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.

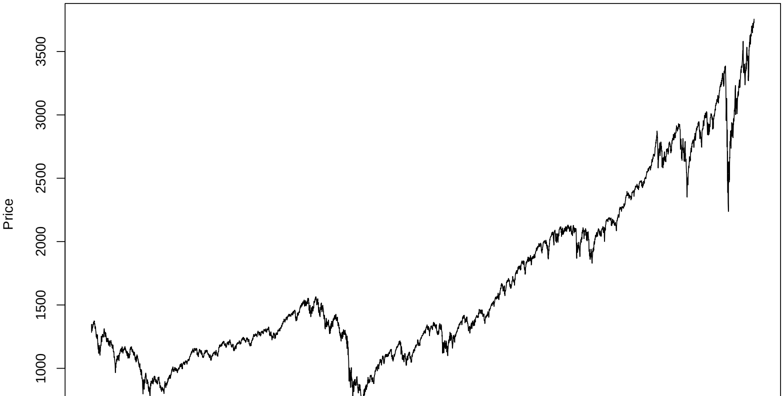
```
setwd("..")
sp<-read.csv("Data/sp500.csv")
dow<-read.csv("Data/dowjones.csv")
nas<-read.csv("Data/nasdaq.csv")
```

Now we will rename some columns and fixing Date and number

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:

```
#Plotting Price against Date for each index
plot(sp, type='l')
```



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.
## -0.1276521 -0.0045195  0.0006476  0.0002135  0.0057220
```

```
summary(dow_logret)
```

```
##        Min.    1st Qu.     Median       Mean    3rd Qu.
## -0.1384181 -0.0046074  0.0005063  0.0001877  0.0055894
```

```
summary(nas_logret)
```

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

Now we will calculate the skewness of our data:

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 = 50

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

```
##
##  Box-Ljung test
##
## data:  sp_logret
## X-squared = 245.36, df = 50, p-value < 2.2e-16
```

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

```
##
##  Box-Ljung test
##
```

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) , t
```

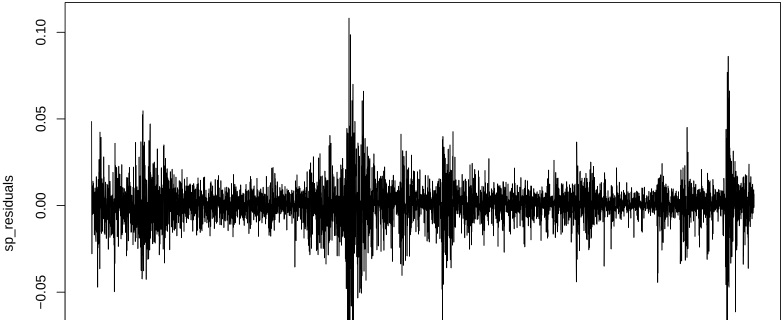
```
##
##  Fitting models using approximations to speed things up.
##
##  ARIMA(2,0,2) with non-zero mean : -29919.24
##  ARIMA(0,0,0) with non-zero mean : -29835.37
##  ARIMA(1,0,0) with non-zero mean : -29919.68
##  ARIMA(0,0,1) with non-zero mean : -29905.35
##  ARIMA(0,0,0) with zero mean     : -29835.89
##  ARIMA(2,0,0) with non-zero mean : -29917.35
##  ARIMA(1,0,1) with non-zero mean : -29917.84
##  ARIMA(2,0,1) with non-zero mean : Inf
##  ARIMA(1,0,0) with zero mean     : -29919.96
##  ARIMA(2,0,0) with zero mean     : -29917.58
```

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')
```



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])
```

Outputting Files

```
setwd('..')
write.csv(sp_logret, 'Data/Processed/sp_logret.csv', row.na
write.csv(dow_logret, 'Data/Processed/dow_logret.csv', row.
write.csv(nas_logret, 'Data/Processed/nas_logret.csv', row.

write.csv(sp_residuals, 'Data/Processed/sp_residuals.csv',
write.csv(dow_residuals, 'Data/Processed/dow_residuals.csv
write.csv(nas_residuals, 'Data/Processed/nas_residuals.csv
```