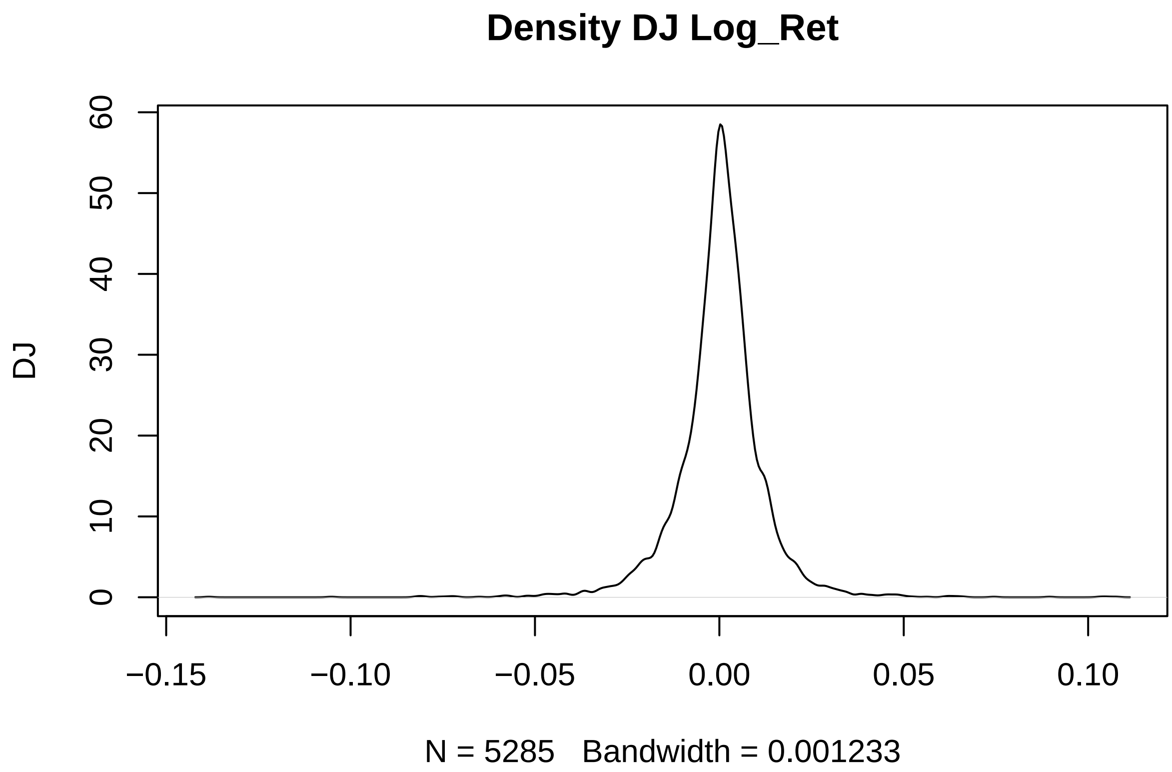
We can see that the returns average around 0% with very high variability during 2008-2009 (caused by the Great Recession) and during 2020 (caused by COVID-19).

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:

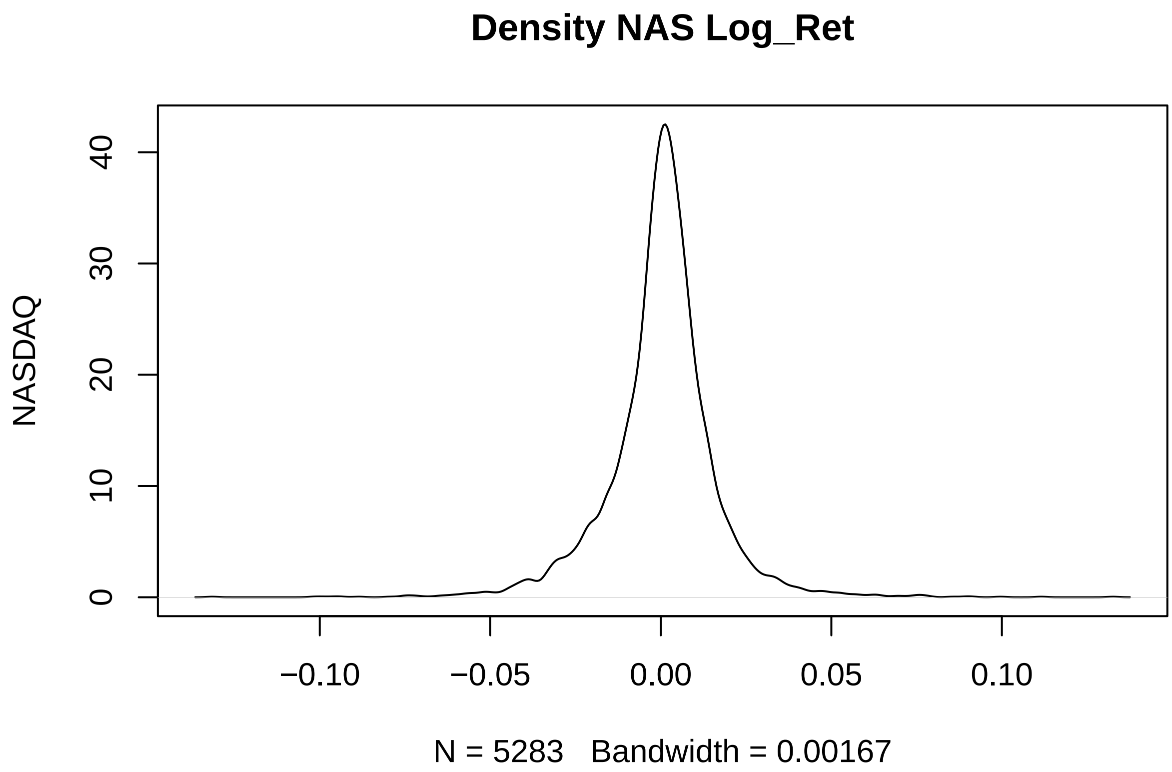
```
plot(density(sp_logret),ylab="SP500",main="Density SP Log_Ret")
```




```
plot(density(dow_logret),ylab="DJ",main="Density DJ Log_Ret")
```

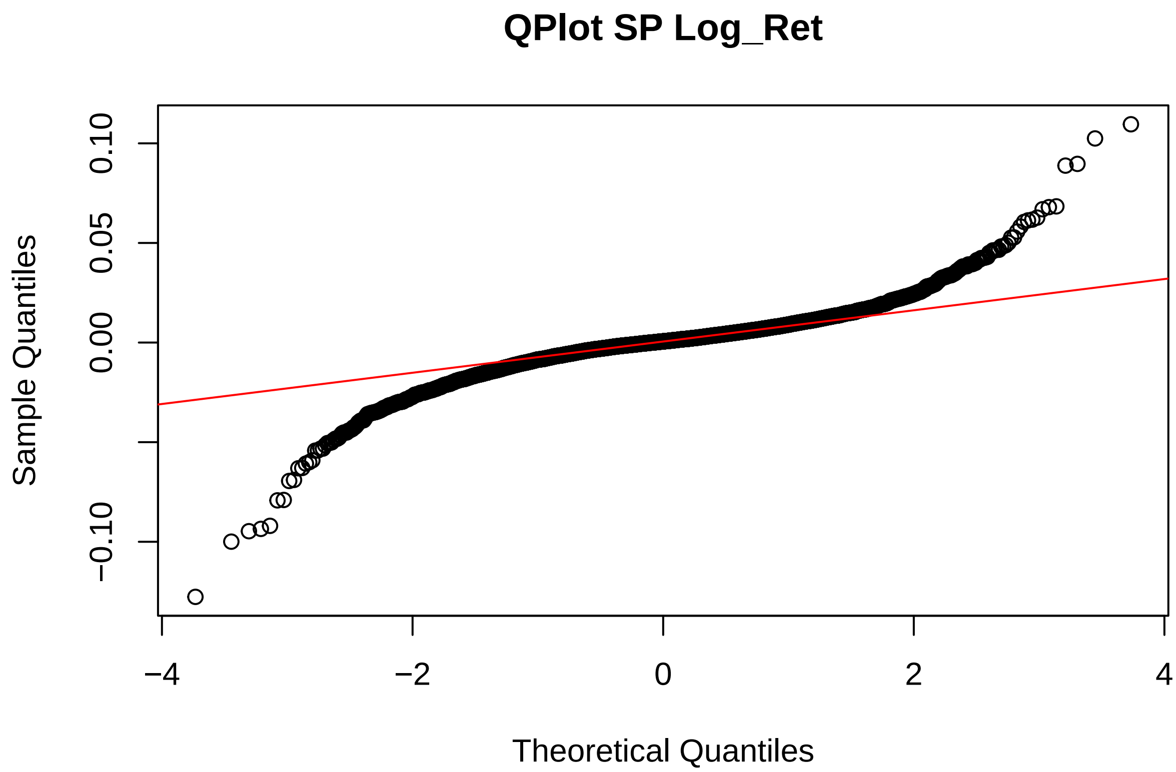



```
plot(density(nas_logret),ylab="NASDAQ",main="Density NAS Log_Ret")
```

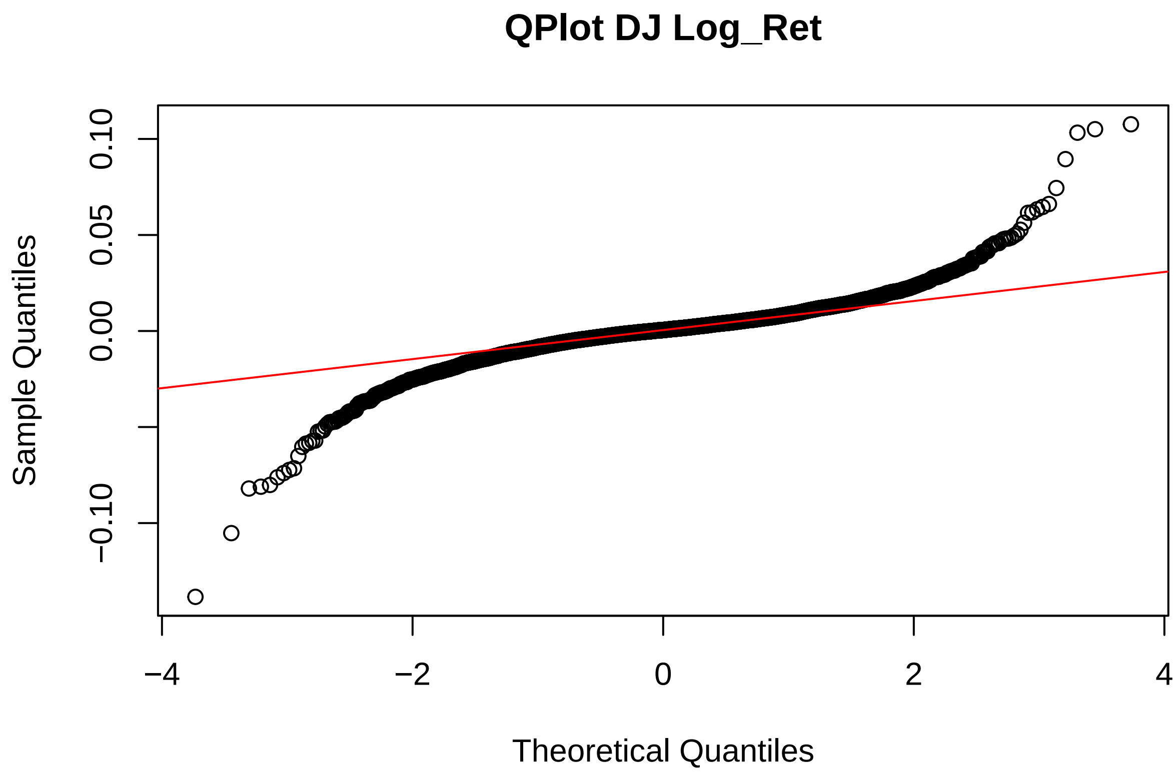


The returns look like they follow a normal distribution. So, we will make qq-plots to further confirm this:

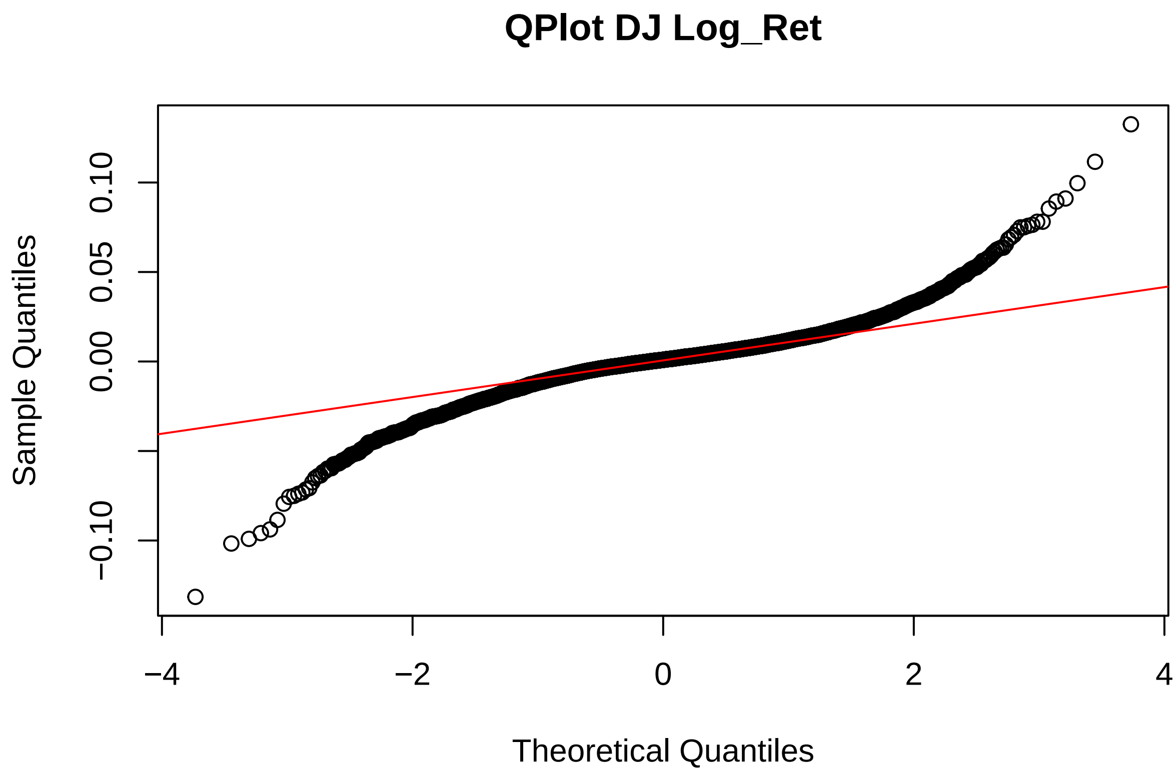
```
qqnorm(sp_logret,main="QPlot SP Log_Ret")
qqline(sp_logret,col='red')
```




```
qqnorm(dow_logret,main="QPlot DJ Log_Ret")
qqline(dow_logret,col='red')
```




```
qqnorm(nas_logret,main="QPlot DJ Log_Ret")
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 descriptive 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 -0.0047659  0.0005935  0.0001795  0.0058089  0.1095720
```

```
summary(dow_logret)
```

```
##        Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.1384181 -0.0046081  0.0005052  0.0001876  0.0055885  0.1076432
```

```
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:

```r
skewness(sp_logret)
```

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

```r
skewness(dow_logret)
```

```
## [1] -0.3770507
```

```r
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:

```r
kurtosis(sp_logret)
```

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

```r
kurtosis(dow_logret)
```

```
## [1] 16.02785
```

```r
kurtosis(nas_logret)
```

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

Let's also calculate the mean of our log returns as well:

```r
mean(sp_logret)
```

```
## [1] 0.0001794844
```

```r
mean(dow_logret)
```

```
## [1] 0.0001875747
```

```r
mean(nas_logret)
```

```
## [1] 0.000215363
```

```r
lag.length = 50

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

```
##
##  Box-Ljung test
##
## data:  sp_logret
## X-squared = 241.2, df = 50, p-value < 2.2e-16
```

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

```
##
##  Box-Ljung test
##
## data:  dow_logret
## X-squared = 250.45, df = 50, p-value < 2.2e-16
```

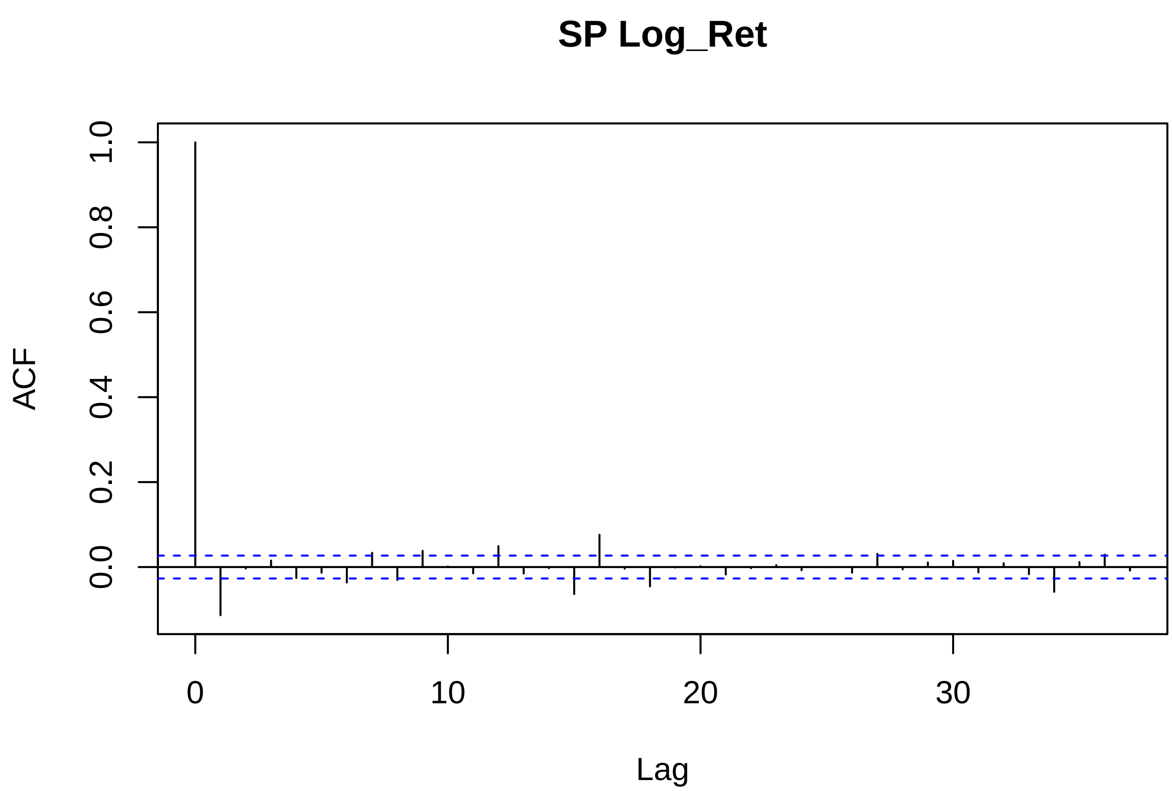
```r
Box.test(nas_logret, lag=lag.length, type="Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  nas_logret
## X-squared = 187.37, df = 50, p-value < 2.2e-16
```

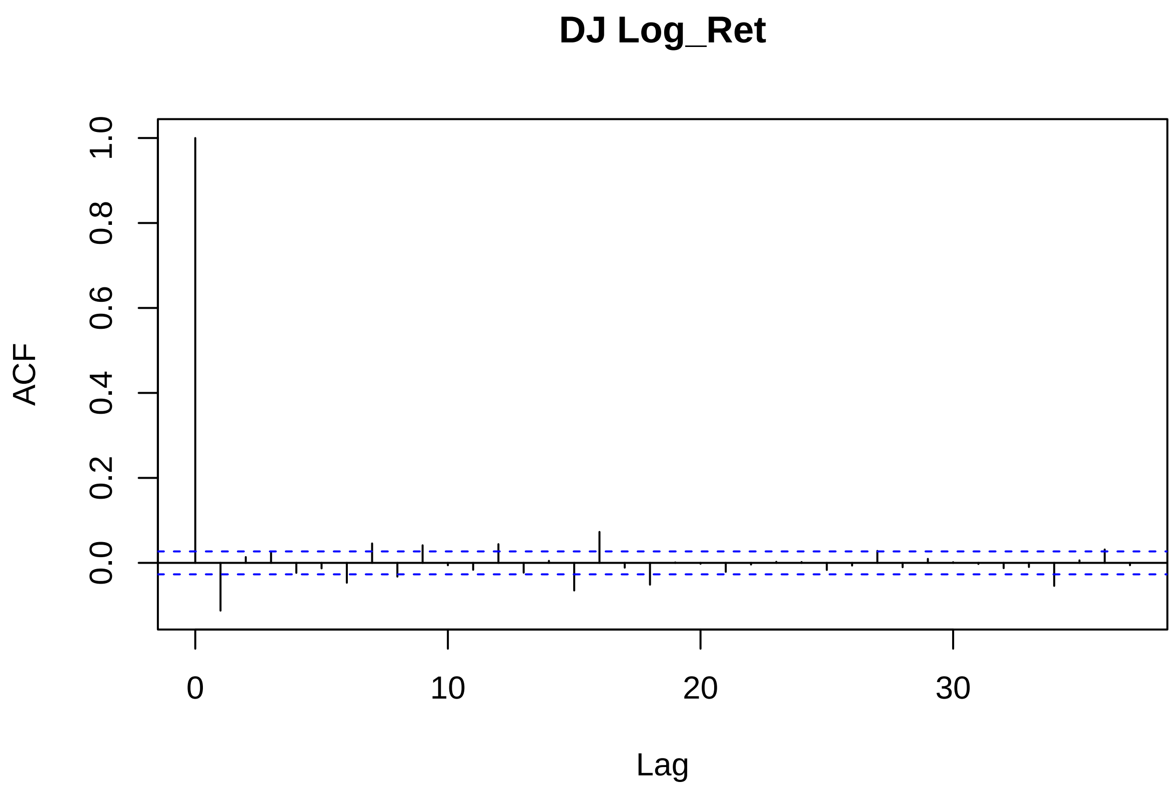
The p-value is very small which means we reject the null hypothesis that our correlations are 0. This means our data is not stationary and we might not use a GARCH model on log-returns directly.

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

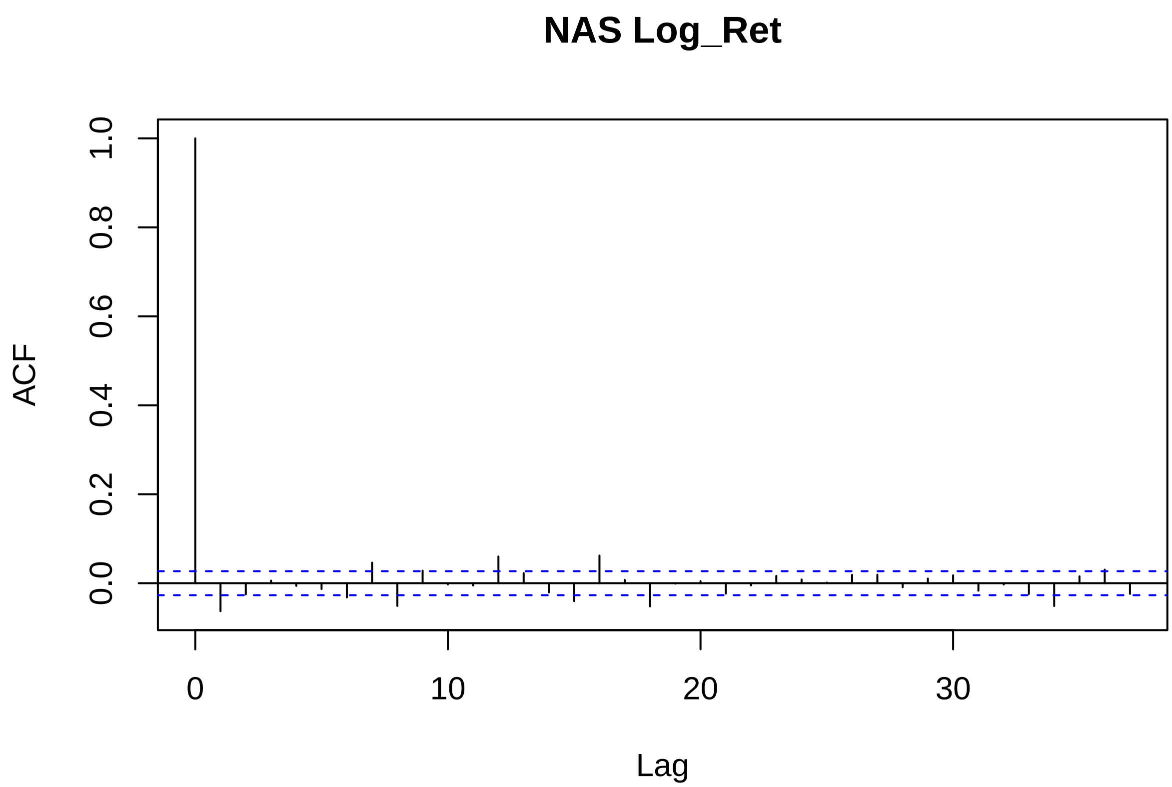
```r
acf(sp_logret,main="SP Log_Ret")
```



```r
acf(dow_logret,main="DJ Log_Ret")
```




```r
acf(nas_logret,main="NAS Log_Ret")
```



As you can see above there is serious correlation on the first lag, again confirm that our series is not stationary, hence we cannot apply a GARCH model directly on the log-returns.

This means we have to build a mean-equation in such a way that the residuals should be stationary. We will also check certain conditions on our residuals such as if they are normal.

Building a mean-equation

To build our mean equation we will be using auto.arima() which will automatically pick the parameters of the arima model that has the lowest AIC.

```
sp_ar <- auto.arima(sp_logret , max.order = c(3 , 0 ,3) , trace = T,  max.d = 0, ic = 'aic')
```

```
##
##  Fitting models using approximations to speed things up...
##
##  ARIMA(2,0,2) with non-zero mean : -31331.26
##  ARIMA(0,0,0) with non-zero mean : -31260.69
##  ARIMA(1,0,0) with non-zero mean : -31335.99
##  ARIMA(0,0,1) with non-zero mean : -31328.27
##  ARIMA(0,0,0) with zero mean     : -31261.61
##  ARIMA(2,0,0) with non-zero mean : -31334.5
##  ARIMA(1,0,1) with non-zero mean : -31335.21
##  ARIMA(2,0,1) with non-zero mean : Inf
##  ARIMA(1,0,0) with zero mean     : -31336.53
##  ARIMA(2,0,0) with zero mean     : -31334.98
```

```
##  ARIMA(1,0,1) with zero mean     : -31335.66
##  ARIMA(0,0,1) with zero mean     : -31328.86
##  ARIMA(2,0,1) with zero mean     : Inf
##
##  Now re-fitting the best model(s) without approximations...
##
##  ARIMA(1,0,0) with zero mean     : -31327.81
##
##  Best model: ARIMA(1,0,0) with zero mean
```

```
dow_ar <- auto.arima(dow_logret , max.order = c(3 , 0 ,3) , trace = T ,   max.d = 0, ic = 'aic')
```

```
##
##  Fitting models using approximations to speed things up...
##
##  ARIMA(2,0,2) with non-zero mean : -31752.93
##  ARIMA(0,0,0) with non-zero mean : -31681.66
##  ARIMA(1,0,0) with non-zero mean : -31753.13
##  ARIMA(0,0,1) with non-zero mean : -31745.56
##  ARIMA(0,0,0) with zero mean     : -31682.39
##  ARIMA(2,0,0) with non-zero mean : -31750.52
##  ARIMA(1,0,1) with non-zero mean : -31751.17
##  ARIMA(2,0,1) with non-zero mean : Inf
##  ARIMA(1,0,0) with zero mean     : -31753.44
##  ARIMA(2,0,0) with zero mean     : -31750.86
##  ARIMA(1,0,1) with zero mean     : -31751.49
##  ARIMA(0,0,1) with zero mean     : -31745.92
##  ARIMA(2,0,1) with zero mean     : Inf
##
##  Now re-fitting the best model(s) without approximations...
##
##  ARIMA(1,0,0) with zero mean     : -31747.32
##
##  Best model: ARIMA(1,0,0) with zero mean
```

```
nas_ar <- auto.arima(nas_logret , max.order = c(3 , 0 ,3) , trace = T ,  max.d = 0, ic = 'aic')
```

```
##
##  Fitting models using approximations to speed things up...
##
##  ARIMA(2,0,2) with non-zero mean : -28726.35
##  ARIMA(0,0,0) with non-zero mean : -28691.02
##  ARIMA(1,0,0) with non-zero mean : -28721.96
##  ARIMA(0,0,1) with non-zero mean : -28711.24
##  ARIMA(0,0,0) with zero mean     : -28692.07
##  ARIMA(1,0,2) with non-zero mean : Inf
##  ARIMA(2,0,1) with non-zero mean : -28721.97
##  ARIMA(3,0,2) with non-zero mean : -28761.67
##  ARIMA(3,0,1) with non-zero mean : -28727.49
##  ARIMA(4,0,2) with non-zero mean : -28742.41
##  ARIMA(3,0,3) with non-zero mean : -28759.66
##  ARIMA(2,0,3) with non-zero mean : -28725.26
##  ARIMA(4,0,1) with non-zero mean : -28732.14
```

```
##  ARIMA(4,0,3) with non-zero mean : -28740.4
##  ARIMA(3,0,2) with zero mean     : -28762.62
##  ARIMA(2,0,2) with zero mean     : -28727.05
##  ARIMA(3,0,1) with zero mean     : -28728.18
##  ARIMA(4,0,2) with zero mean     : -28743.16
##  ARIMA(3,0,3) with zero mean     : Inf
##  ARIMA(2,0,1) with zero mean     : -28722.68
##  ARIMA(2,0,3) with zero mean     : -28726.03
##  ARIMA(4,0,1) with zero mean     : -28733.01
##  ARIMA(4,0,3) with zero mean     : -28741.93
##
##  Now re-fitting the best model(s) without approximations...
##
##  ARIMA(3,0,2) with zero mean     : -28722.43
##
##  Best model: ARIMA(3,0,2) with zero mean
```

So the best models are the following:

- **sp:** AR(1)
- **dow:** AR(1)
- **nas:** ARMA(3,0,2)

You can see more detailed info below:

```
sp_ar
```

```
## Series: sp_logret
## ARIMA(1,0,0) with zero mean
##
## Coefficients:
##           ar1
##       -0.1133
## s.e.   0.0137
##
## sigma^2 estimated as 0.0001556:  log likelihood=15665.9
## AIC=-31327.81   AICc=-31327.81   BIC=-31314.66
```

```
dow_ar
```

```
## Series: dow_logret
## ARIMA(1,0,0) with zero mean
##
## Coefficients:
##           ar1
##       -0.1123
## s.e.   0.0137
##
## sigma^2 estimated as 0.000144:  log likelihood=15875.66
## AIC=-31747.32   AICc=-31747.32   BIC=-31734.18
```

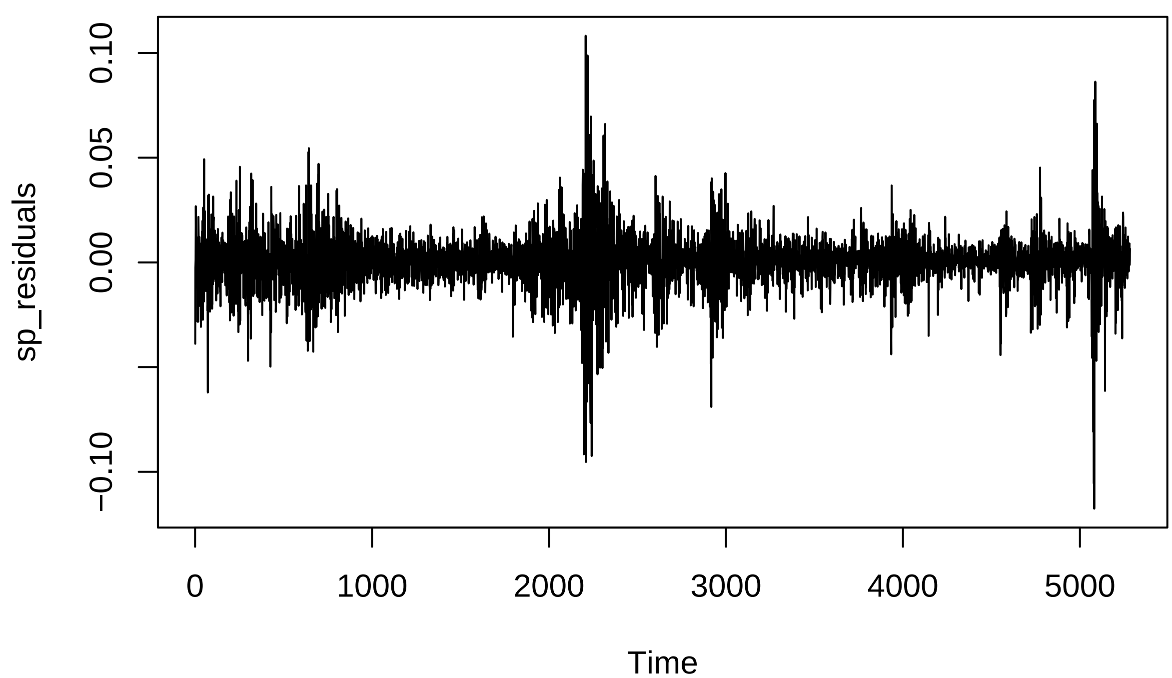
```
nas_ar
```

```
## Series: nas_logret
## ARIMA(3,0,2) with zero mean
##
## Coefficients:
##          ar1      ar2      ar3     ma1     ma2
##      -0.2310  -0.9849  -0.0759  0.1679  0.9732
## s.e.  0.0187   0.0211   0.0143  0.0129  0.0205
##
## sigma^2 estimated as 0.0002546:  log likelihood=14367.21
## AIC=-28722.43   AICc=-28722.41   BIC=-28682.99
```

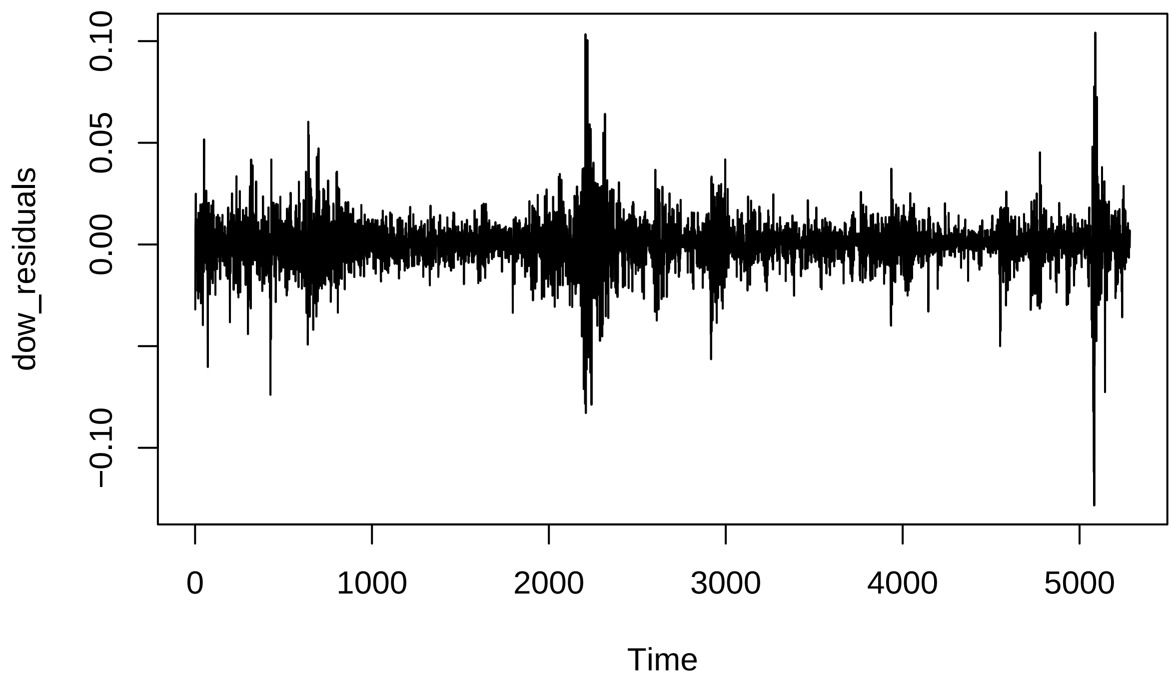
Doing diagnostic checks on residuals

Let's first take a look at how our residuals look:

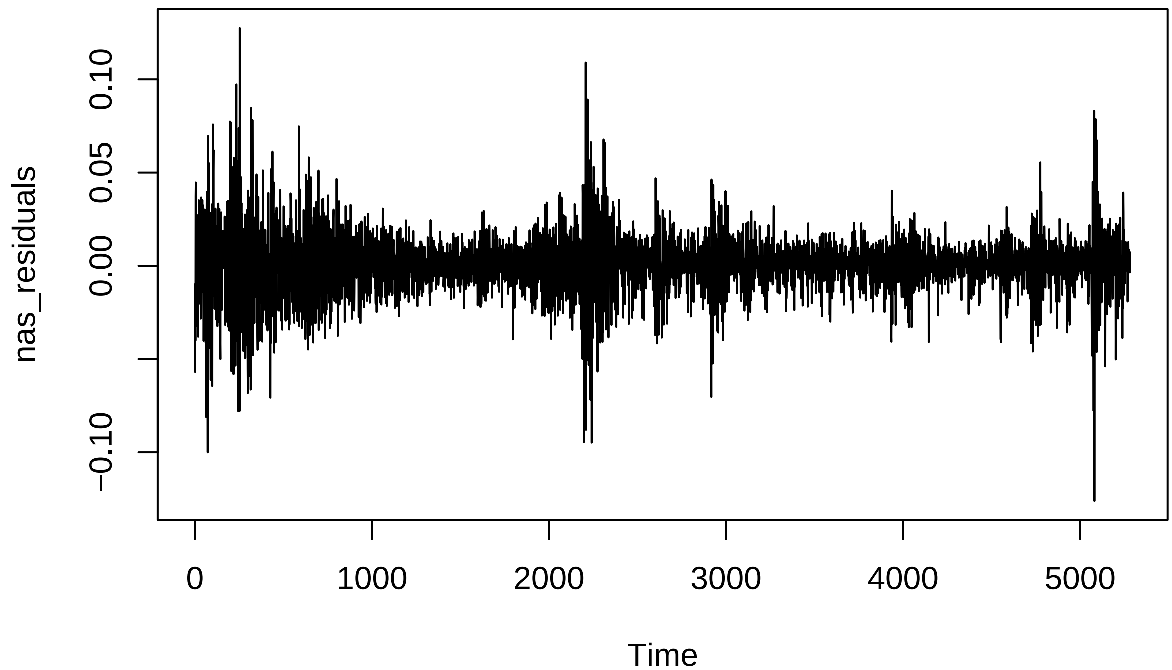
```
sp_residuals <- sp_ar$residuals
dow_residuals <- dow_ar$residuals
nas_residuals <- nas_ar$residuals

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



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

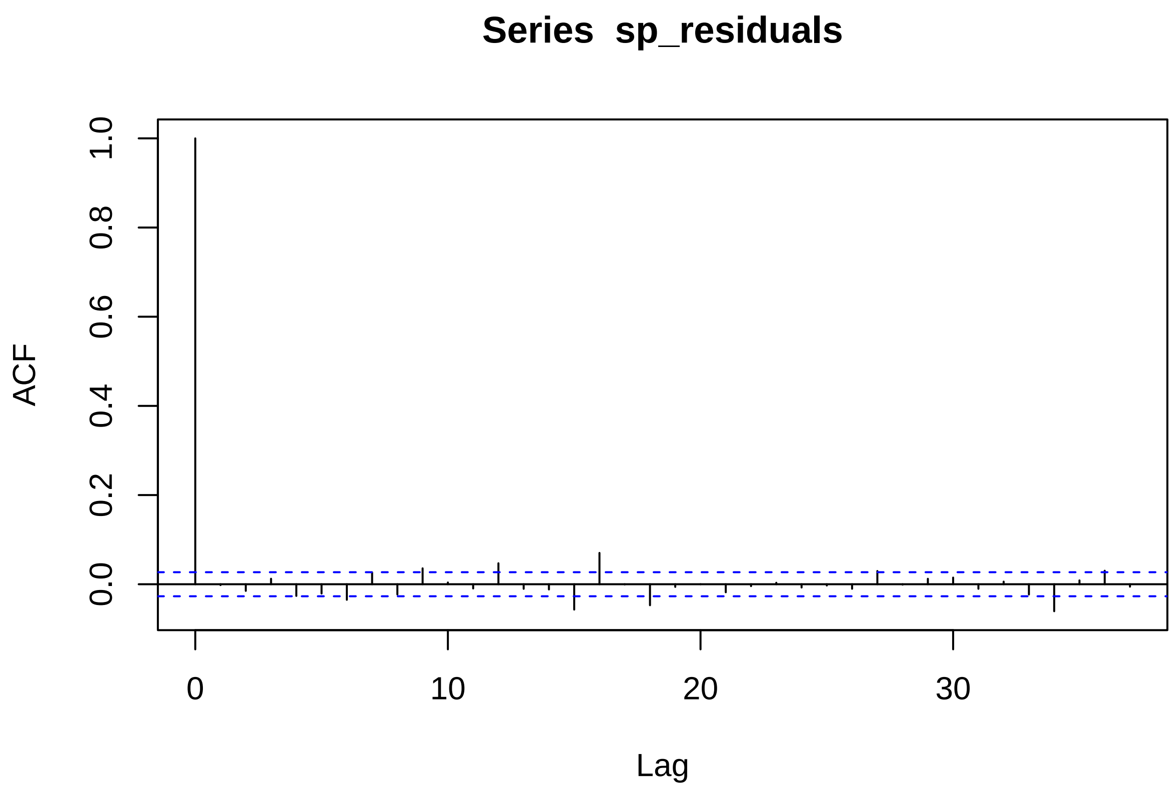



```
plot(nas_residuals, type='l')
```

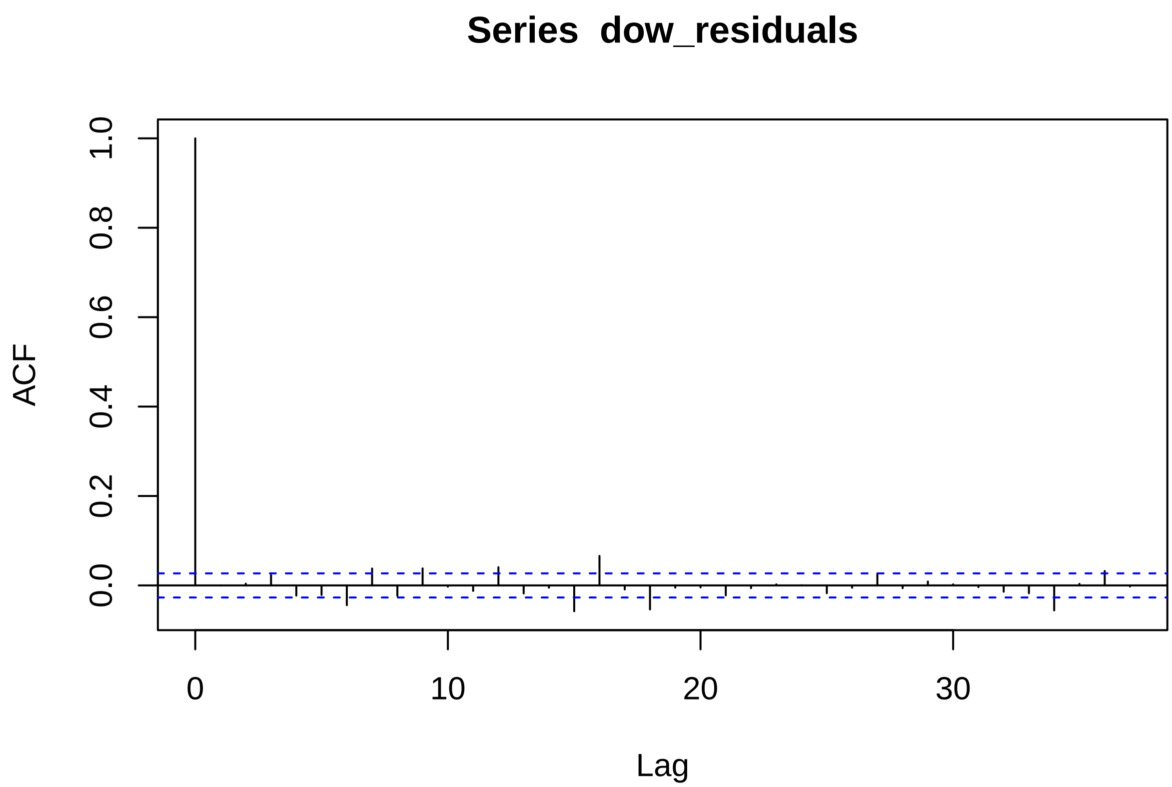


Let's plot ACF and PACF of our residuals.

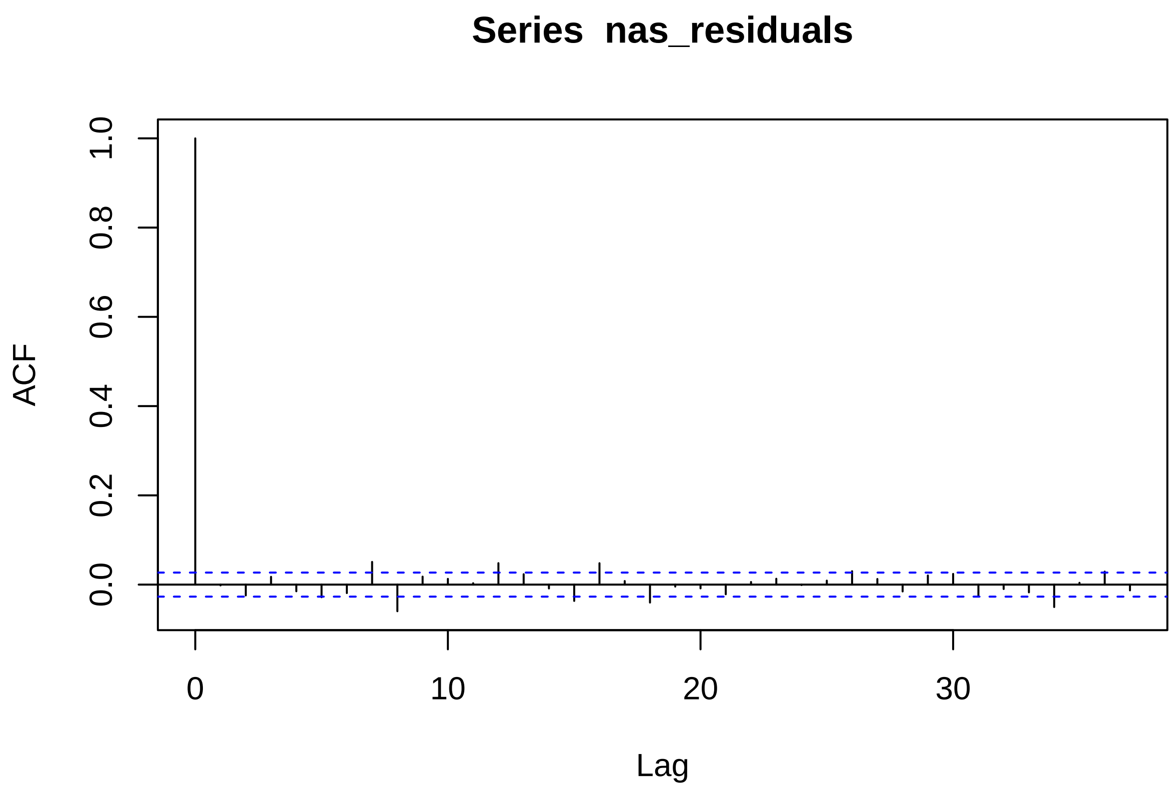
```
acf(sp_residuals)
```




```
acf(dow_residuals)
```




```
acf(nas_residuals)
```



Obtaining some statistics:

```
summary(sp_residuals)
```

```
##        Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.1175861 -0.0047386  0.0007728  0.0001997  0.0057078  0.1082312
```

```
summary(dow_residuals)
```

```
##        Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.1283713 -0.0046050  0.0006516  0.0002085  0.0056010  0.1041829
```

```
summary(nas_residuals)
```

```
##        Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.1261257 -0.0061818  0.0011725  0.0002306  0.0075714  0.1274755
```

```
skewness(sp_residuals)
```

```
## [1] -0.5213592
```

```r
skewness(dow_residuals)
```

```
## [1] -0.4862369
```

```r
skewness(nas_residuals)
```

```
## [1] -0.2016231
```

```r
kurtosis(sp_residuals)
```

```
## [1] 13.36874
```

```r
kurtosis(dow_residuals)
```

```
## [1] 15.17207
```

```r
kurtosis(nas_residuals)
```

```
## [1] 9.419078
```

```r
mean(sp_residuals)
```

```
## [1] 0.0001997381
```

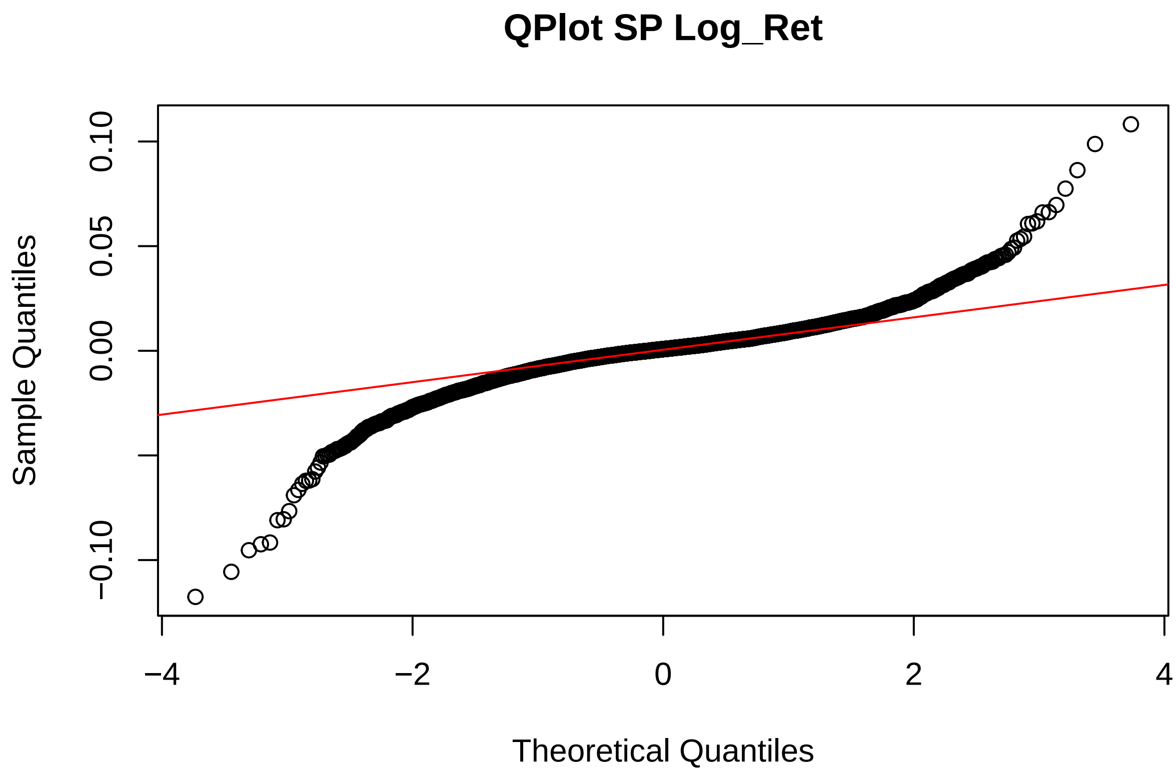
```r
mean(dow_residuals)
```

```
## [1] 0.0002085319
```

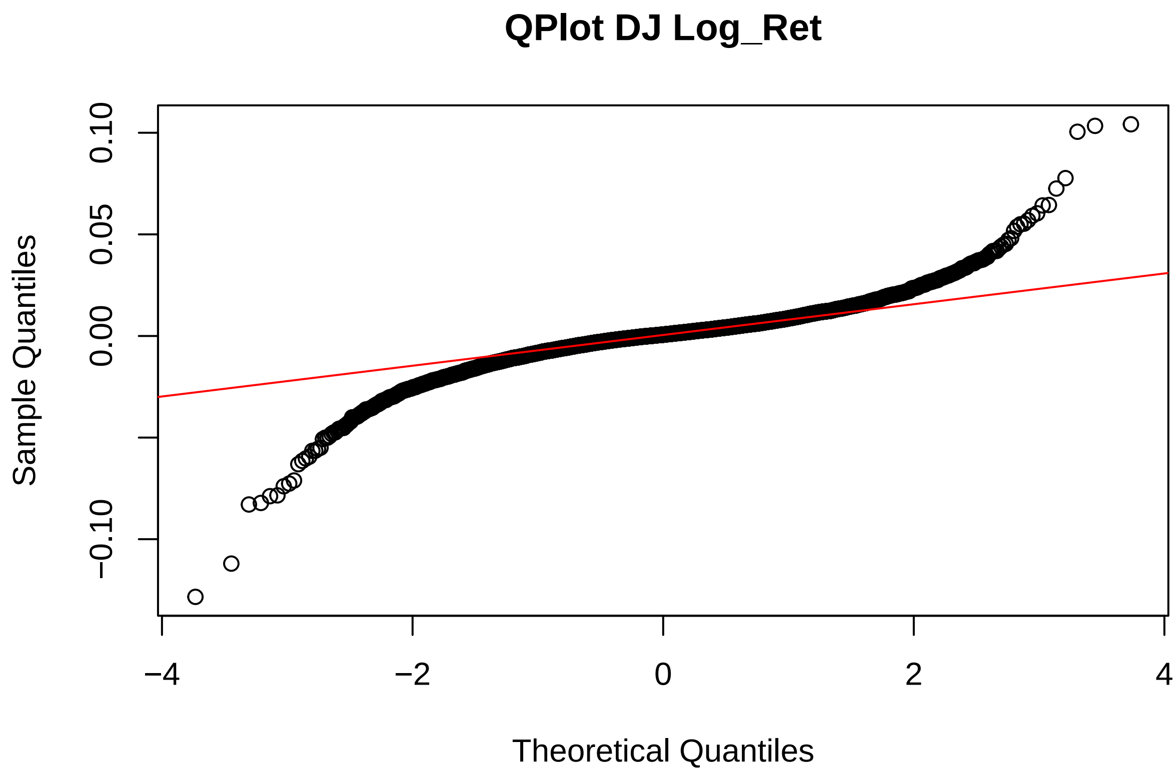
```r
mean(nas_residuals)
```

```
## [1] 0.0002305703
```

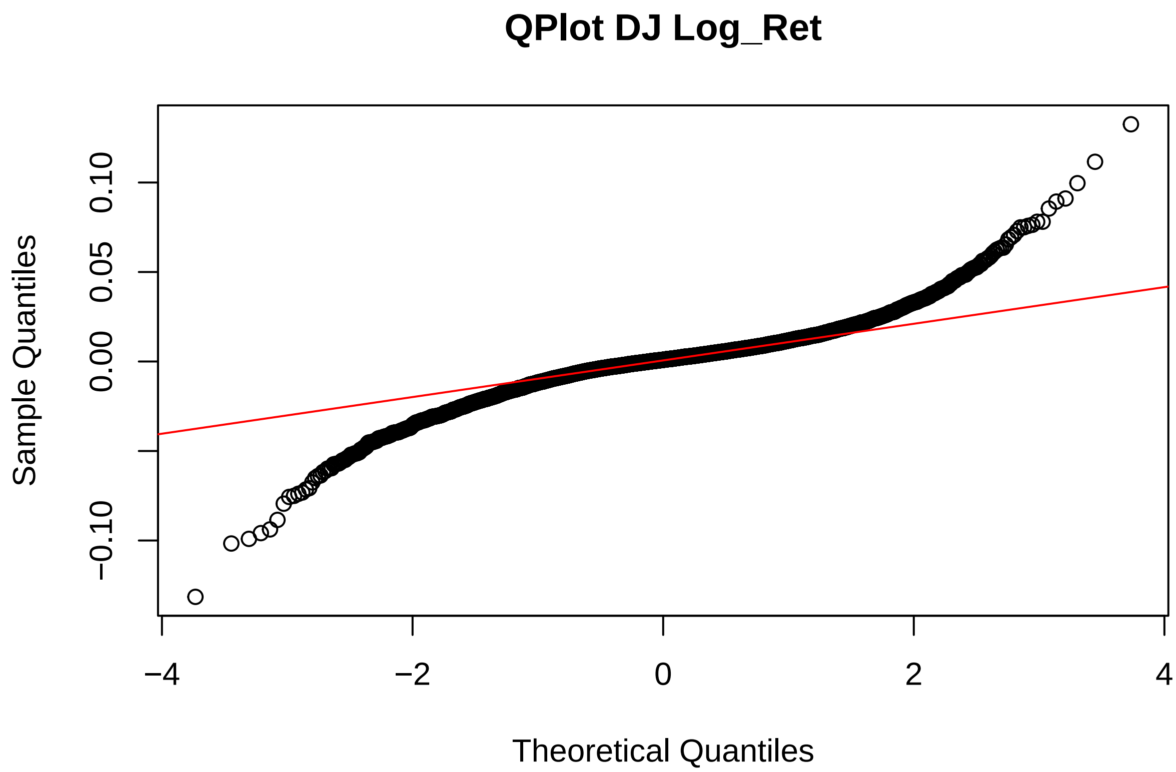
```r
qqnorm(sp_residuals,main="QPlot SP Log_Ret")
qqline(sp_residuals,col='red')
```




```
qqnorm(dow_residuals,main="QPlot DJ Log_Ret")
qqline(dow_residuals,col='red')
```




```
qqnorm(nas_logret,main="QPlot DJ Log_Ret")
qqline(nas_logret,col='red')
```



We can see that our residuals are fairly symmetrical but it has very high kurtosis, meaning we agree with our previous stance that an appropriate distribution would be the Student's t-distribution.

Now let's do a box test to check if our residuals are stationary.

```
lag.length = 50

Box.test(sp_residuals, lag=lag.length, fitdf=1, type="Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  sp_residuals
## X-squared = 163.74, df = 49, p-value = 2.798e-14
```

```
Box.test(dow_residuals, lag=lag.length, fitdf=1, type="Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  dow_residuals
## X-squared = 166.93, df = 49, p-value = 8.882e-15
```

```
Box.test(nas_residuals, lag=lag.length, fitdf=5, type="Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  nas_residuals
## X-squared = 148.3, df = 45, p-value = 5.925e-13
```

No dice! Our data is still non-stationary.

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

As a last-ditch effort, we will adjust our data with change-points to try to make it stationary.

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.

```
source('change_points.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
```

```
dow_changepoint<-wcm.gsa(dow_logret, double.cusum = TRUE)

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

dow_logret_changepoint <- dow_logret - mean_dow
```

```
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
```

Now, let us see if removing the change-points have made our data stationary

```r
lag.length = 50

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

```
##
##  Box-Ljung test
##
## data:  sp_logret_changepoint
## X-squared = 283.45, df = 50, p-value < 2.2e-16
```

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

```
##
##  Box-Ljung test
##
## data:  dow_logret_changepoint
## X-squared = 279.98, df = 50, p-value < 2.2e-16
```

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

```
##
##  Box-Ljung test
##
## data:  nas_logret_changepoint
## X-squared = 187.37, df = 50, p-value < 2.2e-16
```

The test says our data is non-stationary. Let's see if our residuals are stationary.

```r
sp_ar_changepoint <- auto.arima(sp_logret_changepoint , max.order = c(3 , 0 ,3) , trace = T,  max.d = 0
```

```
##
##  Fitting models using approximations to speed things up...
##
##  ARIMA(2,0,2) with non-zero mean : -31528.96
##  ARIMA(0,0,0) with non-zero mean : -31401.71
##  ARIMA(1,0,0) with non-zero mean : -31504.32
##  ARIMA(0,0,1) with non-zero mean : -31500.73
##  ARIMA(0,0,0) with zero mean     : -31403.71
##  ARIMA(1,0,2) with non-zero mean : -31507.33
##  ARIMA(2,0,1) with non-zero mean : Inf
##  ARIMA(3,0,2) with non-zero mean : -31529.72
##  ARIMA(3,0,1) with non-zero mean : Inf
##  ARIMA(4,0,2) with non-zero mean : Inf
##  ARIMA(3,0,3) with non-zero mean : -31527.95
##  ARIMA(2,0,3) with non-zero mean : Inf
##  ARIMA(4,0,1) with non-zero mean : -31521.99
##  ARIMA(4,0,3) with non-zero mean : -31519.38
##  ARIMA(3,0,2) with zero mean     : -31531.7
##  ARIMA(2,0,2) with zero mean     : -31530.96
##  ARIMA(3,0,1) with zero mean     : Inf
##  ARIMA(4,0,2) with zero mean     : Inf
```

```
##  ARIMA(3,0,3) with zero mean     : -31529.94
##  ARIMA(2,0,1) with zero mean     : Inf
##  ARIMA(2,0,3) with zero mean     : -31528.96
##  ARIMA(4,0,1) with zero mean     : -31524
##  ARIMA(4,0,3) with zero mean     : -31521.17
##
##  Now re-fitting the best model(s) without approximations...
##
##  ARIMA(3,0,2) with zero mean     : -31525.71
##
##  Best model: ARIMA(3,0,2) with zero mean
```

```
dow_ar_changepoint <- auto.arima(dow_logret_changepoint , max.order = c(3 , 0 ,3) , trace = T ,   max.d
```

```
##
##  Fitting models using approximations to speed things up...
##
##  ARIMA(2,0,2) with non-zero mean : -31919.96
##  ARIMA(0,0,0) with non-zero mean : -31824.87
##  ARIMA(1,0,0) with non-zero mean : -31923.87
##  ARIMA(0,0,1) with non-zero mean : -31918.61
##  ARIMA(0,0,0) with zero mean     : -31826.87
##  ARIMA(2,0,0) with non-zero mean : -31922.59
##  ARIMA(1,0,1) with non-zero mean : -31922.6
##  ARIMA(2,0,1) with non-zero mean : Inf
##  ARIMA(1,0,0) with zero mean     : -31925.87
##  ARIMA(2,0,0) with zero mean     : -31924.59
##  ARIMA(1,0,1) with zero mean     : -31924.6
##  ARIMA(0,0,1) with zero mean     : -31920.61
##  ARIMA(2,0,1) with zero mean     : Inf
##
##  Now re-fitting the best model(s) without approximations...
##
##  ARIMA(1,0,0) with zero mean     : -31919.65
##
##  Best model: ARIMA(1,0,0) with zero mean
```

```
nas_ar_changepoint <- auto.arima(nas_logret_changepoint , max.order = c(3 , 0 ,3) , trace = T ,  max.d =
```

```
##
##  Fitting models using approximations to speed things up...
##
##  ARIMA(2,0,2) with non-zero mean : -28726.35
##  ARIMA(0,0,0) with non-zero mean : -28691.02
##  ARIMA(1,0,0) with non-zero mean : -28721.96
##  ARIMA(0,0,1) with non-zero mean : -28711.24
##  ARIMA(0,0,0) with zero mean     : -28693.02
##  ARIMA(1,0,2) with non-zero mean : Inf
##  ARIMA(2,0,1) with non-zero mean : -28721.97
##  ARIMA(3,0,2) with non-zero mean : -28761.67
##  ARIMA(3,0,1) with non-zero mean : -28727.49
##  ARIMA(4,0,2) with non-zero mean : -28742.41
##  ARIMA(3,0,3) with non-zero mean : -28759.65
```

```
##  ARIMA(2,0,3) with non-zero mean : -28725.26
##  ARIMA(4,0,1) with non-zero mean : -28732.14
##  ARIMA(4,0,3) with non-zero mean : -28740.4
##  ARIMA(3,0,2) with zero mean     : -28763.67
##  ARIMA(2,0,2) with zero mean     : Inf
##  ARIMA(3,0,1) with zero mean     : -28729.52
##  ARIMA(4,0,2) with zero mean     : -28744.36
##  ARIMA(3,0,3) with zero mean     : -28761.68
##  ARIMA(2,0,1) with zero mean     : -28723.97
##  ARIMA(2,0,3) with zero mean     : -28727.31
##  ARIMA(4,0,1) with zero mean     : -28734.28
##  ARIMA(4,0,3) with zero mean     : -28743.2
##
##  Now re-fitting the best model(s) without approximations...
##
##  ARIMA(3,0,2) with zero mean     : -28723.52
##
##  Best model: ARIMA(3,0,2) with zero mean
```

```r
sp_residuals_changepoint <- sp_ar_changepoint$residuals
dow_residuals_changepoint <- dow_ar_changepoint$residuals
nas_residuals_changepoint <- nas_ar_changepoint$residuals

lag.length = 50

Box.test(sp_residuals_changepoint, lag=lag.length, fitdf=6, type="Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  sp_residuals_changepoint
## X-squared = 155.77, df = 44, p-value = 2.098e-14
```

```r
Box.test(dow_residuals_changepoint, lag=lag.length, fitdf=1, type="Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  dow_residuals_changepoint
## X-squared = 174.98, df = 49, p-value = 4.441e-16
```

```r
Box.test(nas_residuals_changepoint, lag=lag.length, fitdf=5, type="Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  nas_residuals_changepoint
## X-squared = 148.45, df = 45, p-value = 5.618e-13
```

Removing change points looks to have not changed the p-values all that much. Our data is still autocorrelated. Therefore, we will keep the log-returns with the change points intact as that data will be more reliable to train an LSTM with and will be easier to infer on.

Outputting Files

```
setwd('..')
write.csv(sp_logret, 'Data/Processed/sp_logret.csv', row.names=T)
write.csv(dow_logret, 'Data/Processed/dow_logret.csv', row.names=T)
write.csv(nas_logret, 'Data/Processed/nas_logret.csv', row.names=T)

write.csv(sp_residuals, 'Data/Processed/sp_residuals.csv', row.names=T)
write.csv(dow_residuals, 'Data/Processed/dow_residuals.csv', row.names=T)
write.csv(nas_residuals, 'Data/Processed/nas_residuals.csv', row.names=T)
```