```r
#################### Financial Econometrics - Spring 2021 #####################
#################### Karoll Gómez Portilla ####################
#################### Universidad Nacional de Colombia #####################
#################### Lab 1: General introduction (Based on DataCamp) #####################

### Starbucks stock monthly price series from 1998 to 2009

starbx=read.table(file=file.choose(), header=TRUE, sep=",", dec=".")
starbx = read.delim("/Users/karollgomez/Dropbox/Karoll/Cursos/EconometriaFinanciera/NotasClase/Intro/sbuxPrices.csv",header=TRUE,sep = ",", dec = ".")

# Have a look at the structure of data
head(starbx)
tail(starbx)
class(starbx)

# Assign to the variable all the adjusted closing prices while preserving the dimension information
closing =starbx[, "Adj.Close"]

plot(starbx$Adj.Close)
plot(closing, type="l", col="blue", lwd=2, ylab="Adjusted close", main="Monthly closing price of SBUX")

############# Returns

# Computing simple returns Rt=(Pt-Pt-1)/Pt-1, and denote n the number of time periods
sbux = as.matrix(closing)
n = nrow(sbux)
sbux.ret = (sbux[2:n, 1] - sbux[1:(n - 1), 1]) / sbux[1:(n - 1), 1]
names(sbux.ret) = starbx[2:n,1] #Assign the correct dates as names to all elements of the return vector
plot(sbux.ret, type="l")

# Compute the continuously compounded 1-month returns as difference in log prices
# rt=ln(Pt)-ln(Pt-1)
sbux.ccret = log(sbux[2:n,1]) - log(sbux[1:(n - 1),1])
names(sbux.ccret) = starbx[2:n,1] #Assign the correct dates as names to all elements of the return vector
head(sbux.ccret)
plot(sbux.ccret, type="l")

# Compare the simple and cc returns
sbux = cbind(sbux.ret, sbux.ccret)
plot(sbux.ret, type = "l", col = "blue", lwd = 2, ylab = "Return", main = "Monthly Returns on SBUX")
abline(h = 0) # Add horizontal line at zero
legend(x = "bottomright", legend = c("Simple", "CC"), lty = 1, lwd = 2, col = c("blue", "red")) # Add a legend
lines(sbux.ccret, col = "red", lwd="1") # Add the continuously compounded returns

# Exercise: Would it have been a good idea to invest in the SBUX stock over the period in our data set? In case you invested $1 in SBUX on 3/31/1993 (the first day in data), how much would
 that dollar be worth on 3/3/2008 (the last day in data)? What was the evolution of the value of that dollar over time ?

# Compute gross returns
sbux_gret = sbux.ret + 1

# Compute future values
sbux_fv = cumprod(sbux_gret) # function returns the cumulative multiplication results.

# Plot the evolution of the $1 invested in SBUX as a function of time
plot(sbux_fv, type = "l", col = "blue", lwd = 2, ylab = "Dollars", main = "Future Value of $1 invested in SBUX")

############## Homework 1
# Q1. What is the simple monthly return between the end of December 2004 and the end of January 2005?
# Q2. What is the continuously compounded monthly return between December 2004 and January 2005?
# Q3. Assume that all twelve months have the same return as the simple monthly return between the end of December 2004 and the end of January 2005. What would be the annual return with
 monthly compounding in that case?
# Q4. Compute the actual simple annual return between December 2004 and December 2005. Interpret
# Q5. Compute the actual annual continuously compounded return between December 2004 and December 2005. Interpret


######### Stilized facts

# 1. Daily returns show weak autocorrelations
acf(sbux.ccret)

# 2. Unconditional distribution of returns does not follow the normal distribution
hist(sbux.ccret, freq=F, breaks=12)
lines(density(sbux.ccret), col="red", )
lines(seq(min(sbux.ccret), max(sbux.ccret), length=n), dnorm(seq(min(sbux.ccret), max(sbux.ccret), length=n), mean(sbux.ccret), sd(sbux.ccret)), col="blue")

# 3. Standard deviation dominates the mean
mean(sbux.ccret)
sd(sbux.ccret)

# 4. Positive correlation of squared returns
sbux.ccret.2 = (sbux.ccret)^2
plot(sbux.ccret.2,  type="l")
acf(sbux.ccret.2)

################ Monthly stock price of quotes for multiple stocks from 2005 to 2010

sbux=read.table(file=file.choose(), header=TRUE, sep=",", dec=".")  # Starbucks stock
vbltx=read.table(file=file.choose(), header=TRUE, sep=",", dec=".") # Vanguard long term bond index fund
fmagx=read.table(file=file.choose(), header=TRUE, sep=",", dec=".") # Fidelity Magellan stock mutual fund

sbux = read.delim("/Users/karollgomez/Dropbox/Karoll/Cursos/EconometriaFinanciera/NotasClase/Intro/SBUX.csv",header=TRUE,sep = ",", dec = ".")
vbltx= read.delim("/Users/karollgomez/Dropbox/Karoll/Cursos/EconometriaFinanciera/NotasClase/Intro/VBLTX.csv",header=TRUE,sep = ",", dec = ".")
fmagx= read.delim("/Users/karollgomez/Dropbox/Karoll/Cursos/EconometriaFinanciera/NotasClase/Intro/FMAGX.csv",header=TRUE,sep = ",", dec = ".")


# Create merged price data
allprices <- cbind(sbux[,6], vbltx[,6 ], fmagx[,6])
all_prices <- data.frame(allprices)

# Rename columns
colnames(allprices) <- c("VBLTX", "FMAGX", "SBUX")
head(allprices)

# Calculate continuously compounded returns
# rt=ln(1+Rt)
# rt=ln(Pt)-ln(Pt-1)
all_returns = diff(log(allprices), lag=1)
all_returns1 = log(lag(allprices)) - log(allprices)
# rt=ln(Pt/Pt-1)
n = length(allprices)
all_returns2 = log(allprices[-1]/allprices[-n])

# Look at the return data
colnames(all_returns)
head(all_returns)
class(all_returns)
```

```
all_returns <- data.frame(all_returns)

# Plot returns after using the PerformanceAnalytics function chart.TimeSeries().
library(xts)
library(zoo)
library(PerformanceAnalytics)
chart.TimeSeries(all_returns, legend.loc = "bottom", main = " ")

# Create matrix with returns using coredata
return_matrix <- coredata(all_returns)
summary(return_matrix)

# Compute descriptive statistics by column using the base R function apply()
args(apply)
apply(return_matrix, 2, mean) # computed by columns apply(X, margin=2, ...)
apply(return_matrix, 2, var)
apply(return_matrix, 2, sd)

# Annualized simple mean : an estimate of the annual continuously compounded return is just 12 times the monthly continuously compounded return.
exp(12 * apply(return_matrix, 2, mean)) - 1

# Annualized standard deviation values: an estimate of the continuously compounded annual standard deviation is the square root of 12 times the monthly standard deviation.
exp(12 * apply(return_matrix, 2, sd)) - 1

# Generate four panel plots to see stylized facts on returns
par(mfrow = c(2, 2))
hist(return_matrix[, "VBLTX"], main = "VBLTX monthly returns",
     xlab = "VBLTX", probability = T, col = "slateblue1")
boxplot(return_matrix[, "VBLTX"], outchar = T, main = "Boxplot", col = "slateblue1")
plot(density(return_matrix[, "VBLTX"]), type = "l", main = "Smoothed density",
     xlab = "monthly return", ylab = "density estimate", col = "slateblue1")
qqnorm(return_matrix[, "VBLTX"], col = "slateblue1")
qqline(return_matrix[, "VBLTX"])
par(mfrow = c(1, 1))

# Show boxplot of three series on one plot
boxplot(return_matrix[, "VBLTX"], return_matrix[, "FMAGX"], return_matrix[, "SBUX"], names = colnames(return_matrix), col = "slateblue1")

######## Homework 2 ##############
# Look into the daily closing indices of the S&P 500 and the daily closing share prices (in US dollars) of the Apple Inc in the period of January 1985 – February 2011. Download it from
 Yahoo!Finance. With these data compute:
# Q1: Time series plots of the daily indices, the daily log returns, the weekly log returns, and the monthly log returns
# Q2: Histograms and Q-Q plots of the daily, weekly, and monthly log returns
# Q3: Autocorrelations of the daily, weekly, and monthly log returns, the squared daily, weekly, and monthly log returns, and the absolute daily, weekly, and monthly log returns
# Q4: Interpret results according with stylized features of financial returns:
# (a) Stationarity: The prices of an asset recorded over times are often not stationary due to, for example, the steady expansion of economy, the increase of productivity resulting from
 technology innovation, and economic recessions or financial crisis
# (b) Heavy tails. The probability distribution of return rt often exhibits heavier tails than those of a normal distribution.
# (c) Asymmetry. The distribution of return rt is often negatively skewed
# (d) Volatility clustering. This term refers to the fact that large price changes (i.e. returns with large absolute values) occur in clusters.
# (e) Long range dependence. The returns themselves hardly show any serial correlation, which, however, does not mean that they are independent.
# (f) Leverage effect. Asset returns are negatively correlated with the changes of their volatilities (Black 1976, Christie 1982).
```