COMPUTATIONAL FINANCE & RISK MANAGEMENT

UNIVERSITY of WASHINGTON

Department of Applied Mathematics

More Plots of Financial Time Series

CFRM 425 (008)

R Programming for Quantitative Finance

References/Reading/Topics

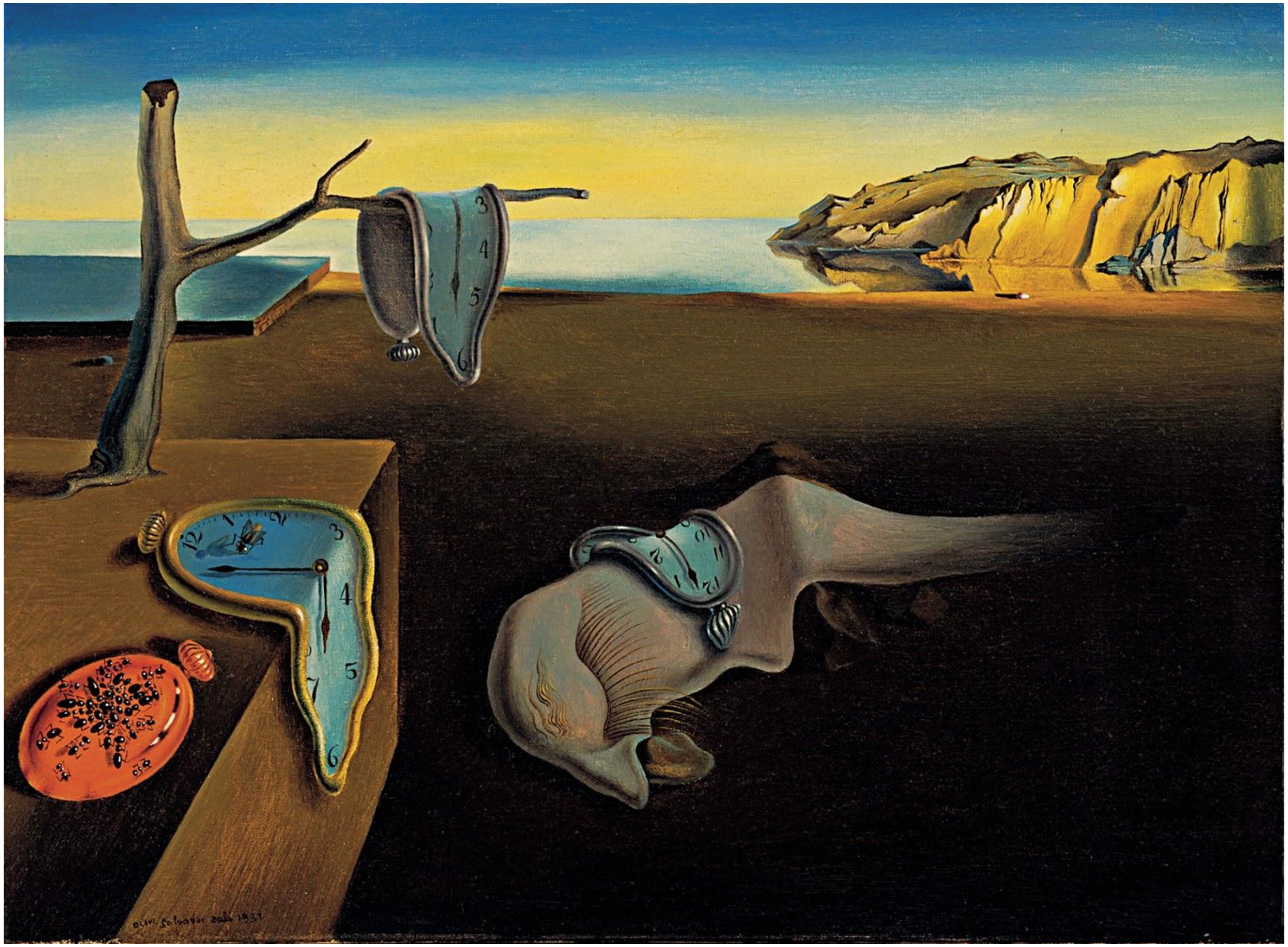
- These slides
- Hanson, *Quantitative Finance Applications in R: Plotting xts Time Series* (2014)

  https://blog.revolutionanalytics.com/2014/01/quantitative-finance-applications-in-r-plotting-xts-time-series.html

- Remark: This slide presentation content is for the most part not discussed explicitly in the book

- Topics:
  - Price series plots with technical indicators: **`chartSeries(.)`** in the quantmod package
    - ➢ Series of closing prices
    - ➢ Candlestick charting
  - Plots of multiple cumulative returns

Price series plots with technical indicators: **`chartSeries(.)`** in the quantmod package



quantmod::chartSeries(.)

- Use the same AMZN data as before
- Select adjusted prices only

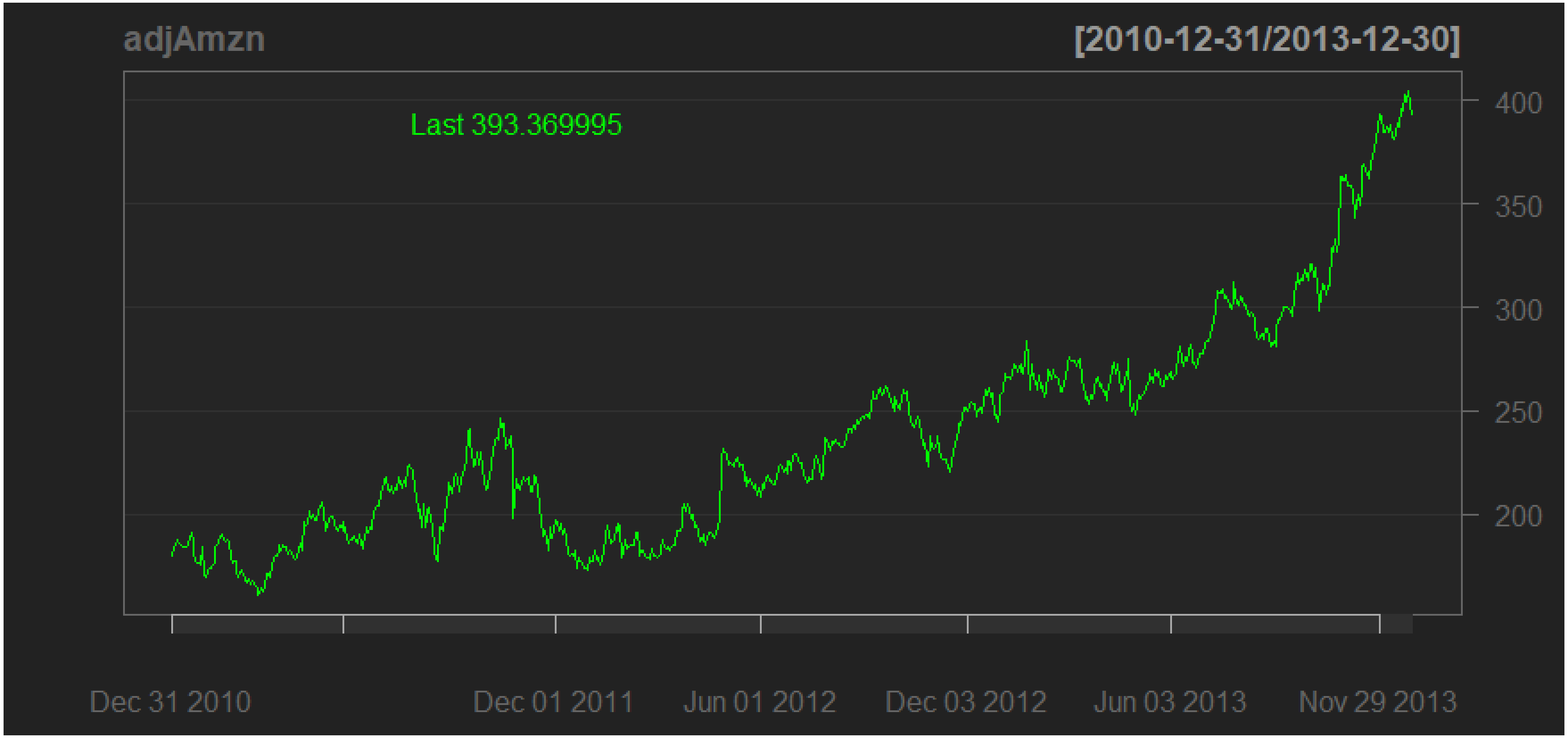
```
## ChartSeries plot:
# Basic: No parameter settings:
adjAmzn <- Ad(AMZN)
chartSeries(adjAmzn)

# Some customisations:
chartSeries(Ad(AMZN), subset = "2012", theme = "white" )

# up.col sets color for line
chartSeries(Ad(AMZN), subset = "2012-01::2012-06",
            theme = chartTheme("white", up.col = "darkblue" ),
            major.ticks="months")
```

quantmod::chartSeries(.)

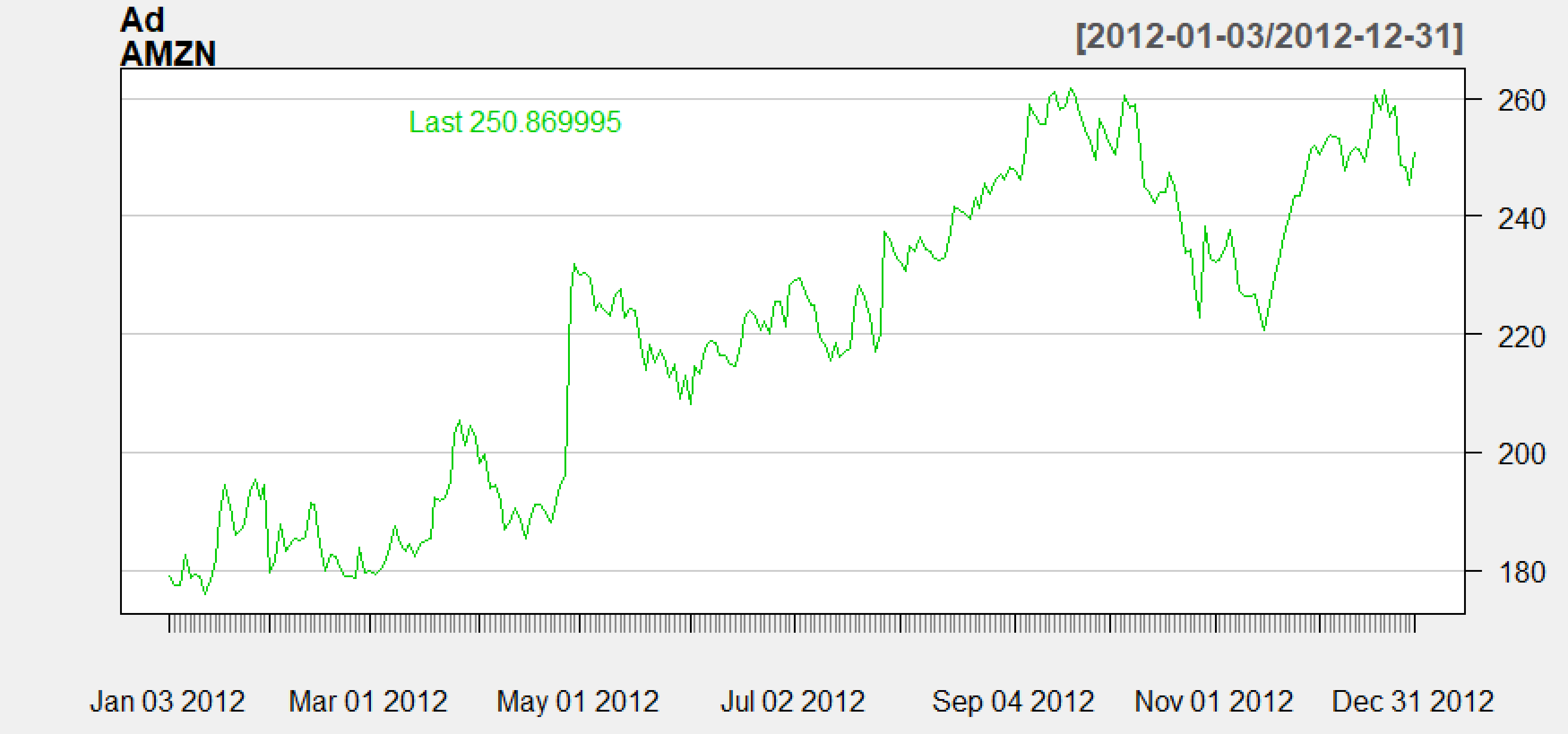
```
## ChartSeries plot:
# Basic: No parameter settings:
adjAmzn <- Ad(AMZN)
chartSeries(adjAmzn)
```



quantmod::chartSeries(.)

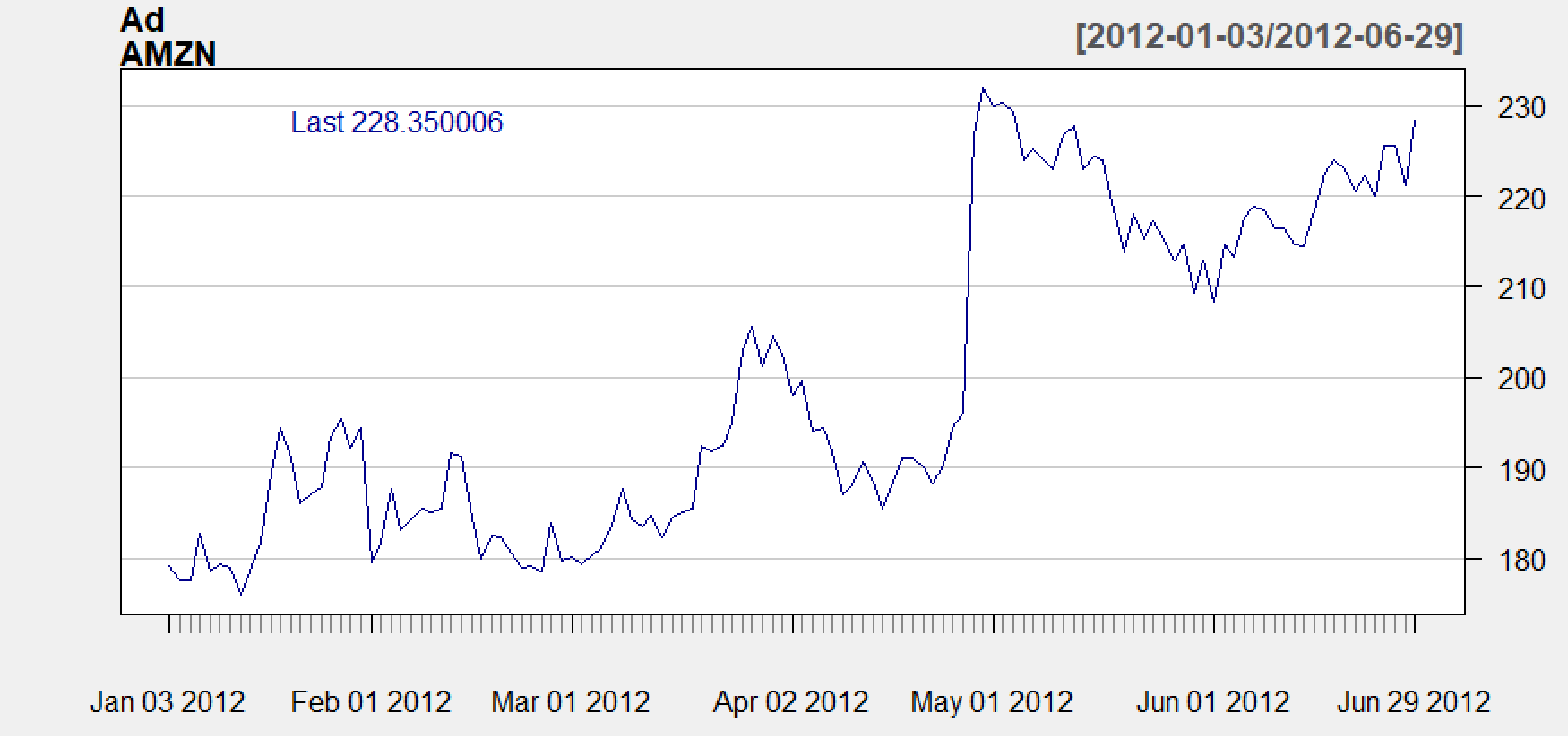
```
# Some customisations:

chartSeries(Ad(AMZN), subset = "2012", theme = "white" )
```



quantmod::chartSeries(.)

```
# up.col sets color for line
chartSeries(Ad(AMZN), subset = "2012-01::2012-06",
            theme = chartTheme("white", up.col = "darkblue" ),
            major.ticks="months")
```



quantmod::`chartSeries(.)`

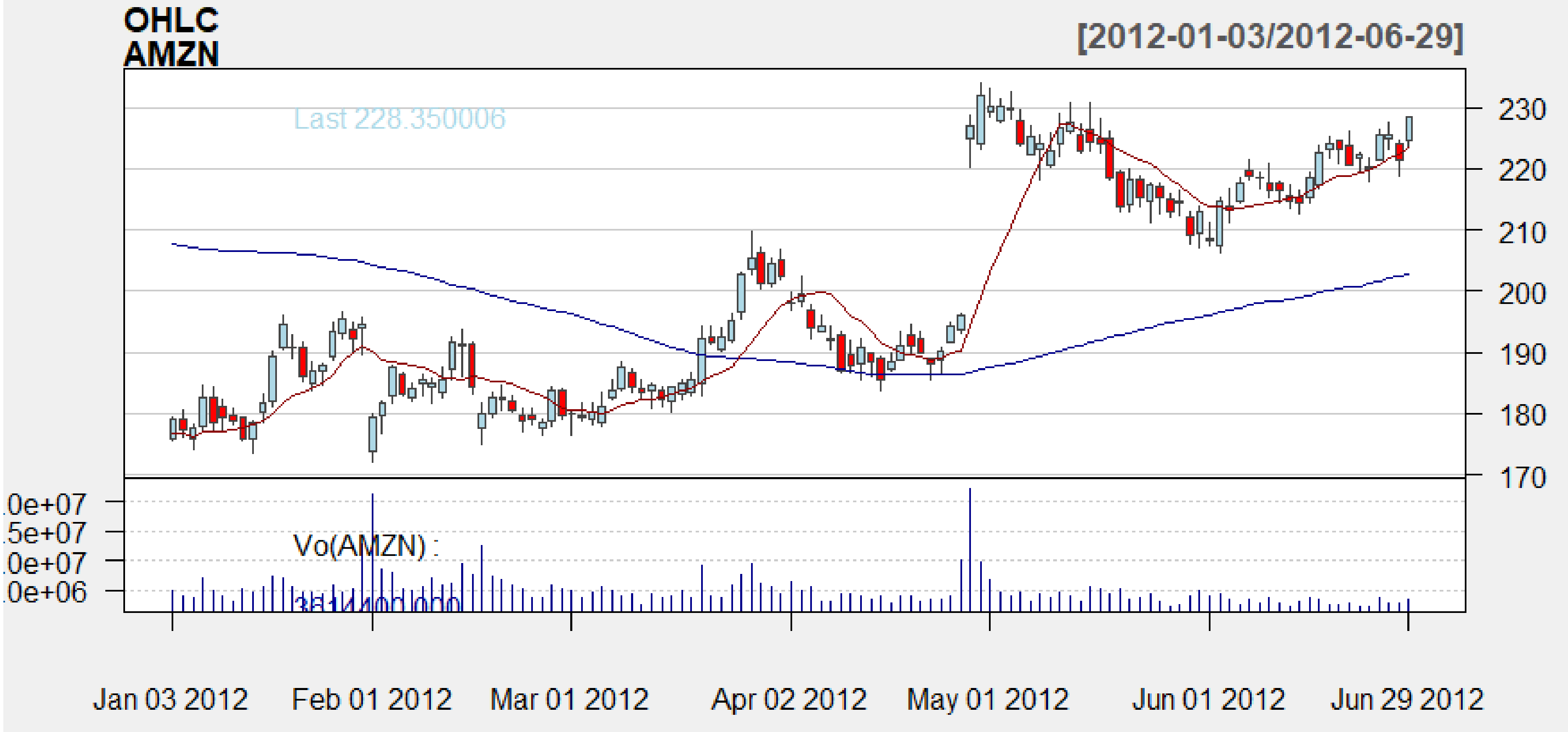
- Candlestick charting
- Use OHLC(.) function in quantmod
- We will also
  - Overlay technical indicators
  - Add a subpanel for volume

quantmod::chartSeries(.)

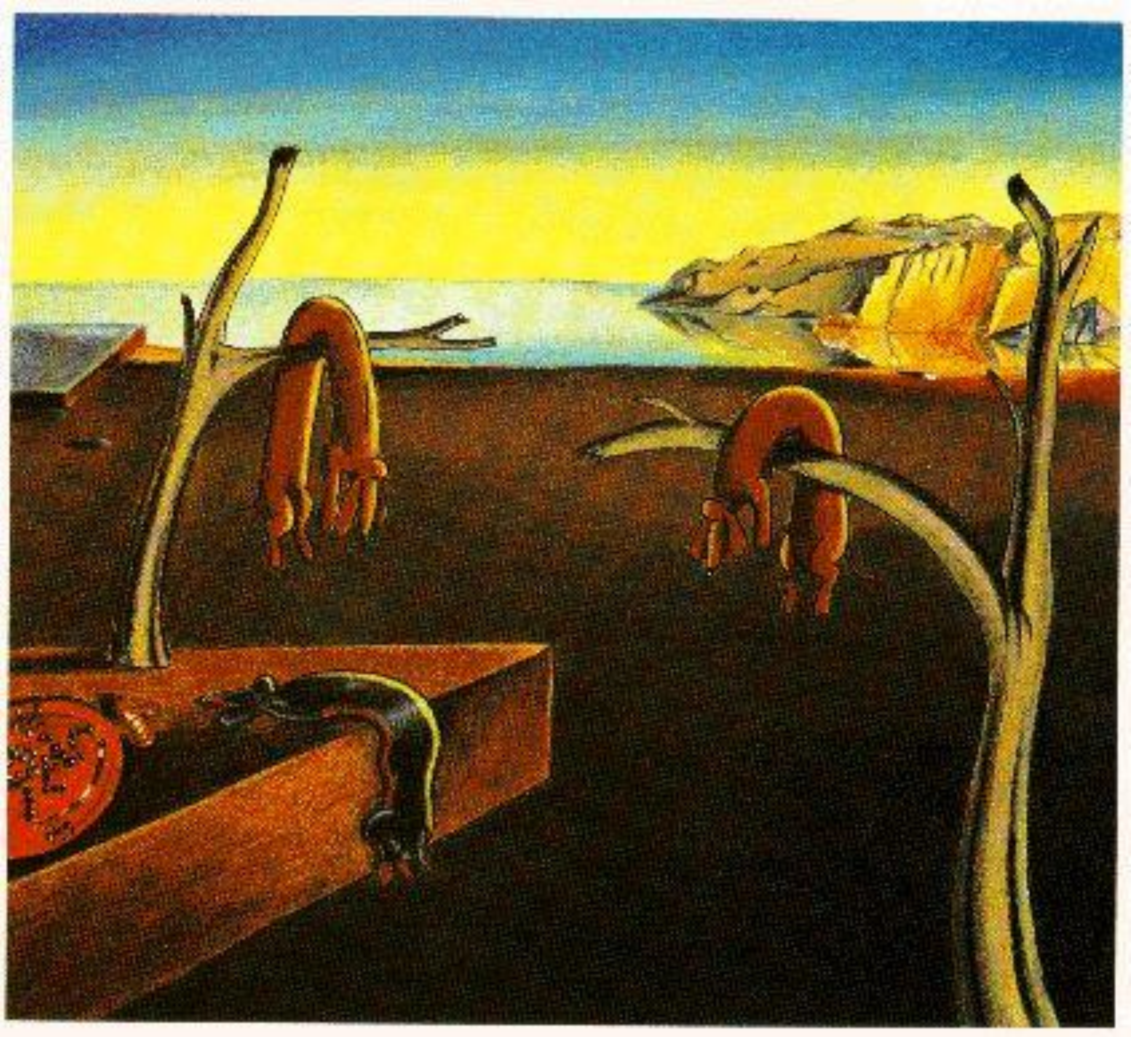
```
# Add candles:
chartSeries(OHLC(AMZN), subset = "2012-01::2012-06",
            theme = chartTheme("white", up.col = "lightblue",
                               dn.col = "red"),
            major.ticks="months", minor.ticks = FALSE)
addSMA(n = 100, col = "darkblue")    # addSMA(.) is overlayed
addSMA(n = 10, col = "darkred")      # addSMA(.) is overlayed
# addTA(.) plots are placed in panels below the plot:
addTA(Vo(AMZN), col = "darkblue", type = "h")
```

quantmod::chartSeries(.)

- With OHLC => candlesticks



Plots of Multiple Returns
Plots of Cumulative Returns



Plots of Cumulative Returns

- Use overload of **`plot(.)`** for zoo objects
- The zoo class is the parent of the xts class (in object-oriented parlance)
- Procedure:
  - Convert to different time interval if desired (eg monthly etc)
  - Calculate cumulative returns by dividing each (adjusted) price in the series by the initial price
  - Downcast the xts object to a zoo object
  - Call **plot(returns as zoo object)** – overloaded for a zoo input
- For multiple assets:
  - Loop through symbols to convert to different time interval (if desired)
  - Merge the adjusted prices for each asset into a new xts object
  - Put the initial prices for each into a vector
  - Loop through each column of the merged xts object and divide each price by its initial price
  - Downcast to a zoo object
  - Plot all cumulative returns

Plots of Cumulative Returns

```
syms <- c("AMZN", "SPY", "IBM")
getSymbols(Symbols = syms, from = "2010-12-31",
           to = "2017-12-31")
head(AMZN, 3)
tail(SPY, 3)
head(IBM, 3)

## Conversion to monthly data and then extract
## Adjusted Closing prices:
# Remark: Convert to months (or weeks, quarters etc) BEFORE
# filtering out the adjusted prices.  Will save yourself
# some hassle.
for(symbol in syms) {
  x <- get(symbol)  # Converts from string to object of same name
  x <- to.period(x, "months")
  colnames(x) <- gsub("x", symbol, colnames(x))  # Restore column name
  assign(symbol, x) # Assign monthly returns to sec code object name
}
```

Plots of Cumulative Returns

```
# Now, after adjusting to monthly, extract the
# adjusted closing prices only:
adjPrices <- merge(Ad(AMZN), Ad(SPY), Ad(IBM))
head(adjPrices, 3)
class(adjPrices)    # class that adjPrices object is (xts, derives from zoo)
typeof(adjPrices)   # types held in time series (double)

# Now, let's calculate the cumulative returns:
initPrices <- c(as.numeric(adjPrices[1, 1]),
                as.numeric(adjPrices[1, 2]),
                as.numeric(adjPrices[1, 3]))


cumuRtns <- adjPrices
for(i in 1:3) {
  cumuRtns[,i] <- adjPrices[,i]/initPrices[i]
}
```

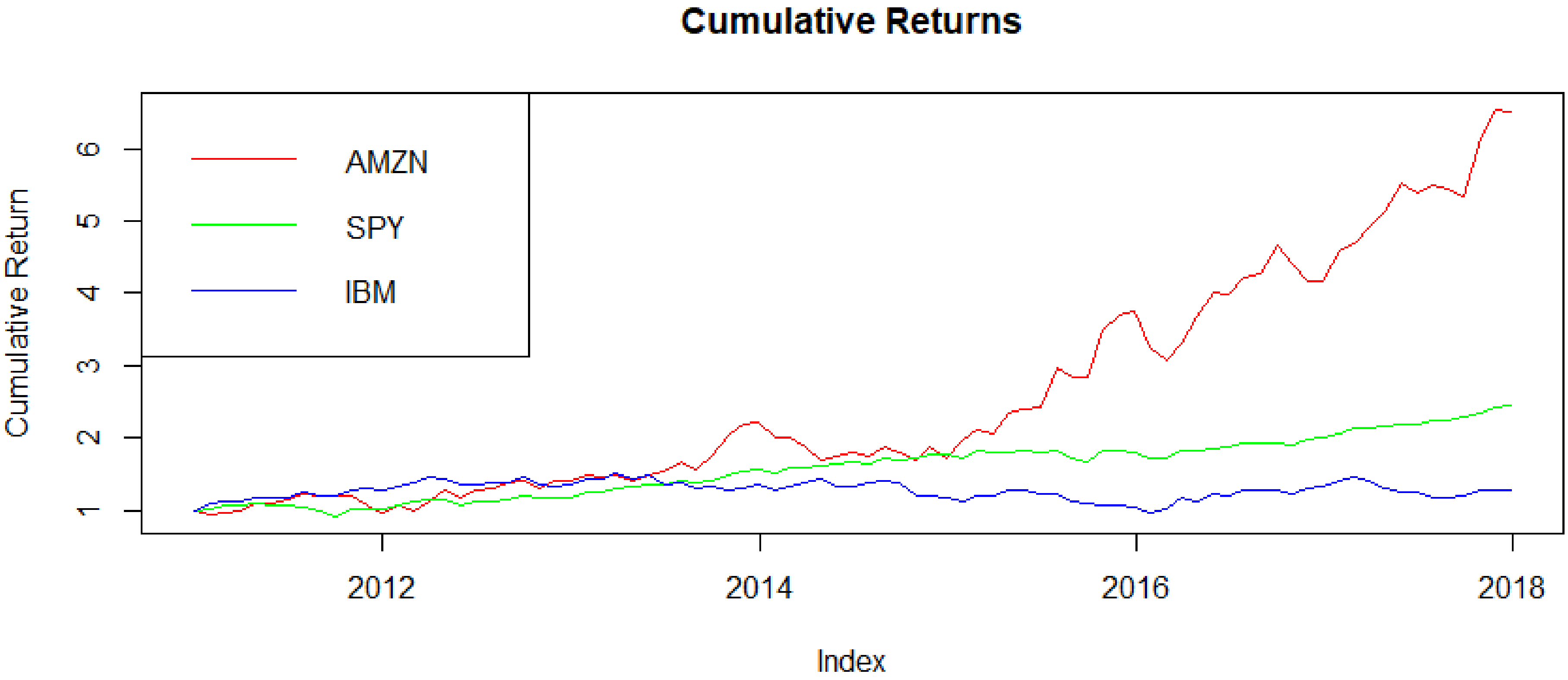
Plots of Cumulative Returns

```
# Now, to plot the cumulative returns, the easiest
# way is to downcast the cumuRtns xts object to
# a zoo object:
zoo.cumuRtns <- as.zoo(cumuRtns)


# Then, call the overload of the plot(.)
# function for zoo objects. screens = 1 puts
# all return series in the same plot:
plot(zoo.cumuRtns, ylab = "Cumulative Return",
     main = "Cumulative Returns", col = rainbow(ncol(zoo.cumuRtns)),
     screens = 1)
# Set a legend in the upper left hand corner to match color to return series
legend(x = "topleft", legend = c("AMZN", "SPY", "IBM"),
       lty = 1, col = rainbow(ncol(zoo.cumuRtns)))
```

Plots of Cumulative Returns

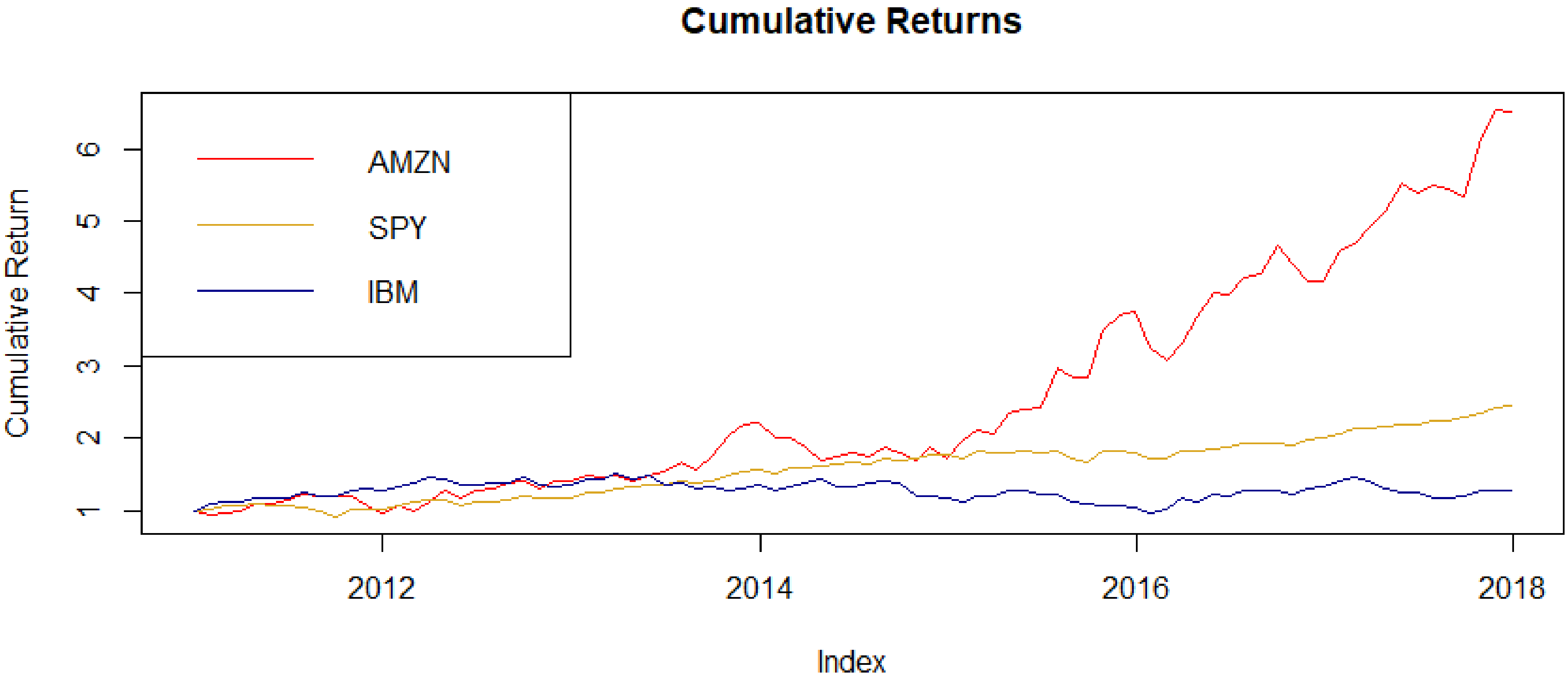
-



Plots of Cumulative Returns

```
# Alternatively, choose your own colors:
myColors <- c("red", "goldenrod", "darkblue")
plot(zoo.cumuRtns, ylab = "Cumulative Return",
     main = "Cumulative Returns", col = myColors,
     screens = 1)
# Set a legend in the upper left hand corner to match color to return series
legend(x = "topleft", legend = c("AMZN", "SPY", "IBM"),
       lty = 1, col = myColors)
```

Plots of Cumulative Returns



[END]