Code for Finance workshop

Zhengkai Fu

07/07/2020

Load library

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 3.6.3

## Warning: package 'ggplot2' was built under R version 3.6.3

## Warning: package 'tibble' was built under R version 3.6.3

## Warning: package 'tidyr' was built under R version 3.6.3

## Warning: package 'dplyr' was built under R version 3.6.3

## Warning: package 'forcats' was built under R version 3.6.3
```

```
library(quantmod)
```

```
## Warning: package 'quantmod' was built under R version 3.6.3
```

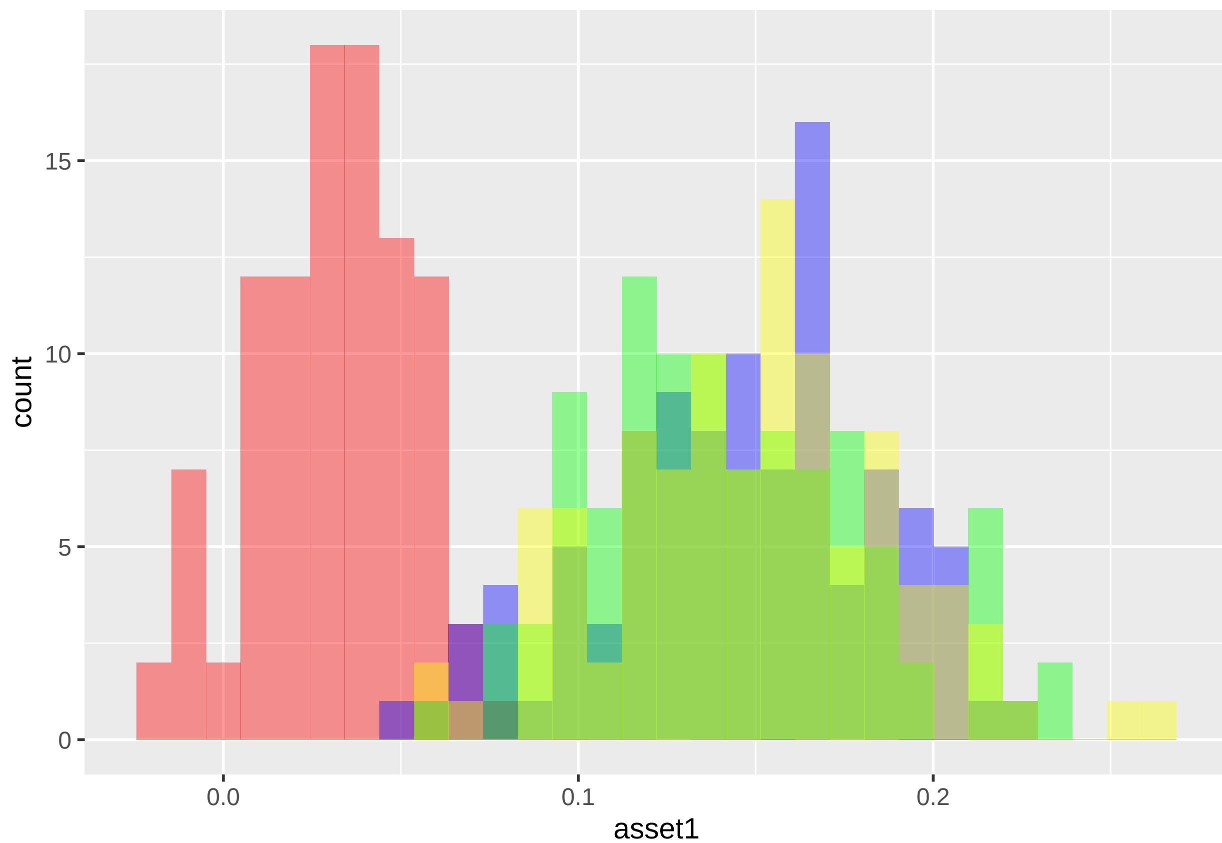
```
library(anytime)
library(xts)
```

Simulation for diversification

```
set.seed(202001303)

asset_1 <- data.frame(return = rnorm(100, mean = 0.03, sd = 0.02))

# perfect positive linear correlation
asset_2 <- data.frame(return = 0.09 + asset_1 * 1.8)
## theoretical mean: 0.09 + 0.03 * 1.8 = 0.144
## theoretical var: 1.8^2 * 0.02^2
## theoretical sd: 1.8 * 0.02 = 0.036
```

```r
# perfect negative linear correlation
asset_3 <- data.frame(return = 0.198 - asset_1 * 1.8)
## theoretical mean: 0.198 - 0.03 * 1.8 = 0.144
## theoretical var: 1.8^2 * 0.02^2
## theoretical sd: 1.8 * 0.02 = 0.036

# somewhere in between correlated
asset_4 <- data.frame(return = rnorm(100, mean = 0.144, sd = 0.036))


combined_return <- data.frame(cbind(asset_1, asset_2, asset_3, asset_4)) %>%
  `colnames<-`(c("asset1", "asset2", "asset3", "asset4"))


# plot 4 assets' returns
ggplot(combined_return) +
  geom_histogram(aes(asset1), fill = 'red', alpha = 0.4) +
  geom_histogram(aes(asset2), fill = 'blue', alpha = 0.4) +
  geom_histogram(aes(asset3), fill = 'green', alpha = 0.4) +
  geom_histogram(aes(asset4), fill = "yellow", alpha = 0.4)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
cor(asset_1, asset_2)
```

```
##        return
## return      1
```

```
cor(asset_1, asset_3)
```

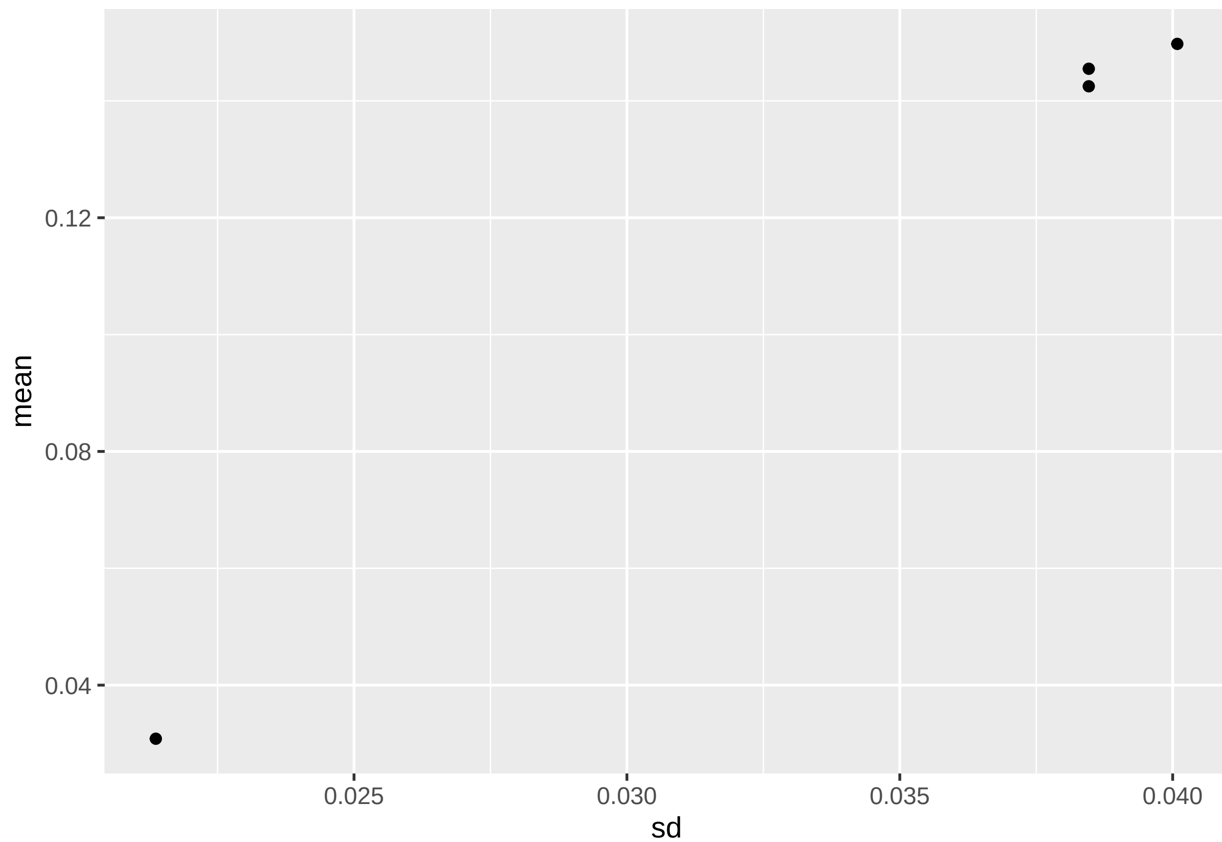
```
##        return
## return     -1
```

```
cor(asset_1, asset_4)
```

```
##             return
## return -0.07895565
```

```
# 4 assets' summary
## mean
mean_1 = mean(asset_1$return)
mean_2 = mean(asset_2$return)
mean_3 = mean(asset_3$return)
mean_4 = mean(asset_4$return)

## sd
```

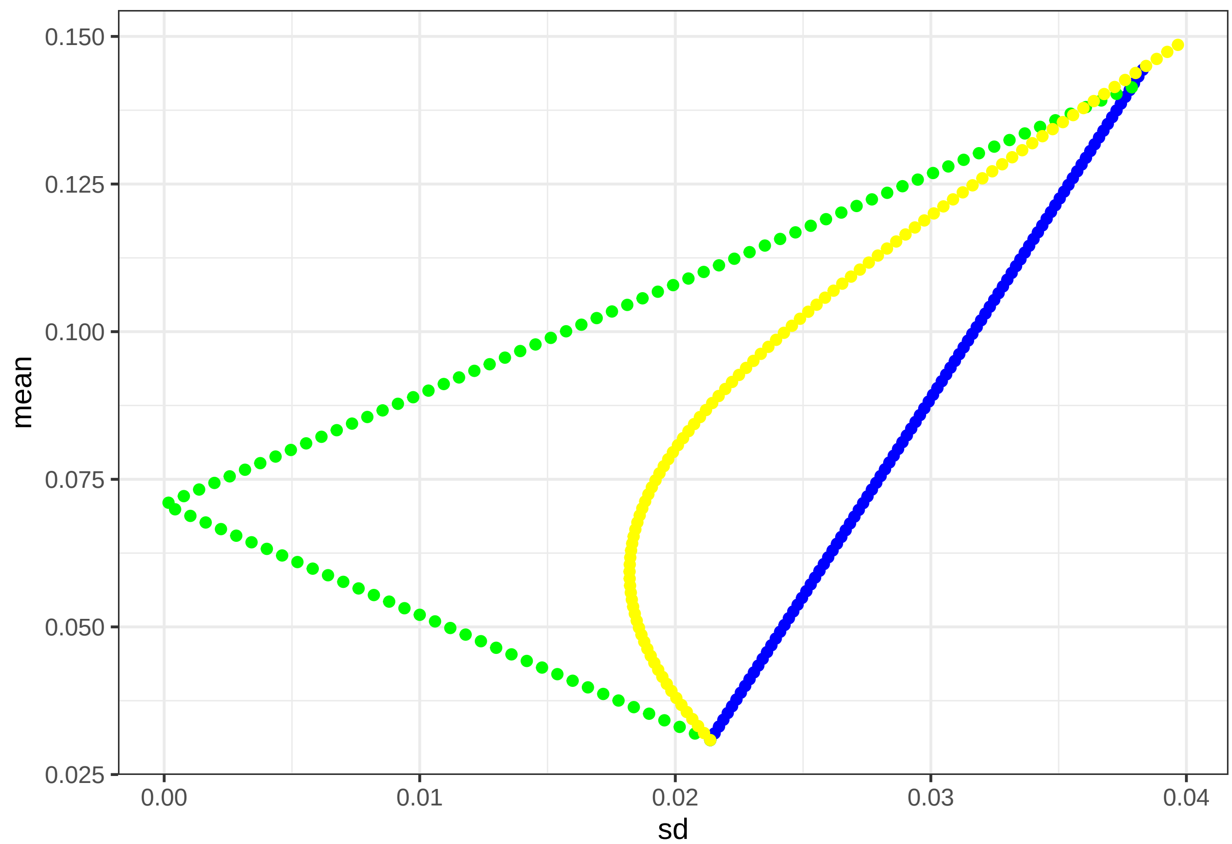
```
sd_1 = sqrt(var(asset_1$return))
sd_2 = sqrt(var(asset_2$return))
sd_3 = sqrt(var(asset_3$return))
sd_4 = sqrt(var(asset_4$return))

# plot result of 4 simulated assets
tibble(sapply(combined_return, sd), sapply(combined_return, mean)) %>%
  `colnames<-`(c("sd", "mean")) %>%
  ggplot(aes(x=sd, y=mean)) +
    geom_point()
```



Mixing assets

```
#### PERFECTLY POSITIVE ####
## mix options, sequence from 0.01 (1% of asset 1) to 1.00 (100% of asset 1)
mix <- seq(100)/100

# each column is a mix option
positive_porfolio <- data.frame(mix[1] * asset_1 + (1-mix[1]) * asset_2)
for (i in (2:100)){
  temp = mix[i] * asset_1 + (1-mix[i]) * asset_2
  positive_porfolio <- cbind(positive_porfolio, temp)
```

```
}

# change column names
col_names_mix <- c(paste0(1:100, "%"))
colnames(positive_porfolio) <- col_names_mix

positive_portfolio_result <- data.frame(sapply(positive_porfolio, sd), sapply(positive_porfolio, mean))
  `colnames<-`(c("sd", "mean"))

#### PERFECTLY NEGATIVE ####
negative_portfolio <- data.frame(mix[1] * asset_1 + (1-mix[1]) * asset_3)
for (i in (2:100)){
  temp = mix[i] * asset_1 + (1-mix[i]) * asset_3
  negative_portfolio <- cbind(negative_portfolio, temp)
}

# cahnge column names
colnames(negative_portfolio) <- col_names_mix

negative_portfolio_result <- data.frame(sapply(negative_portfolio, sd), sapply(negative_portfolio, mean
  `colnames<-`(c("sd", "mean"))

#### IN BETWEEN ####
middle_portfolio <- data.frame(mix[1] * asset_1 + (1-mix[1]) * asset_4)
for (i in (2:100)){
  temp = mix[i] * asset_1 + (1-mix[i]) * asset_4
  middle_portfolio <- cbind(middle_portfolio, temp)
}

# cahnge column names
colnames(middle_portfolio) <- col_names_mix

middle_portfolio_result <- data.frame(sapply(middle_portfolio, sd), sapply(middle_portfolio, mean)) %>%
  `colnames<-`(c("sd", "mean"))


#### plot combination ####
ggplot(NULL, aes(x=sd, y=mean)) +
  geom_point(data = positive_portfolio_result, color = 'blue') +
  geom_point(data = negative_portfolio_result, color = "green") +
  geom_point(data = middle_portfolio_result, color = "yellow") +
  theme_bw()
```



Compare volatility

Get Tesla, NASDAQ data

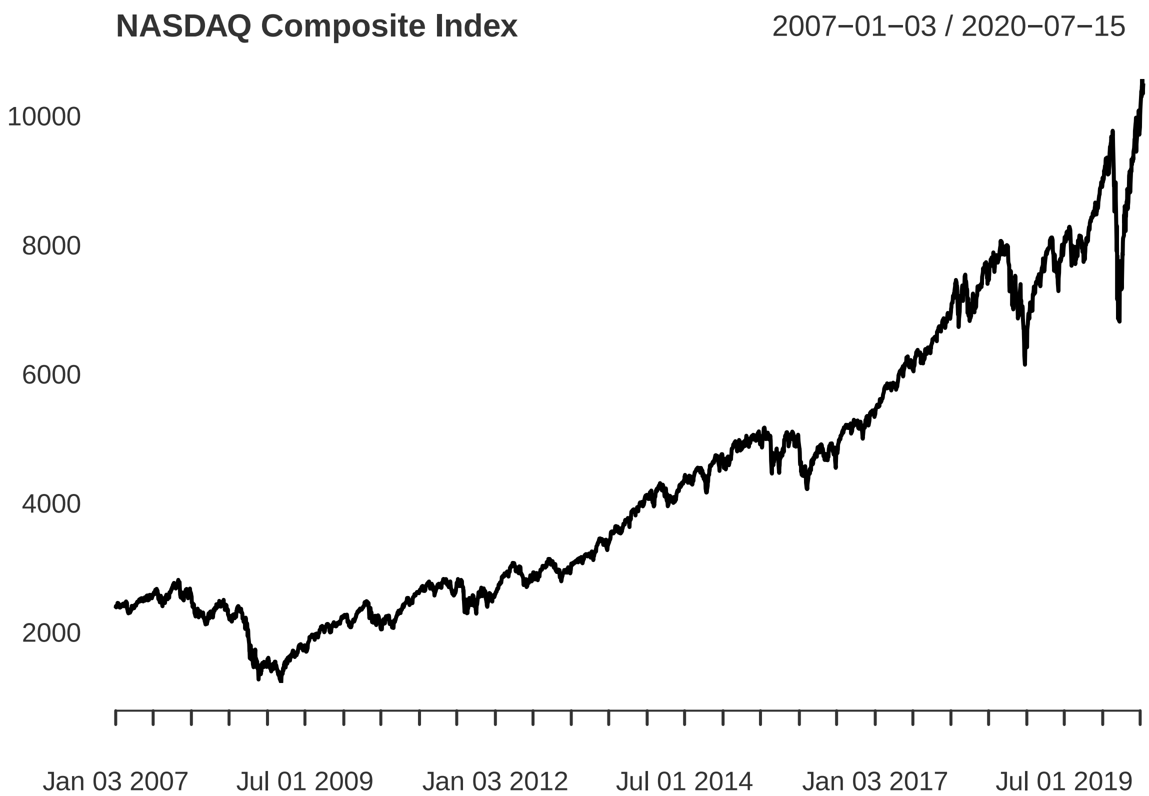
```
nasdaq <- getSymbols("^IXIC", src = "yahoo")
```

```
## '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.
```

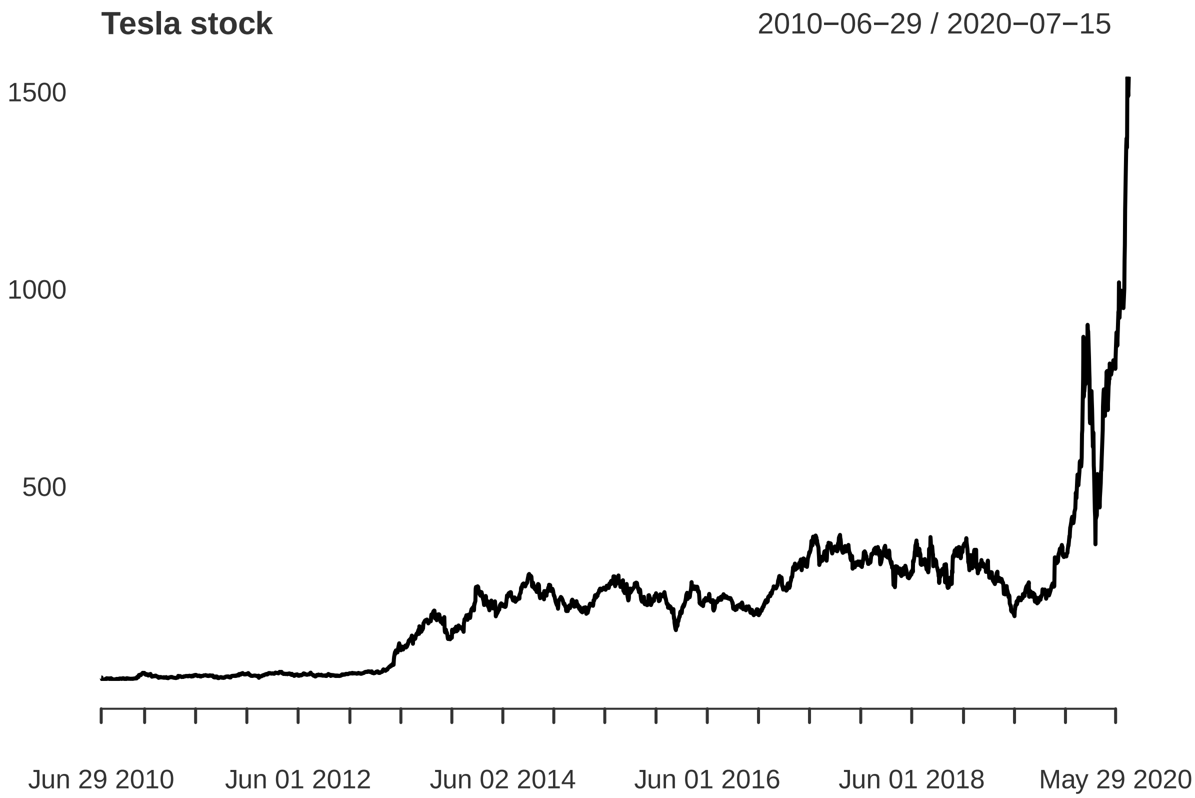
```
tesla <- getSymbols("TSLA", src = "yahoo")

# use close price
## despite of command above, the result of nasdaq is not stored in "nasdaq" but in "IXIC"
## same for tesla
nasdaq <- IXIC$IXIC.Close
tesla <- TSLA$TSLA.Close
```

Plot 2 assets

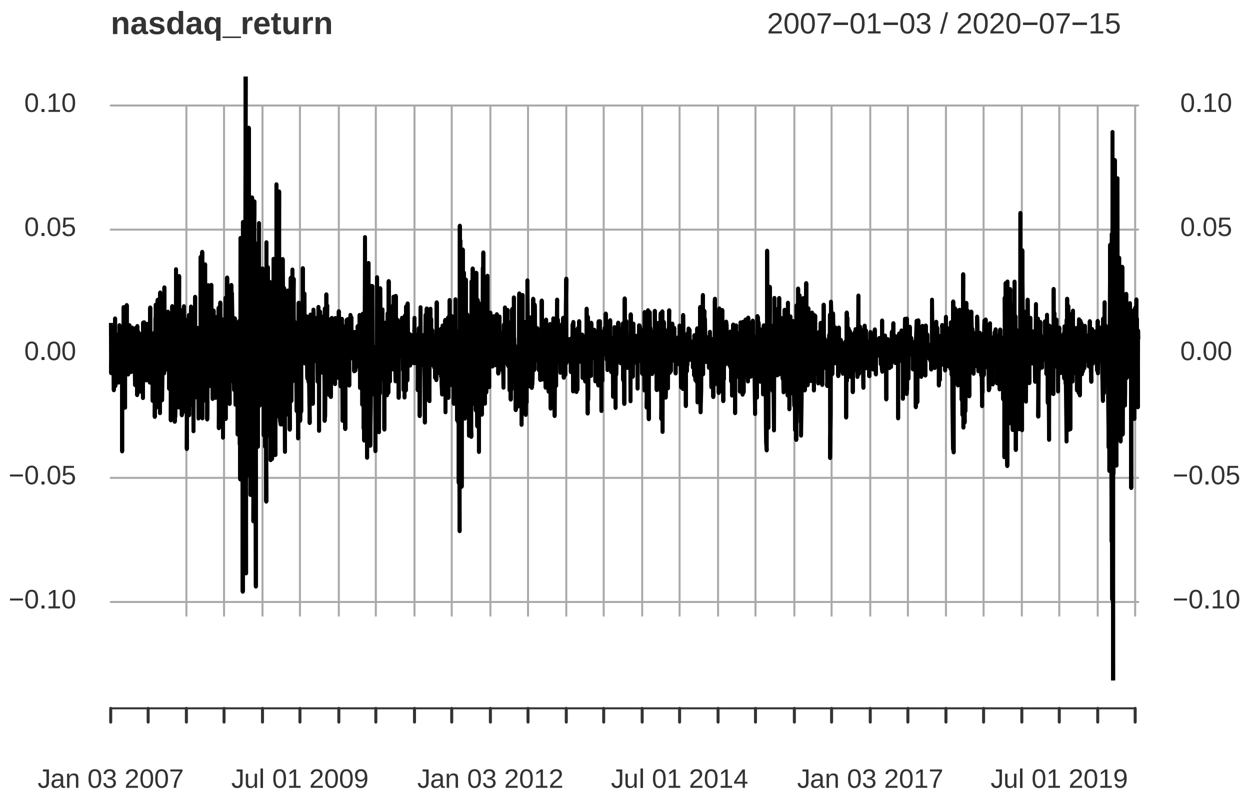
```r
plot.xts(nasdaq, grid.col = 'white',  yaxis.right = FALSE,
         main = 'NASDAQ Composite Index')
```



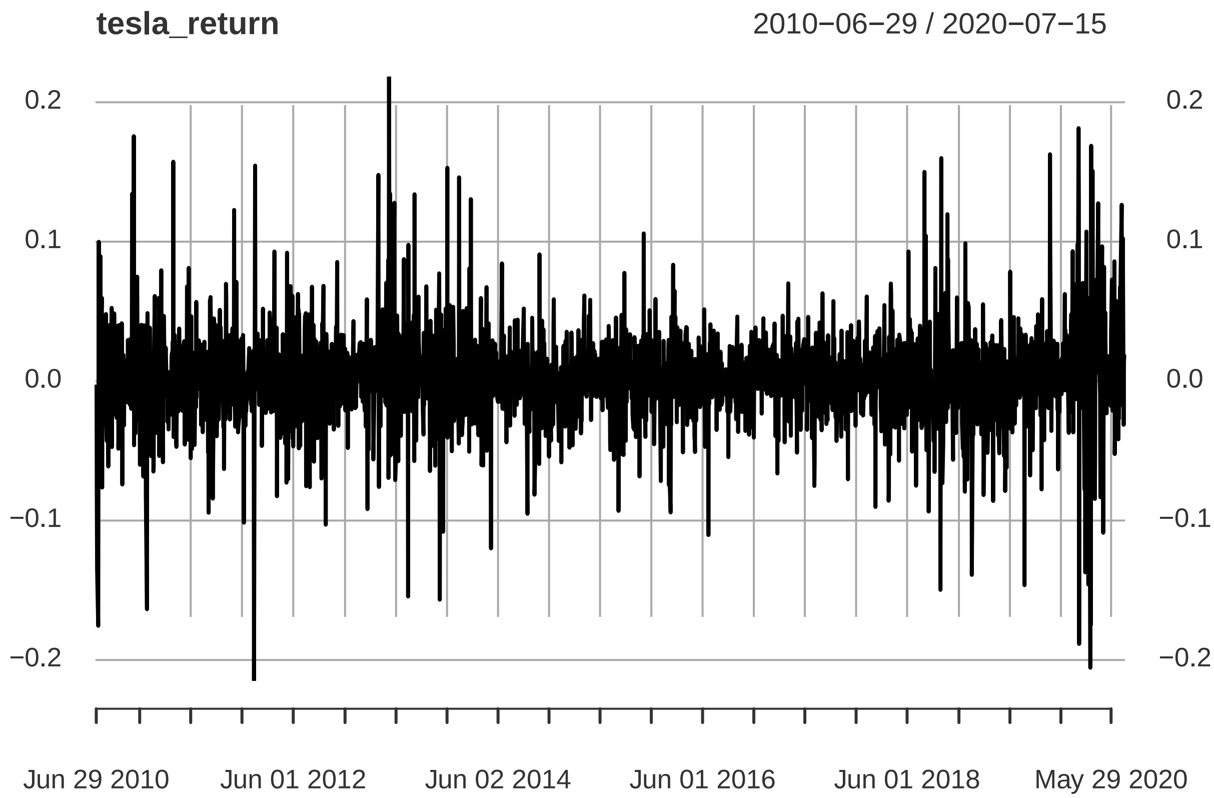

```r
plot.xts(tesla, grid.col = 'white', yaxis.right = FALSE,
         main = 'Tesla stock')
```



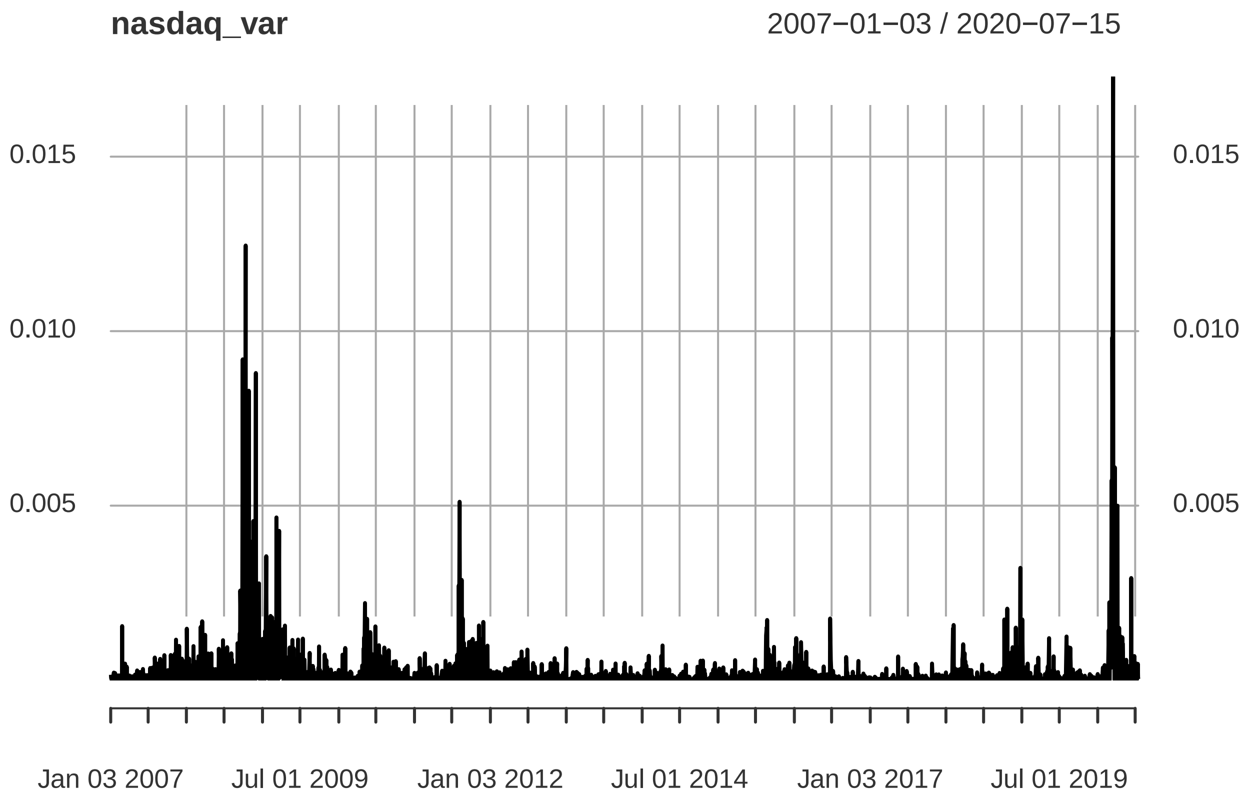

```r
# plot returns
nasdaq_return <- diff(log(nasdaq))
tesla_return <- diff(log(tesla))

plot.xts(nasdaq_return)
```



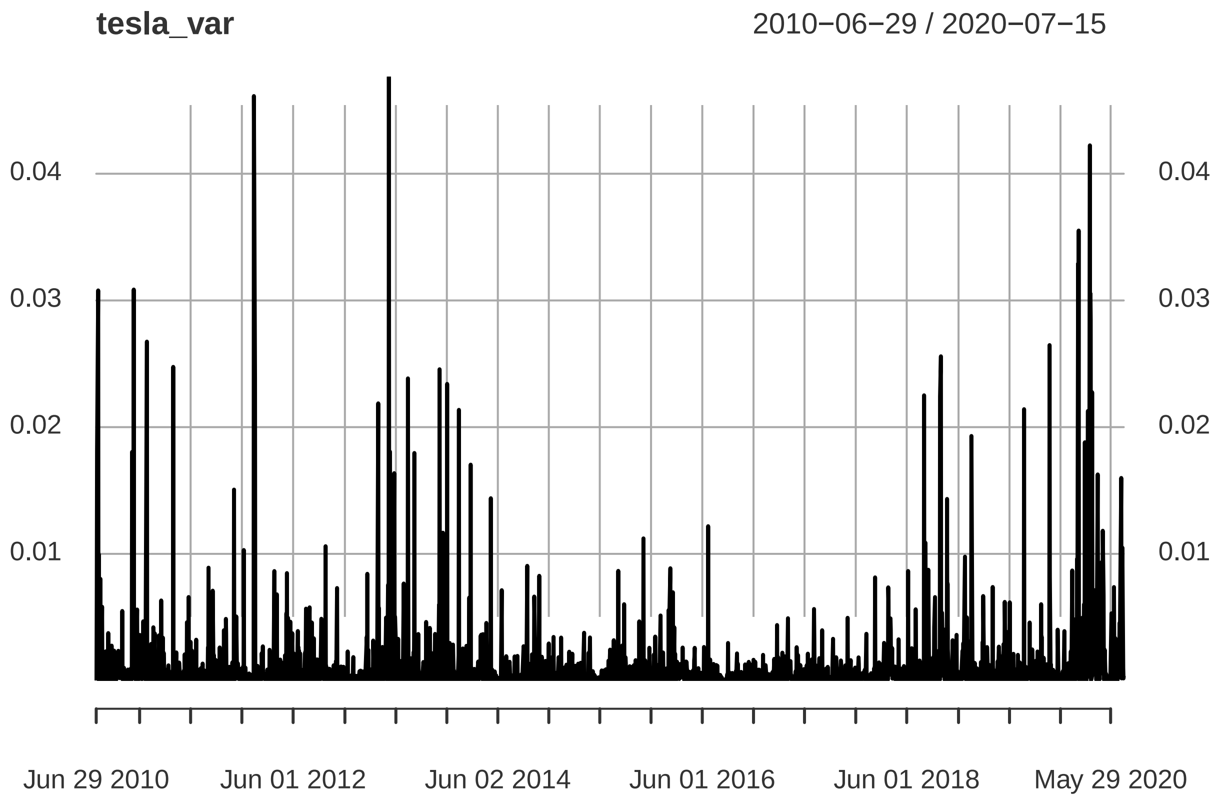

```r
plot.xts(tesla_return)
```




```r
# plot dynamic variance
nasdaq_var <- nasdaq_return^2
tesla_var <- tesla_return^2

plot.xts(nasdaq_var)
```




```r
plot.xts(tesla_var)
```




```
# There are better models to actually fit volatility. Methods above are just for simple demonstration p
```

Table for comparison

```
nasdaq_row <- cbind(mean(nasdaq_return, na.rm=TRUE), mean(sqrt(nasdaq_var), na.rm=TRUE))
tesla_row <- cbind(mean(tesla_return, na.rm=TRUE), mean(sqrt(tesla_var), na.rm=TRUE))

table <- data.frame(rbind(nasdaq_row, tesla_row))
colnames(table) <- c("average return", "average sd")
rownames(table) <- c("NASDAQ", "TESLA")

kableExtra::kable(table, format = "markdown")
```

	average return	average sd
NASDAQ	0.0004319	0.0092834
TESLA	0.0016495	0.0234192