

## 2. Theory

It is assumed that the reader of this paper already has basic knowledge of the mathematical principles of time series analysis. Therefore, this section will only briefly describe the mathematical models and processes.

### 2.1. Time-Series

Almost anything with a data point to a given timestamp can be named a time-series. The monthly gross domestic product, the weekly US gasoline production, or daily stock price movements. In this paper lies the focus of the analysis of financial time series. Due to trades often only take place during the week, there are gaps in the time series on the weekends, an exception would be the trading of cryptocurrencies like Bitcoin which are also tradeable at the weekends.

A series of data points with more or less equidistant time points  $t$  with the sample length of  $T$ , is called a time-series  $x_t, t = 1, \dots, T$  [eco1]. The analysis of a time-series  $x_t$  involves creating a reasonable model that can be utilized to perform forecast predictions.

#### 2.1.1. Stationarity

In order to fit a suitable model with a given time series  $x_t$ , the assumptions of stationarity must be met. In this practical application, only the following weak-stationarity properties are required.

$$E[x_t] = \mu \quad (1)$$

$$Var(x_t) = \sigma_x^2 \quad (2)$$

$$Cov(x_t, x_{t-k}) = R(k) \quad (3)$$

Many financial time-series are subject to shift, trends or changing volatility. In figure ?? are the stock prices of Alphabet Inc Class A (Google) visualized. This time-series shows a clear upwards drift and towards the end the volatility increases.

```
## Warning: package 'quantmod' was built under R version 4.0.3
```

```
## Loading required package: xts
```

```
## Warning: package 'xts' was built under R version 4.0.3
```

```
## Loading required package: zoo
```

```
## Warning: package 'zoo' was built under R version 4.0.3
```

```
##
```

```
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
```

```
##
```

```
## as.Date, as.Date.numeric
```

```
## Loading required package: TTR

## Warning: package 'TTR' was built under R version 4.0.3

## Registered S3 method overwritten by 'quantmod':
##   method      from
##   as.zoo.data.frame zoo

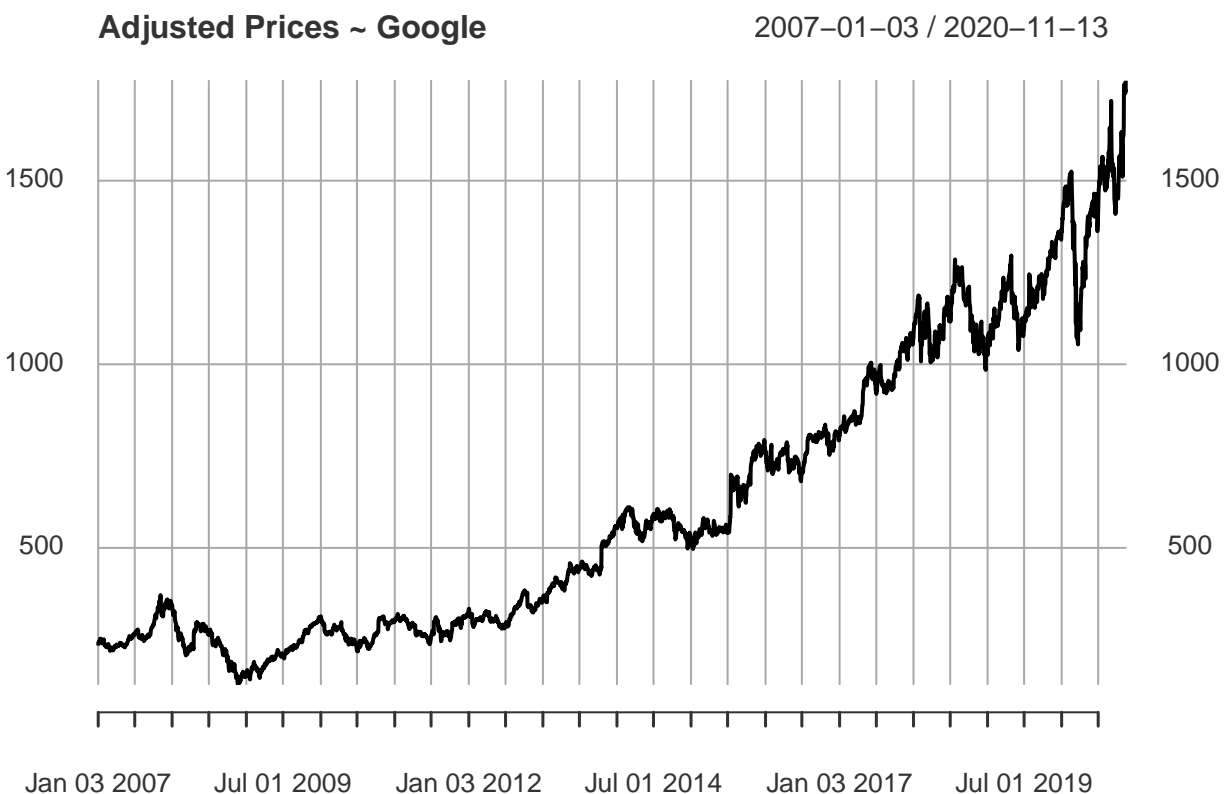
## Version 0.4-0 included new data defaults. See ?getSymbols.

## Warning: package 'PerformanceAnalytics' was built under R version 4.0.3

##
## Attaching package: 'PerformanceAnalytics'

## The following object is masked from 'package:graphics':
##
##   legend

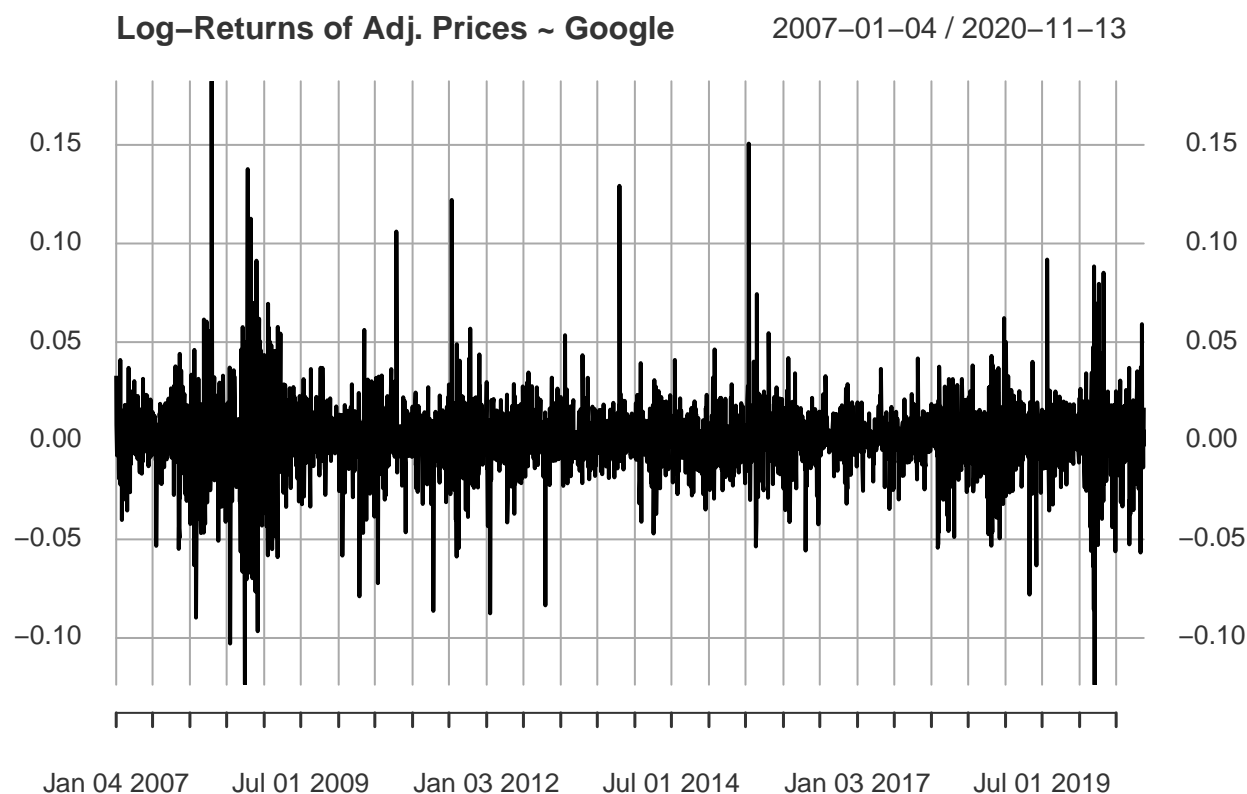
## 'getSymbols' currently uses auto.assign=TRUE by default, but will
## use auto.assign=FALSE in 0.5-0. You will still be able to use
## 'loadSymbols' to automatically load data. getOption("getSymbols.env")
## and getOption("getSymbols.auto.assign") will still be checked for
## alternate defaults.
##
## This message is shown once per session and may be disabled by setting
## options("getSymbols.warning4.0"=FALSE). See ?getSymbols for details.
```



To improve the violated properties the first difference can be applied and additionally a logarithmic transformation can be performed [slide\_eco3\_1]. The log-returns transformation can only be performed to strict positive data.

$$\text{LogReturn} = \log(x_t) - \log(x_{t-1})$$

The result is the so-called log-returns.



Applying the transformation to the data causes the drift to disappear, but the series still contains stronger and weaker volatile phases. This effect often occurs in non-stationary financial data and is called volatility cluster. This special property is used for the modelling of forecast models, which will be discussed in chapter 2.2.

### 2.1.2. Autocorrelation

The autocorrelation function (ACF) reveals how the correlation between any two data points of the time series changes as their separation changes [acf]. More precisely, acf measures the dependence between  $x_t$  and  $x_{t \pm k}$  at lag  $k$ . The partial autocorrelation (PACF) measures the dependency between  $x_t$  and  $x_{t-k}$  at lag  $k$  [eco1]. For stationary time series, ACF can be used to identify the model order of a MA-process, PACF for AR-processes.

In the following figure ?? are acf's of the non-stationary adjusted Google stock and their log-returns visualized. Both graphics show the typical pattern of a non-stationary time series. The left plot shows the dependence structure of the time series. This means that it takes a long time until the series changes. Often a large value is followed by another large value, which indicates a strong trend. This property of the series can be seen in figure ?. In the right plot the acf of the log-returns are visualized. Remember figure ??,

there is no trend visible and therefore no long dependency structures are visible, only slight dependencies can be seen.

In the following section 2.2. the characteristics of the autocorrelation function can be used for the verification of ARIMA and ARCH-processes.

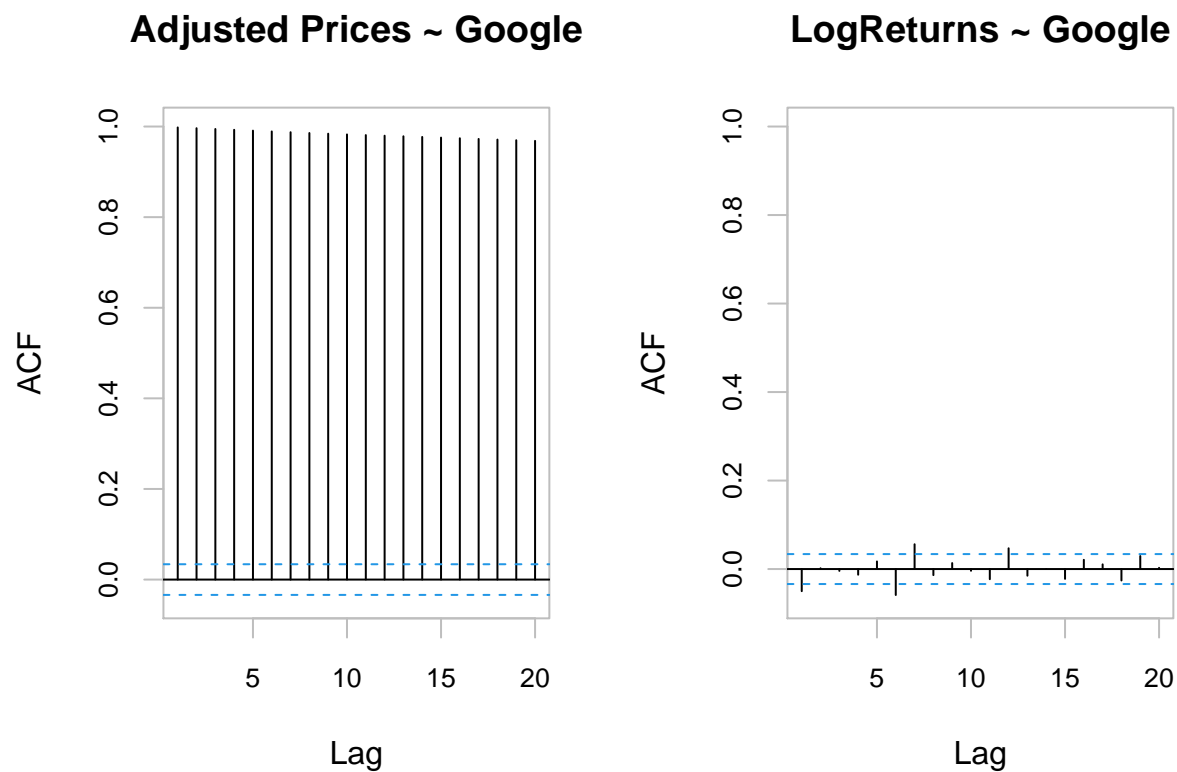


Figure 1: Acf and Pacf of the LogReturns of Google

## 2.2. Models

The following processes are used to determine certain properties and characteristics of a time series so that they are transformed into a model. The goal is to fit the time series as well as possible in order to create reliable forecasts.

### 2.2.1. ARIMA

An ARIMA( $p, d, q$ ) process is defined as follows.

$$x_t = c + a_1 x_{t-1} + \dots + a_p x_{t-p} + \epsilon_t + b_1 \epsilon_{t-1} + \dots + b_q \epsilon_{t-q} \quad (4)$$

- $p$  and  $q$  are the AR- and MA-model orders
- $a$  and  $b$  are the AR- and MA-model parameters
- $d$  is the differential parameter
- $\epsilon_t$  is a white noise sequence
- $x_t$  is the given data  $x_1, \dots, x_T$

The mean of an ARIMA-process can be computed as:

$$\mu = \frac{c}{1 - a_1 - \dots - a_p}$$

ARIMA processes can be divided into 4 different models. Choosing a model that best represents the time series is a difficult task. The goal is to find the best possible model with as few parameters as possible.

The previously introduced ACF and PACF can help to choose simple models. Provided that the time series is stationary, the model orders can be determined directly. For an AR( $p$ )-process (ARIMA( $p, 0, 0$ )), the ACF plot will gradually decrease and simultaneously the PACF should have a sharp drop after  $p$  significant lags. For an MA( $q$ )-process (ARIMA( $0, 0, q$ )) the opposite is true, the ACF should show a sharp drop after a certain  $q$  number of lags while PACF should show a gradual decreasing trend. If both ACF and PACF show a gradual decreasing pattern, then the ARMA( $p, 0, q$ )-process should be considered for modeling [arima].

### 2.2.2. ARCH & GARCH

The volatility clustering mentioned in section 2.1.1 can be handled with an auto-regressive conditional heteroscedastic process.

$$\begin{aligned} \epsilon_t &= \log(x_t) - \log(x_{t-1}) \\ \epsilon_t &= \sigma_t u_t \\ \sigma_t^2 &= c\sigma^2 + \sum_{k=1}^m \beta_k \epsilon_{t-k}^2 \end{aligned} \quad (5)$$

with:

- $x_t$  is the original data (often non-stationary)
- $\epsilon_t$  is the stationary log-return
- $u_t$  is independent and identically distributed (iid) and standardized random variable
- $\sigma^2$  is the unconditional variance of the process  $\epsilon_t$ .
- $\sigma_t^2$  is the conditional variance of the process  $\epsilon_t$ .

The ARCH-process can be generalized by adding the lagged conditional variances to the equation 5.

$$\begin{aligned}
\epsilon_t &= \log(x_t) - \log(x_{t-1}) \\
\epsilon_t &= \sigma_t u_t \\
\sigma_t^2 &= c\sigma^2 + \sum_{j=1}^n \alpha_j \sigma_{t-j}^2 + \sum_{k=1}^m \beta_k \epsilon_{t-k}^2
\end{aligned} \tag{6}$$

### 2.2.3. ARIMA-GARCH

Another process is the combination of ARIMA and GARCH processes.

$$y_t = \mu + a_1 y_{t-1} + \dots + a_p y_{t-p} + \epsilon_t + b_1 \epsilon_{t-1} + \dots + b_q \epsilon_{t-q} \tag{7}$$

$$\begin{aligned}
\epsilon_t &= \sigma_t u_t \\
\sigma_t^2 &= c\sigma^2 + \sum_{j=1}^n \alpha_j \sigma_{t-j}^2 + \sum_{k=1}^m \beta_k \epsilon_{t-k}^2
\end{aligned} \tag{8}$$

Is called the mean-equation 7

Is called the variance-equation 8

[@eco2]

## 2.3. Moving Average Filters

moving average filters are basically used to identify trends and smooth out price fluctuations. As a commonly used tool moving average filters are very simple in its usage, historical data was summarized and divided by the length of the filter. The actual challenge in using Moving average filters is to figure out which length of the filter brings the most useful information.

### 2.3.1. Equally-weighted Moving Average

SMA stands for Simple Moving Average which, depending on the length of the filter( $L$ )  $L$  observations since the last noted observation will be considered. they're getting summarized and divided by the filterlength equals the EqMA. For every timestep a new observation is considered and the last one eliminated.

EqMA

$$y_t = \frac{1}{L} \sum_{k=0}^{L-1} x_{t-k} \quad (9)$$

- $L$  = filterlength

### 2.3.2. Exponentially-weighted Moving Average

Since not all observations are having the same influence of future value we can apply a weight to past observations. One method will be exponentially weighted Moving average. So we chose an optimal parameter to give past observations weights decreasing by alpha

A skillfull trader chose an optimal to increase the performance of the measurement. Weights could also be given individually by adding a weight vector to the filter.

EMA

$$y_t = \frac{1}{\sum_{k=0}^m \alpha^k} \sum_{k=0}^m \alpha^k x_{t-k} \quad (10)$$

- $m$  = filterlength
- $\alpha$  = Parameter to weigh the observations

### 2.3.3. Moving Average Crossings

$$y_t = \frac{1}{L_1} \sum_{k=0}^{L_1-1} x_{t-k} - \frac{1}{L_2} \sum_{k=0}^{L_2-1} x_{t-k} \quad (11)$$

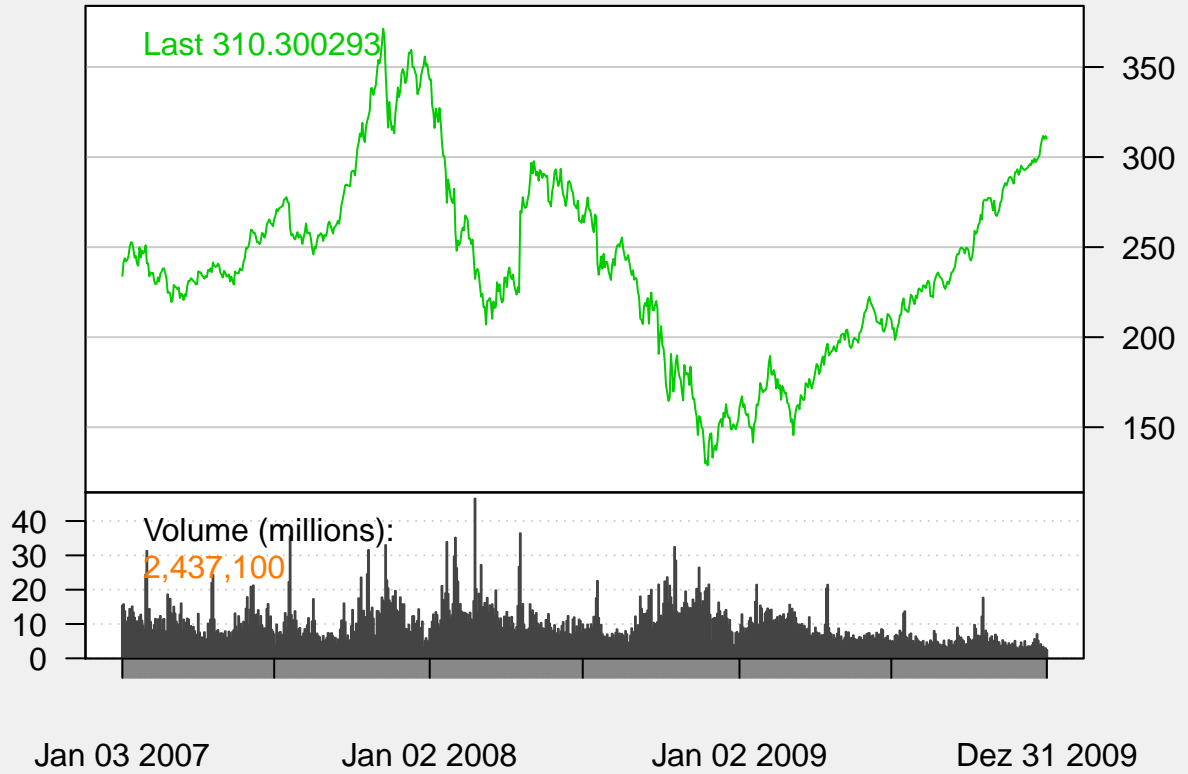
- $L_1$  = filterlength 1
- $L_2$  = filterlength 2
- $0 < \alpha < 1$  = Parameter to weigh the observations

Moving average crossings are basically just different MA's with different lengths applied to a time-series. The points the filters then cross will be used as a trading signal to go long, short or hold.

An easy example of Ma average crossings with 2 mas of different length is visualized in # ??.

GOOGL

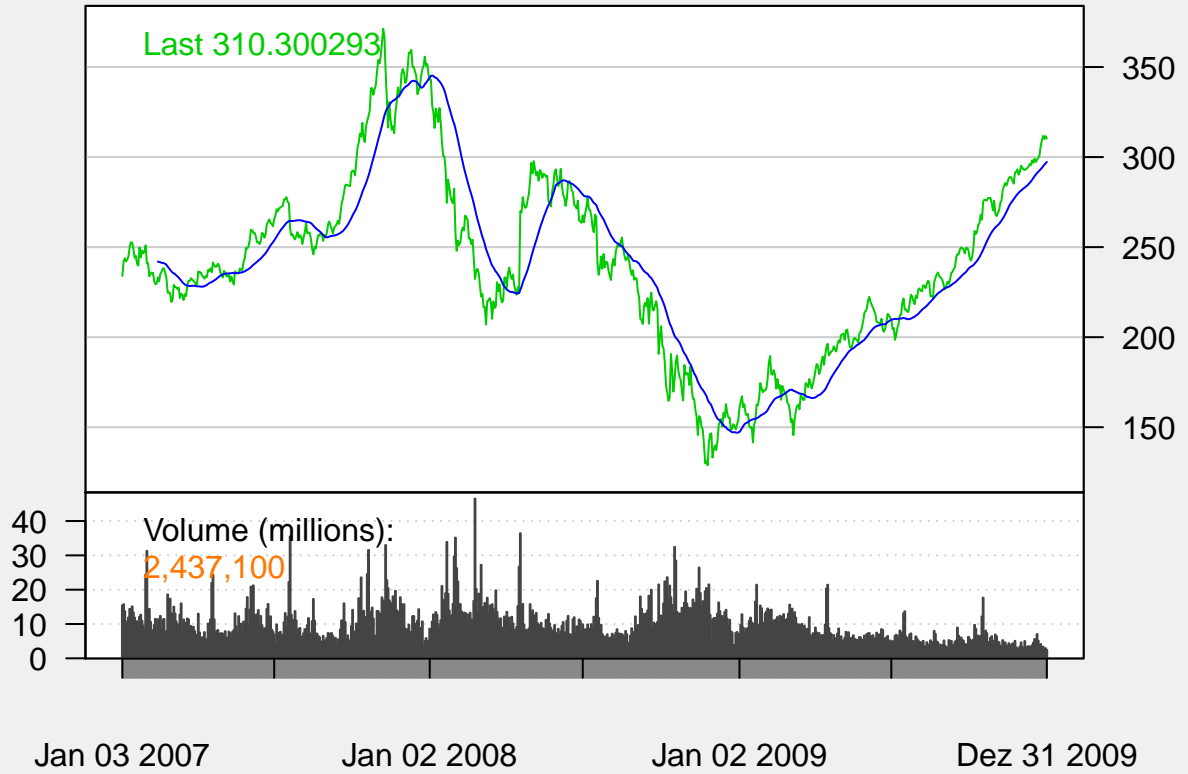
[2007-01-03/2009-12-31]





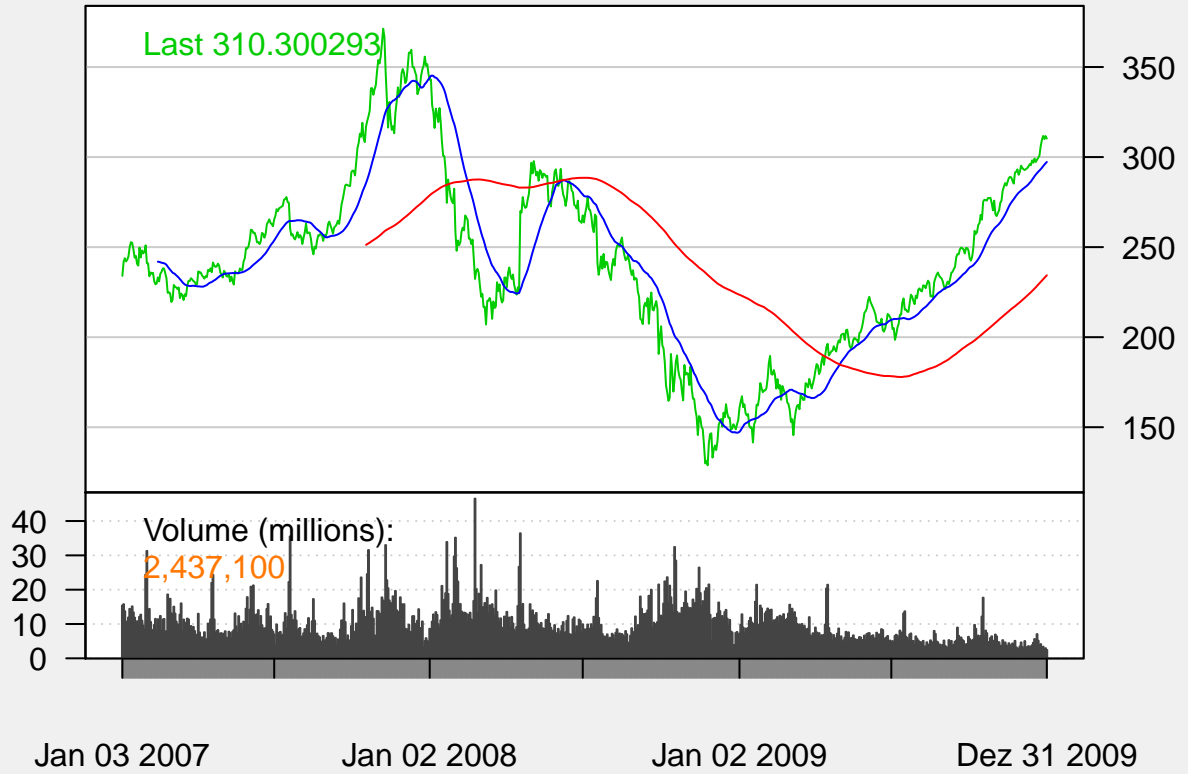
GOOGL

[2007-01-03/2009-12-31]



GOOGL

[2007-01-03/2009-12-31]



## 2.4. Real Strength Index

## 2.5. Sharpe Ratio

Sharpe ratio is a very powerful and widely used ratio to measure performance. It describes return per risk

$$sharperatio_p = \frac{R_p - R_f}{\sigma} \quad (12)$$

- $R_p$  = Return of Portfolio
- $R_f$  = Risk free Rate, mostly treasury bonds
- $\sigma_p$  = standard deviation of portfolios excess return (risk)

## 2.6. Carry

## 2.7. Value

## 2.8 Bollinger bands

Try other github branch