Time Series Analysis Using R

Dataset Description: We used a dataset from Yahoo Finance's common stock price data for this study. Its popular and much important dataset for time series analysis of stock price prediction. Finance is a media property of the Yahoo! Network, which has been owned by Verizon Media since 2017. It offers stock quotes, press releases, financial reports, and original material, as well as financial news, data, and commentary. It also has several online resources for handling personal finances. For several exchanges, Yahoo Finance offers real-time streaming quotes. During trading hours on an exchange, and in some cases during pre-market and post-market hours, real-time data is available. When a red or green backdrop flashes behind the stock price, you'll know it's real-time.

Install important packages:

```
library(quantmod)
library(fpp)
library(backtest)
```

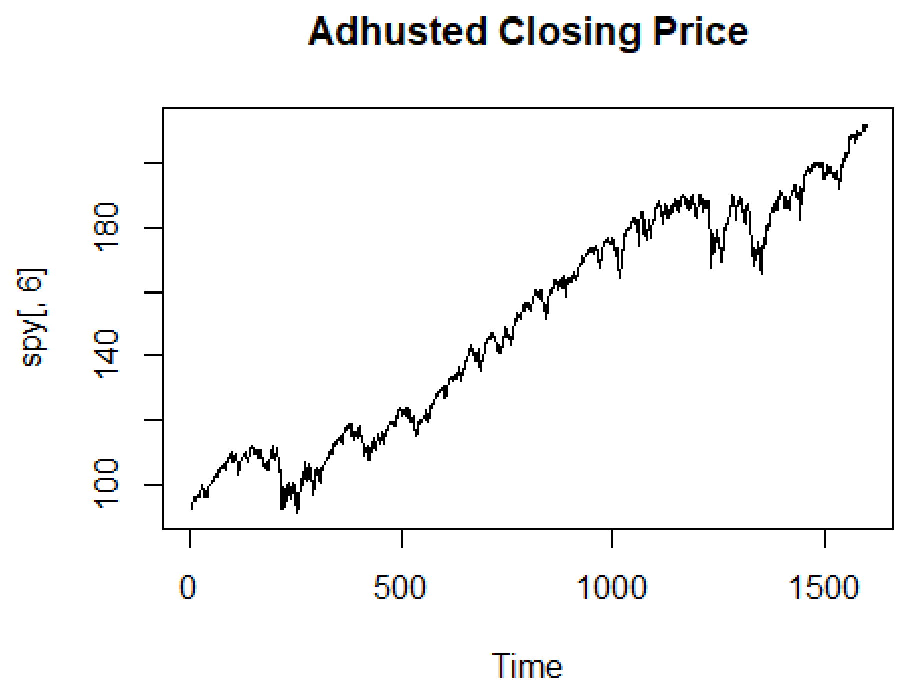
Read dataset from yahoo finance:

```
spy <- getSymbols("SPY", from="2010-10-01", src="yahoo", auto.assign=F)
head(spy)
```

```
##            SPY.Open SPY.High SPY.Low SPY.Close SPY.Volume SPY.Adjusted
## 2010-10-01   114.99   115.12  113.93    114.61  174638700     93.12812
## 2010-10-04   114.37   114.85  113.18    113.75  166153200     92.42933
## 2010-10-05   114.80   116.32  114.67    116.04  229634100     94.29011
## 2010-10-06   116.02   116.33  115.56    116.03  148626600     94.28198
## 2010-10-07   116.50   116.53  115.19    115.89  164860000     94.16824
## 2010-10-08   116.05   116.86  115.61    116.54  177760100     94.69640
```

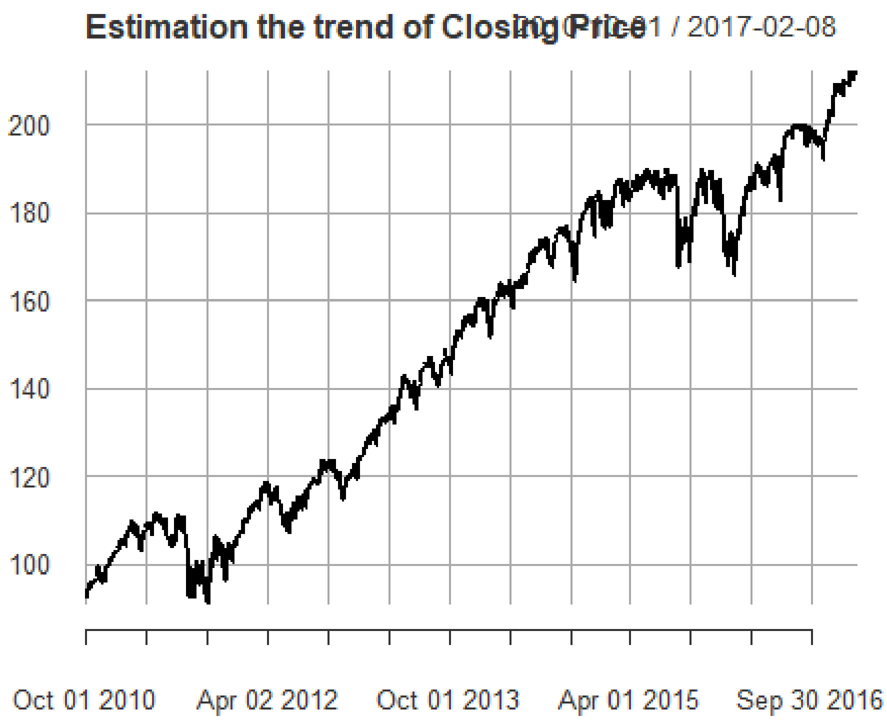
Time series analysis plot: Once read a time series data into R, the next move is usually to make a plot of the time series data, We can make time series plot by using plot.ts() function in R.

```
spy <- spy[1:1600,]
adj <- spy[,6]
plot.ts(spy[,6], main = "Adhusted Closing Price")
```



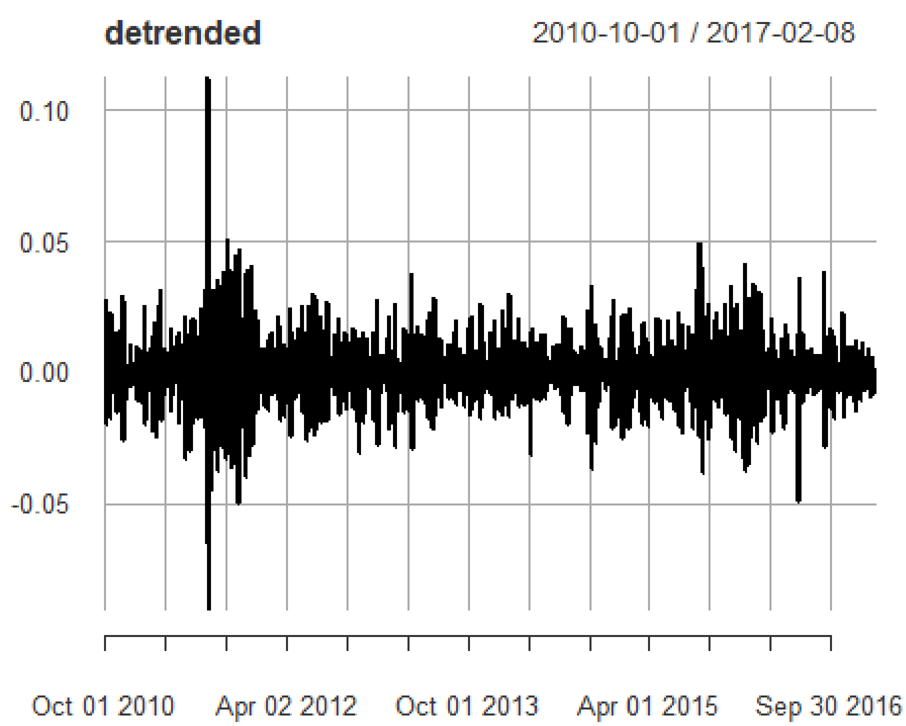
We can assume no kind of seasonality on this data. To confirm the seasonality, we can use various kind of approaches.

```
plot(spy[,6], main = "Estimation the trend of Closing Price")
```



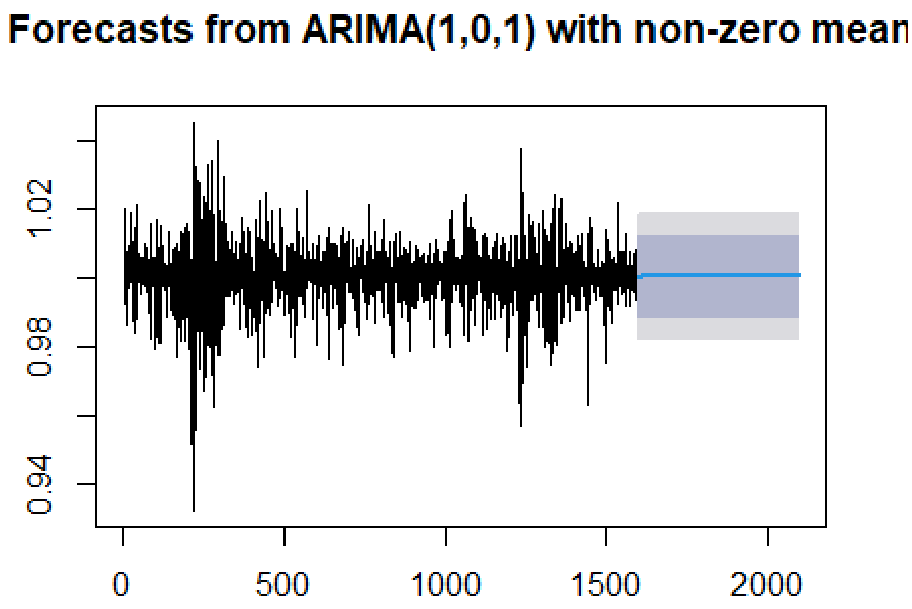
Plot the detrended time series:

```
logr <- periodReturn(adj,period = "daily", type = "log", leading = TRUE)
logr <- 1+logr
head(logr)

##            daily.returns
## 2010-10-01     1.0000000
## 2010-10-04     0.9924682
## 2010-10-05     1.0199319
## 2010-10-06     0.9999138
## 2010-10-07     0.9987928
## 2010-10-08     1.0055930

detrended<-diff(logr)
plot(detrended)
```



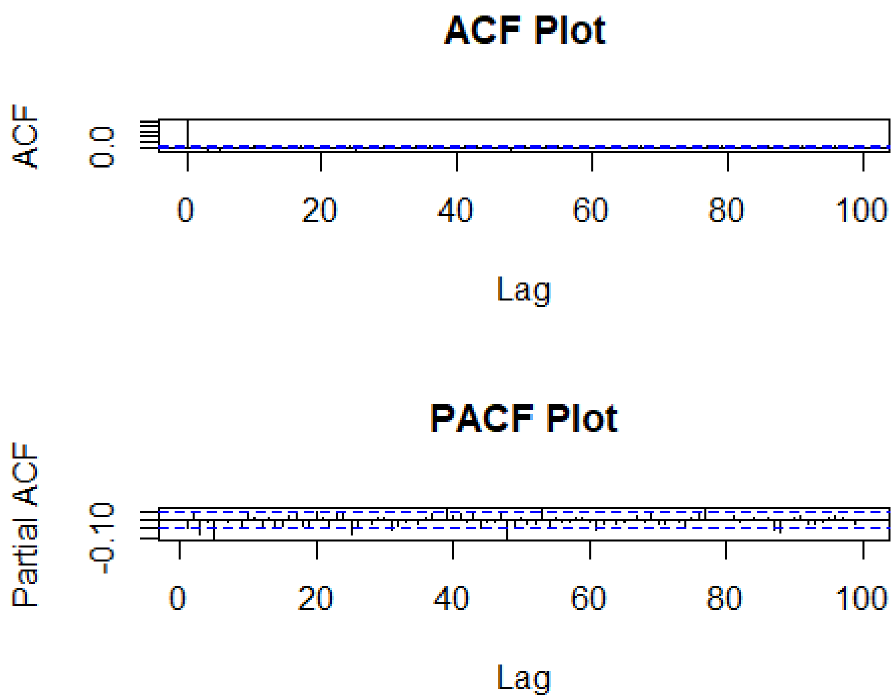
Check seasonality of the data:
This can then for example be used in the forecast packages auto.arima() function.

```
m1 <- auto.arima(logr, seasonal = FALSE)
m1

## Series: logr
## ARIMA(1,0,1) with non-zero mean
##
## Coefficients:
##           ar1     ma1    mean
##       -0.8617  0.8143  1.0005
## s.e.   0.0608  0.0688  0.0002
##
## sigma^2 estimated as 8.611e-05:  log likelihood=5219.09
## AIC=-10430.17   AICc=-10430.15   BIC=-10408.66

summary(m1)

## Series: logr
## ARIMA(1,0,1) with non-zero mean
##
## Coefficients:
##           ar1     ma1    mean
##       -0.8617  0.8143  1.0005
## s.e.   0.0608  0.0688  0.0002
##
## sigma^2 estimated as 8.611e-05:  log likelihood=5219.09
## AIC=-10430.17   AICc=-10430.15   BIC=-10408.66
##
## Training set error measures:
##                        ME        RMSE         MAE          MPE      MAPE
## Training set 1.178634e-05 0.009270893 0.006482475 -0.007455705 0.6488903
##                   MASE       ACF1
## Training set 0.6745224 0.01153074

plot(forecast(m1,h=500))
```


Forecasts from ARIMA(1,0,1) with non-zero mean
1.02
0.98
0.94
0
500
1000
1500
2000

Calculate the empirical autocorrelation function (acf) of the detrended series with R:

```
par(mfrow = c(2,1))
acf.logr = acf(logr, main='ACF Plot', lag.max=100)
pacf.logr = pacf(logr, main='PACF Plot', lag.max=100)
```




```
print(adf.test(logr))

## Warning in adf.test(logr): p-value smaller than printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  logr
## Dickey-Fuller = -12.458, Lag order = 11, p-value = 0.01
## alternative hypothesis: stationary
```

interpretation of the acf plot: ACF and PACF plots to decide whether or not to go with ARIMA model. We know that A correlogram (also called Auto Correlation Function ACF. Plot or Autocorrelation plot) is a visual way to show serial correlation in data that changes over time (i.e. time series data). and partial correlation is a conditional correlation. It is the correlation between two variables under the assumption that we know and take into account the values of some other set of variables. The ACF and PACF shown in the figure above respectively.

The End