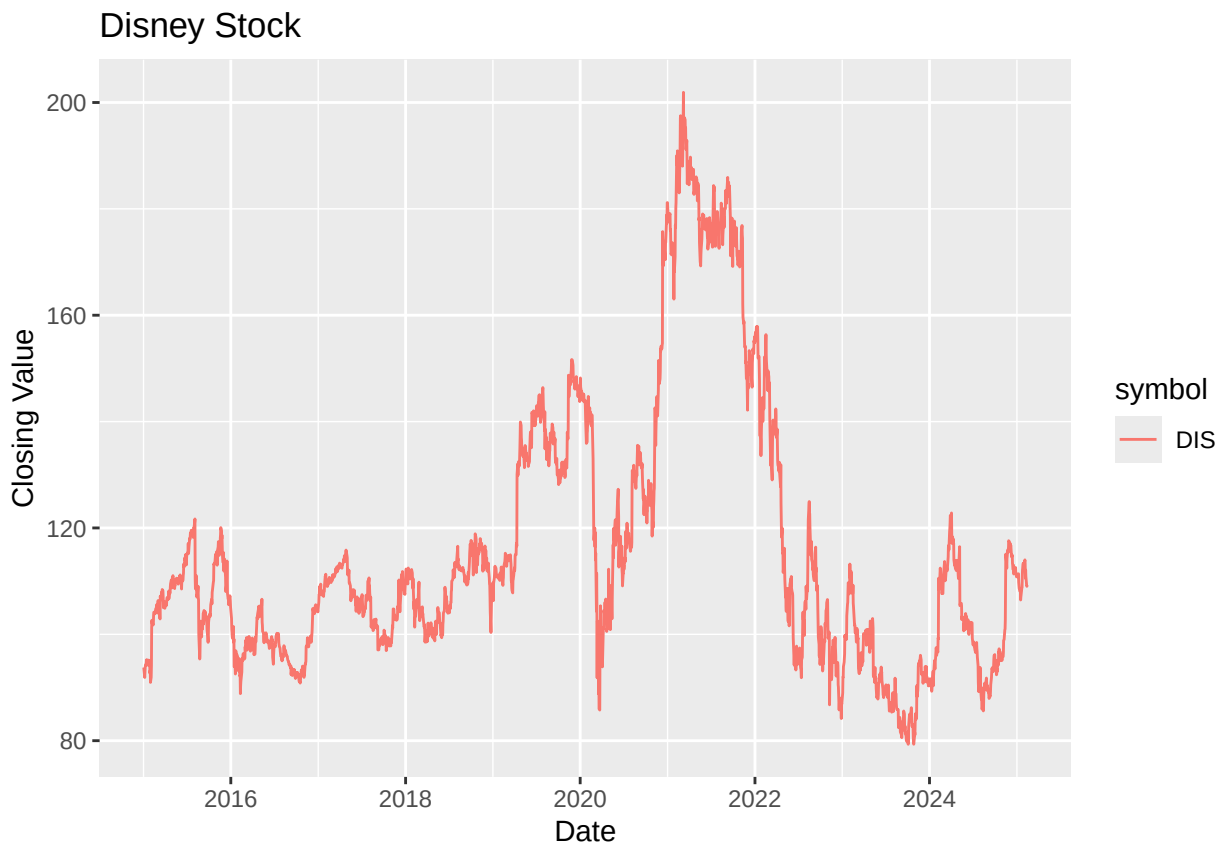
TidyQuant

```
library(tidyquant)
```

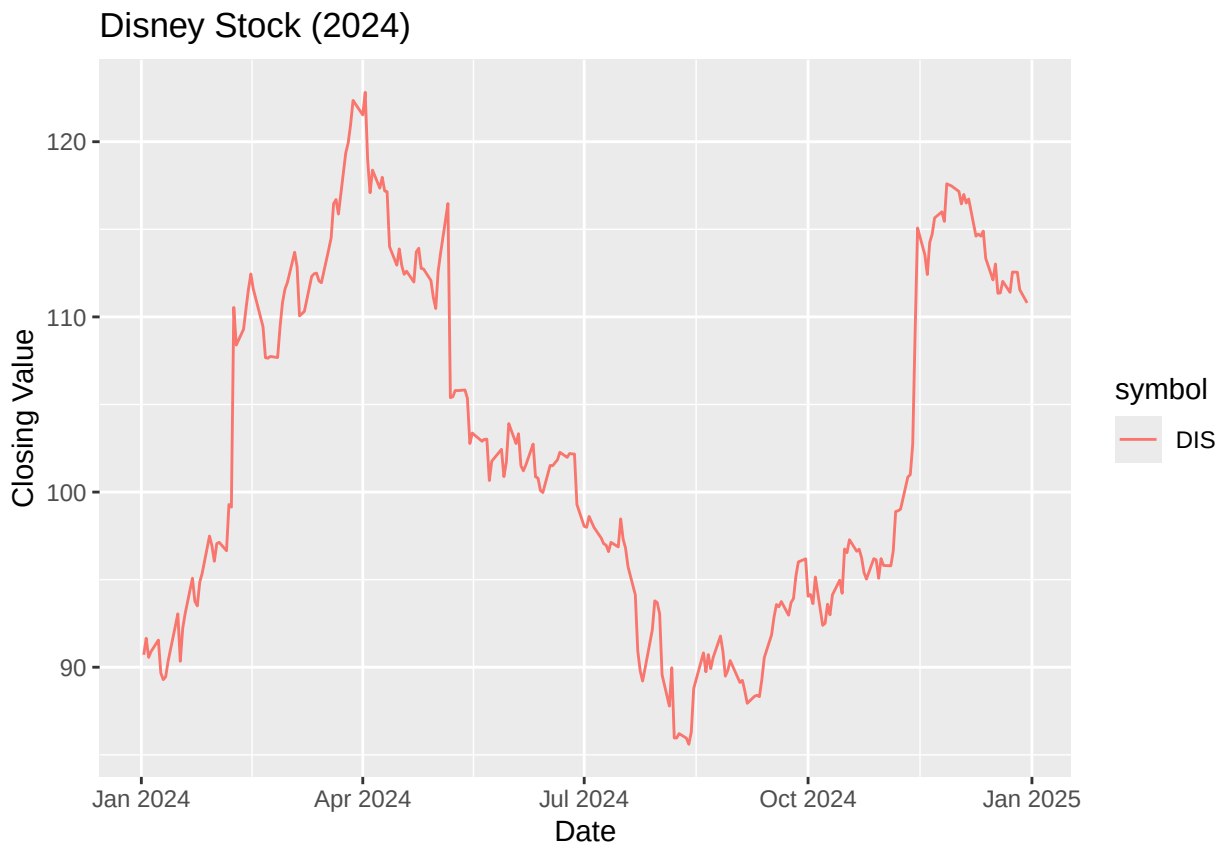
```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
## -- Attaching core tidyquant packages ------------------------ tidyquant 1.0.10 --
## v PerformanceAnalytics 2.0.8      v TTR                0.24.4
## v quantmod             0.4.26     v xts                0.14.1
## -- Conflicts ----------------------------------------- tidyquant_conflicts() --
## x zoo::as.Date()                 masks base::as.Date()
## x zoo::as.Date.numeric()         masks base::as.Date.numeric()
## x PerformanceAnalytics::legend() masks graphics::legend()
## x quantmod::summary()            masks base::summary()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(ggplot2)

Disney <- tq_get(x="DIS")

ggplot(Disney, aes(x=date, y=close, color=symbol)) +
  geom_line() +
  labs(title = "Disney Stock",
       x ="Date", y ="Closing Value")
```




```
disney2024 <- tq_get(x="DIS", get="stock.prices", from="2024-01-01", to="2024-12-31")

ggplot(disney2024, aes(x=date, y=close, color=symbol)) +
  geom_line() +
  labs(title = "Disney Stock (2024)",
       x ="Date", y ="Closing Value")
```




```
library(tidyquant)
library(ggplot2)
library(dplyr)
```

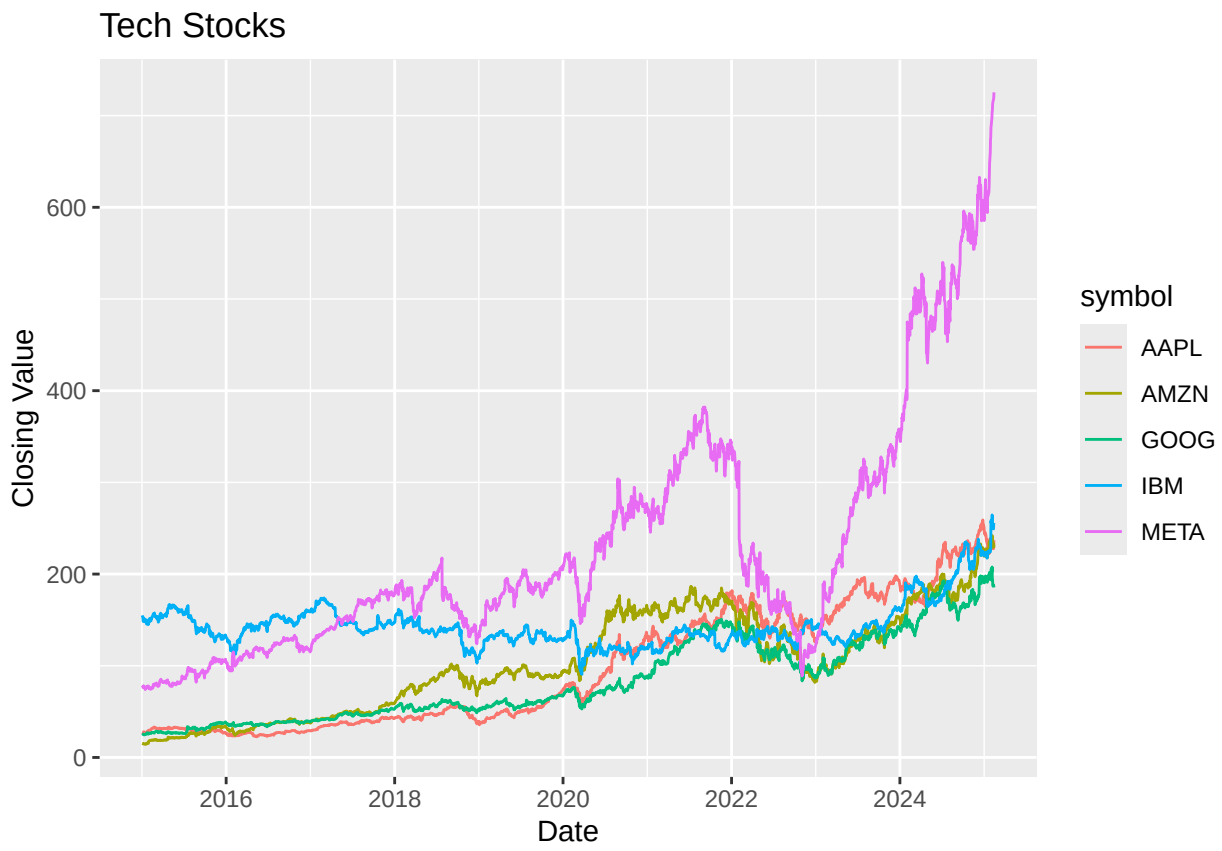
```
##
## ########################## Warning from 'xts' package ##########################
## #                                                                              #
## # The dplyr lag() function breaks how base R's lag() function is supposed to  #
## # work, which breaks lag(my_xts). Calls to lag(my_xts) that you type or       #
## # source() into this session won't work correctly.                             #
## #                                                                              #
## # Use stats::lag() to make sure you're not using dplyr::lag(), or you can add #
## # conflictRules('dplyr', exclude = 'lag') to your .Rprofile to stop            #
## # dplyr from breaking base R's lag() function.                                 #
## #                                                                              #
## # Code in packages is not affected. It's protected by R's namespace mechanism #
## # Set 'options(xts.warn_dplyr_breaks_lag = FALSE)' to suppress this warning.  #
## #                                                                              #
## ################################################################################

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:xts':
##
##     first, last
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

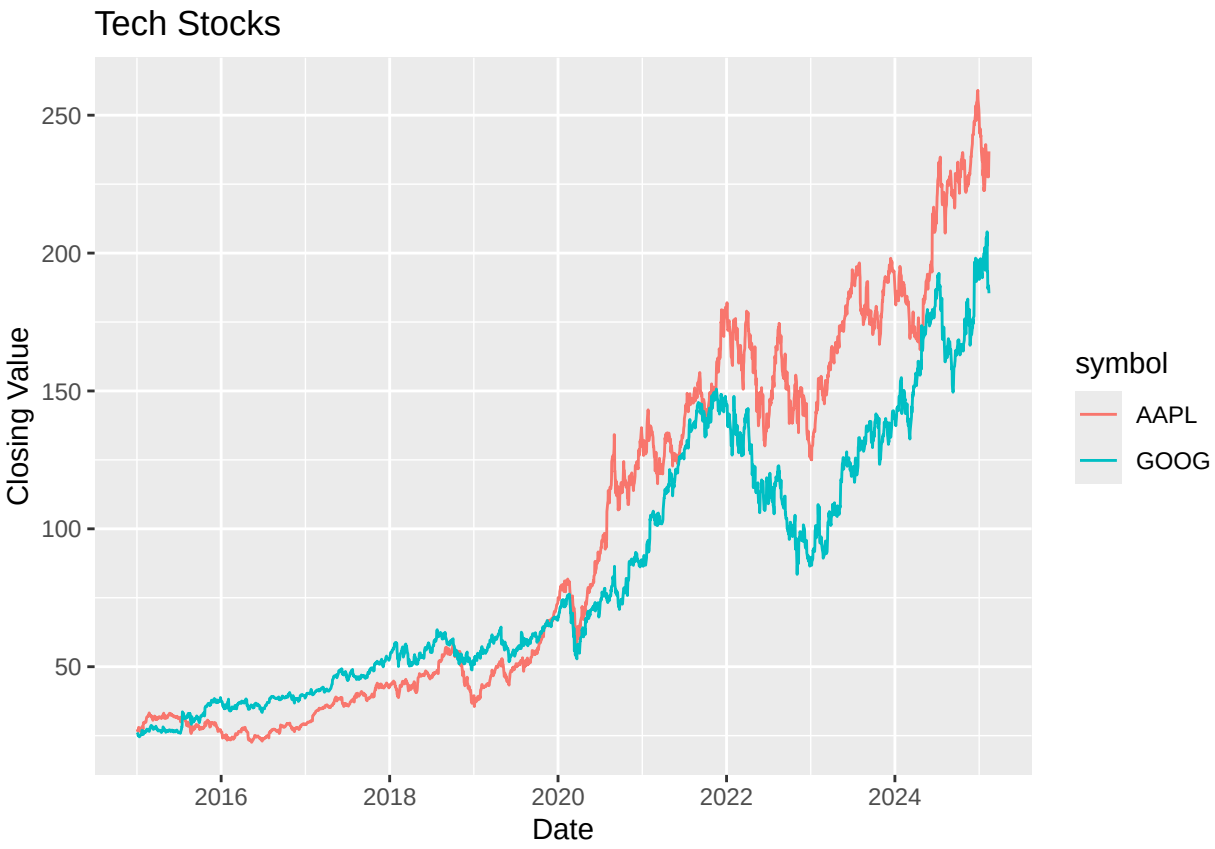
```
tech_stocks <- tq_get(c("AMZN","AAPL","GOOG","IBM","META"), get="stock.prices")
tech_stocks <- tech_stocks %>% filter(!is.na(close))

str(tech_stocks)
```

```
## tibble [12,720 x 8] (S3: tbl_df/tbl/data.frame)
##  $ symbol  : chr [1:12720] "AMZN" "AMZN" "AMZN" "AMZN" ...
##  $ date    : Date[1:12720], format: "2015-01-02" "2015-01-05" ...
##  $ open    : num [1:12720] 15.6 15.4 15.1 14.9 15 ...
##  $ high    : num [1:12720] 15.7 15.4 15.1 15.1 15.2 ...
##  $ low     : num [1:12720] 15.3 15 14.6 14.8 14.8 ...
##  $ close   : num [1:12720] 15.4 15.1 14.8 14.9 15 ...
##  $ volume  : num [1:12720] 55664000 55484000 70380000 52806000 61768000 ...
##  $ adjusted: num [1:12720] 15.4 15.1 14.8 14.9 15 ...
```

```
ggplot(tech_stocks, aes(x = date, y = close, color = symbol)) +
  geom_line() +
  labs(title = "Tech Stocks",
       x = "Date", y = "Closing Value")
```




```
df = filter(tech_stocks, symbol=="AAPL" | symbol=="GOOG")

ggplot(df, aes(x = date, y = close, color = symbol)) +
  geom_line() +
  labs(title = "Tech Stocks",
       x = "Date", y = "Closing Value")
```


Tech Stocks
Closing Value
250
200
150
100
50
2016
2018
2020
2022
2024
Date
symbol
AAPL
GOOG