**Q1. Obtain the Time Series Analysis of Log-Returns for S&P500**

This report utilizes daily data of the S&P 500 (GSPC) index, spanning from January 1, 2009, to October 31, 2023. It also includes time series transformation to log returns, autocorrelation analysis, stationarity testing, normality tests, ARIMA modelling, and residual analysis.

# install.packages("quantmod")

library(quantmod)

# Obj 1: Load stock prices by symbol

getSymbols("^GSPC", from = '2009-01-01',

to = "2023-10-31")

**Q2. Transforming Series into Log-Returns**

The data was transformed into log returns. In this report, log return was used as the target of research.

Mathematical equation for log-returns can be expressed as:

where; denotes the natural logarithm, is the value of the time series at time , and is the value of the time series at the previous time step. The 1-Liner code for the output is given below;

y = length(GSPC$GSPC.Adjusted[-1])

Rett = GSPC$GSPC.Adjusted[-1]/GSPC$GSPC.Adjusted[-y]-1

ret = diff(log(GSPC$GSPC.Adjusted))

#The 1-liner code responsible for the plot output shown below:

plot(100\*Rett, type = "l", xlab = "Time", ylab = "log\_return", col = 2)

lines(100\*ret, col = 1, cex = 0.1)

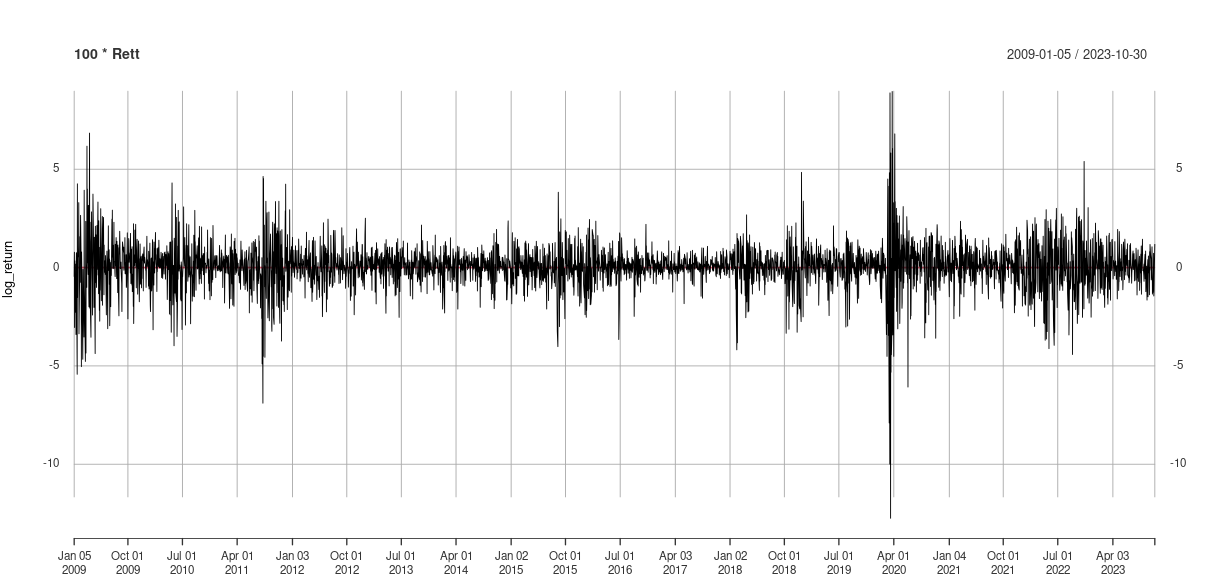


Figure 1 Log\_Returns

I plotted the log return against time and the mean of the log return as seen above is almost zero. The volatility is larger at the beginning of 2020.

The mean of log returns is almost zero(0) and the min and max are approximately around zero. From the output, the kurtosis is 14.15 which is larger than a normal distribution. Kurtosis greater than 3 indicates thicker, heavier tails (leptokurtic), so the S&P500(GSPC) return has a heavier tail than normal. The skewness is -0.6.

**Q3. Autocorrelation Analysis**

The autocorrelation function (ACF) and partial autocorrelation function (PACF) were scrutinized to identify any autocorrelation in the log return.

#The 1-liner code responsible for the plot output shown below:

Acf(ret)

Pacf(ret)



Figure 2. PACF Figure 3. ACF

The PACF chart shows a significant lag at lag 1, with a p-value of less than 0.05. The ACF chart also shows a significant lag at lag 1, with a p-value of less than 0.05. This confirms the finding from the PACF chart that the current value of the series is correlated with the value of the series at lag 1.

**Q4. Ljung-Box Test**

The Ljung-Box test is performed to assess the presence of autocorrelation in the log-returns series.

#The code for the output:

Ljung\_box = Box.test(ret,lag = 10, type = "Ljung-Box")

p\_value = Ljung\_box$p.value

if (p\_value < 0.05){

print("Reject H0. The Data Exhibits Autocorrelation")

} else{

print("Fail to Reject H0. There is no significant evidence of any Autocorrelation.")

}

The p-value is less than 0.05, Therefore, reject the null hypothesis. There is significant evidence of autocorrelation in the data.

**Q5. Stationarity Testing**

The KPSS test, also known as the Kwiatkowski-Phillips-Schmidt-Shin test, was used for the stationary test. The null hypothesis (H0) of the KPSS test is that the time series is stationary around a deterministic trend. The alternative hypothesis (H1) is that the time series is non-stationary and has a unit root.

kpss <- kpss.test(ret)

p\_value <- kpss$p.value

if (p\_value > 0.05){

print("Fail to Reject H0. The Data is Stationary")

} else{

print("Reject H0. The Data is not Stationary")

}

**Q6. Normality Test**

The Jarque-Bera test rejected the null hypothesis that the returns are normally distributed. The Jarque-Bera test is used to evaluate whether a sample of data comes from a normal distribution.

ja <- jarque.bera.test(ret)

jp\_value <- ja$p.value

if (jp\_value > 0.05){

print("Fail to reject H0. Returns Appears to be Normally Distributed")

} else{

print("Reject H0. Returns do not appear to be Normally Distributed")

}

In this case, the p-value is less than 0.05, which means that we can reject the null hypothesis at the 5% significance level.

**Q7. ARIMA Modeling**

ARIMA model is fitted to the log returns series to capture the underlying time series structure.

fitted\_model = auto.arima(ret)

ar\_coe = model$coef[1:2]

ma\_coe = model$coef[(2+1):(2+2)]

print(ar\_coe)

print(ma\_coe)

**Q8. Coefficient of the ARIMA Model**

The coefficients for the ARIMA(2,0,2) model are as follows:

Autoregressive (AR) Coefficients:

AR1 : -1.6900228 AR2 : -0.8508925

Moving Average (MA) Coefficients:

MA1 : 1.5953570 MA2 : 0.7573147

These coefficients represent the strength and direction of the relationships between the current observation and its past values and error terms in the ARIMA model.

The respective equation given these coefficients are;

eqn = paste0("y[t] = ",paste(ar\_coe,collapse = "+"),"\*y[t-",1:2,"] + ",

+ paste(ma\_coe,collapse = "+"),"e[t-",1:2,"]")

print(eqn)

[1] "y[t] = -1.6900228046971+-0.850892517558664\*y[t-1] + 1.59535700650594+0.7573146576004e[t-1]"

[2] "y[t] = -1.6900228046971+-0.850892517558664\*y[t-2] + 1.59535700650594+0.7573146576004e[t-2]"

**Q9. Residual Analysis**

The ARIMA model's residuals are examined to ensure they meet the model's assumptions. The results of the residual analysis are reported.

Using the results from checkresiduals(fitted\_model) function;

Ljung-Box test

data: Residuals from ARIMA(2,0,2) with non-zero mean

Q\* = 12.233, df = 6, p-value = 0.05697

Model df: 4. Total lags used: 10



Figure 4. Residuals

Based on the results, the residuals from the ARIMA(2,0,2) model with non-zero mean fully meet the requirements for residuals from an ARIMA fit.

The residuals have a mean of zero, as indicated by the zero mean in the condition for stationarity. This is a desirable property for the residuals in a time series model, suggesting that, on average, the model is unbiased in its predictions.

The residuals have finite variance. i.e, they are not "exploding" or becoming extremely large.