

## Prices

```
alt.data.AMZN <- getSymbols("AMZN", from = "2010-12-31", to = "2013-12-31", auto.assign = F)
```

'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.

```
head(alt.data.AMZN)
```

	AMZN.Open	AMZN.High	AMZN.Low	AMZN.Close	AMZN.Volume	AMZN.Adjusted
2010-12-31	181.96	182.30	179.51	180.00	3451900	180.00
2011-01-03	181.37	186.00	181.21	184.22	5331400	184.22
2011-01-04	186.15	187.70	183.78	185.01	5031800	185.01
2011-01-05	184.10	187.45	184.07	187.42	3418800	187.42
2011-01-06	186.50	187.41	185.25	185.86	3179700	185.86
2011-01-07	187.88	188.45	183.74	185.49	5221700	185.49

```
plot(alt.data.AMZN$AMZN.Adjusted)
```



```
data.missings.amzn <- alt.data.AMZN[-400:-500,]
```

```
plot(data.missings.amzn$AMZN.Close)
```



```
summary(alt.data.AMZN)
```

Index	AMZN.Open	AMZN.High	AMZN.Low
Min. :2010-12-31	Min. :161.2	Min. :163.5	Min. :160.6
1st Qu.:2011-09-29	1st Qu.:192.8	1st Qu.:195.3	1st Qu.:190.2
Median :2012-06-28	Median :226.5	Median :230.6	Median :224.6
Mean :2012-06-30	Mean :238.0	Mean :240.8	Mean :235.0
3rd Qu.:2013-04-02	3rd Qu.:266.6	3rd Qu.:269.3	3rd Qu.:263.7
Max. :2013-12-30	Max. :404.6	Max. :405.6	Max. :399.2

AMZN.Close	AMZN.Volume	AMZN.Adjusted
Min. :161.0	Min. : 984400	Min. :161.0
1st Qu.:193.3	1st Qu.: 2662775	1st Qu.:193.3
Median :227.2	Median : 3707050	Median :227.2
Mean :238.1	Mean : 4322605	Mean :238.1
3rd Qu.:266.4	3rd Qu.: 5162025	3rd Qu.:266.4
Max. :404.4	Max. :24134200	Max. :404.4

```
df_AMZN <- data.table(Date = index(alt.data.AMZN), alt.data.AMZN)
```

```
ggplot(df_AMZN, aes(Date, AMZN.Adjusted)) +  
  geom_line()
```



## Weekly Prices:

```
wk <- to.weekly(df_AMZN)
```

```
head(wk)
```

	df_AMZN.Open	df_AMZN.High	df_AMZN.Low	df_AMZN.Close	df_AMZN.Volume
2010-12-31	181.96	182.30	179.51	180.00	3451900
2011-01-07	181.37	188.45	181.21	185.49	22183400
2011-01-14	185.04	188.94	182.51	188.75	15899000
2011-01-21	188.66	191.60	176.84	177.42	20294700
2011-01-28	177.95	185.00	166.90	171.14	48516100
2011-02-04	170.16	177.19	167.41	175.93	24378200

	df_AMZN.Adjusted
2010-12-31	180.00
2011-01-07	185.49
2011-01-14	188.75
2011-01-21	177.42
2011-01-28	171.14
2011-02-04	175.93

## Monthly Prices:

```
mo <- to.monthly(df_AMZN)
```

```
head(mo)
```

	df_AMZN.Open	df_AMZN.High	df_AMZN.Low	df_AMZN.Close	df_AMZN.Volume
Dec 2010	181.96	182.30	179.51	180.00	3451900
Jan 2011	181.37	191.60	166.90	169.64	113611300
Feb 2011	170.52	191.40	169.51	173.29	95776400

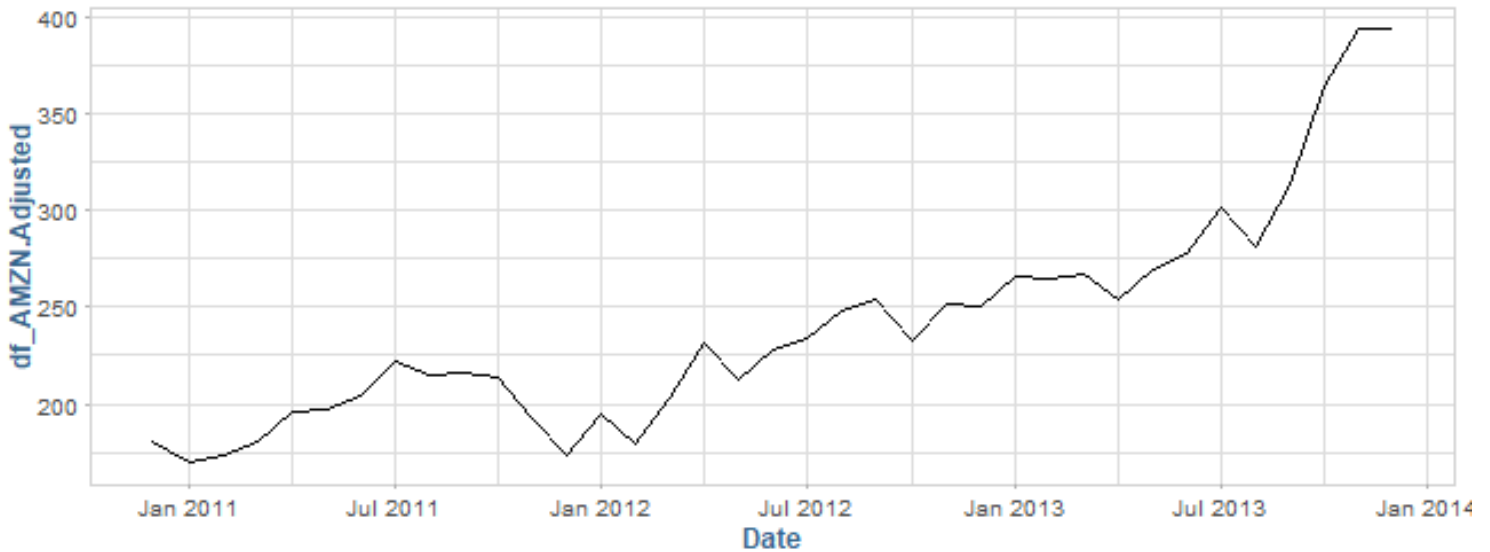
Mar 2011	173.53	181.57	160.59	180.13	118979100
Apr 2011	181.58	197.80	175.37	195.81	116749400
May 2011	196.57	206.39	190.88	196.69	106274500

df\_AMZN.Adjusted

Dec 2010	180.00
Jan 2011	169.64
Feb 2011	173.29
Mar 2011	180.13
Apr 2011	195.81
May 2011	196.69

```
df_mo <- data.table(Date = index(mo), mo)
```

```
ggplot(df_mo, aes(Date, df_AMZN.Adjusted)) +  
  geom_line()
```

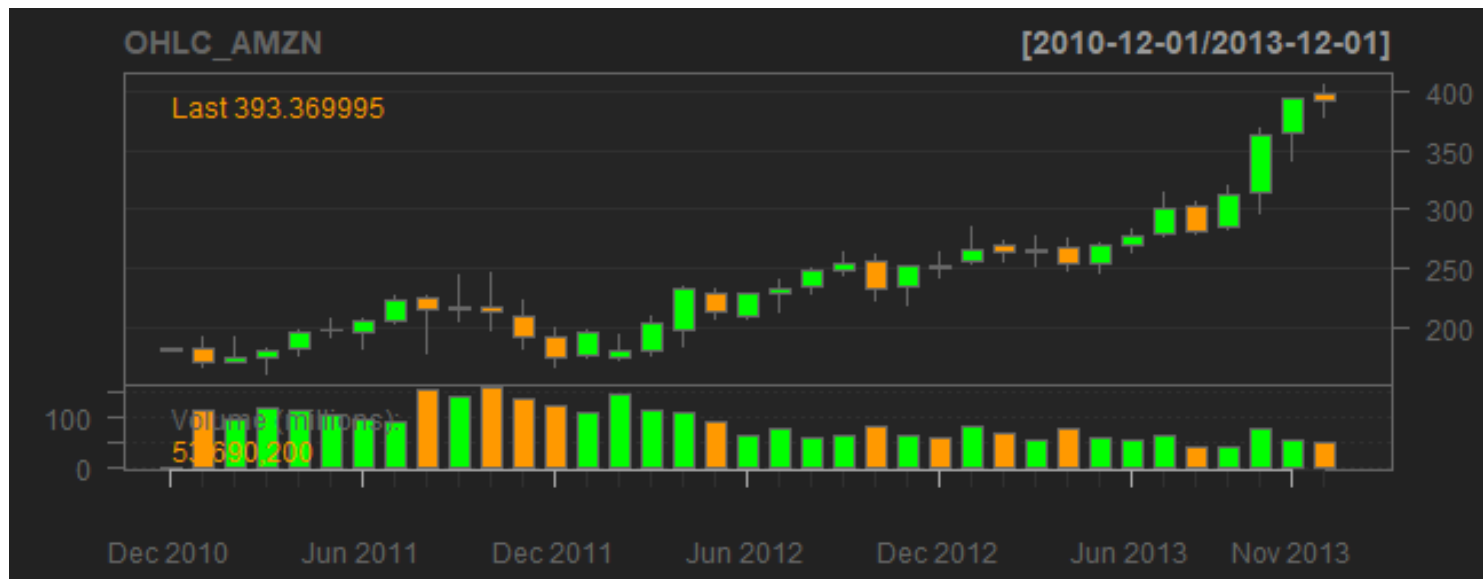


```
OHLC_AMZN <- as.quantmod.OHLC(mo)
```

```
class(OHLC_AMZN)
```

```
[1] "quantmod.OHLC" "zoo"
```

```
chartSeries(OHLC_AMZN)
```



## Comparing Multiple Securities

```
from <- "2010-12-31"; to <- "2013-12-31"

data.MSFT <- getSymbols("MSFT", from = from, to = to, auto.assign = F)
data.AMZN <- getSymbols("AMZN", from = from, to = to, auto.assign = F)
data.GSPC <- getSymbols("^GSPC", from = from, to = to, auto.assign = F)
data.IBM <- getSymbols("IBM", from = from, to = to, auto.assign = F)

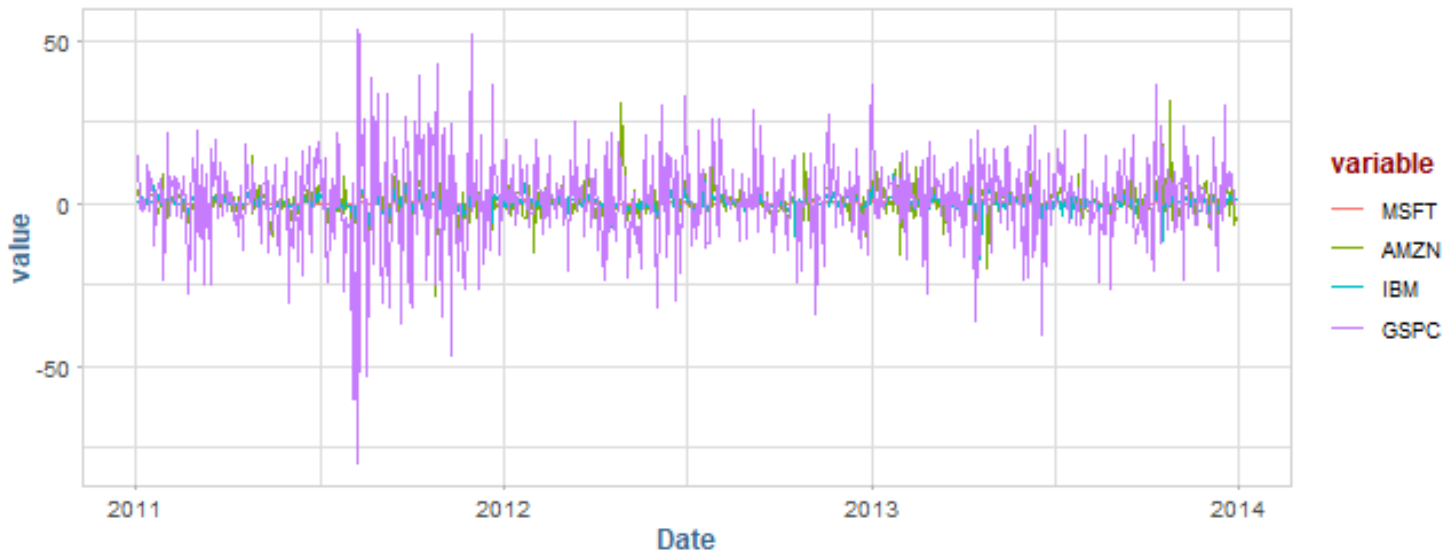
combined <- data.table(Date = index(data.MSFT),
                        MSFT = data.MSFT$MSFT.Close,
                        AMZN = data.AMZN$AMZN.Close,
                        IBM = data.IBM$IBM.Close,
                        GSPC = data.GSPC$GSPC.Close)

colnames(combined) <- c("Date", "MSFT", "AMZN", "IBM", "GSPC")

combined.returns <- data.table(Date = combined$Date[-1], apply(combined[, 2:5], 2, diff))

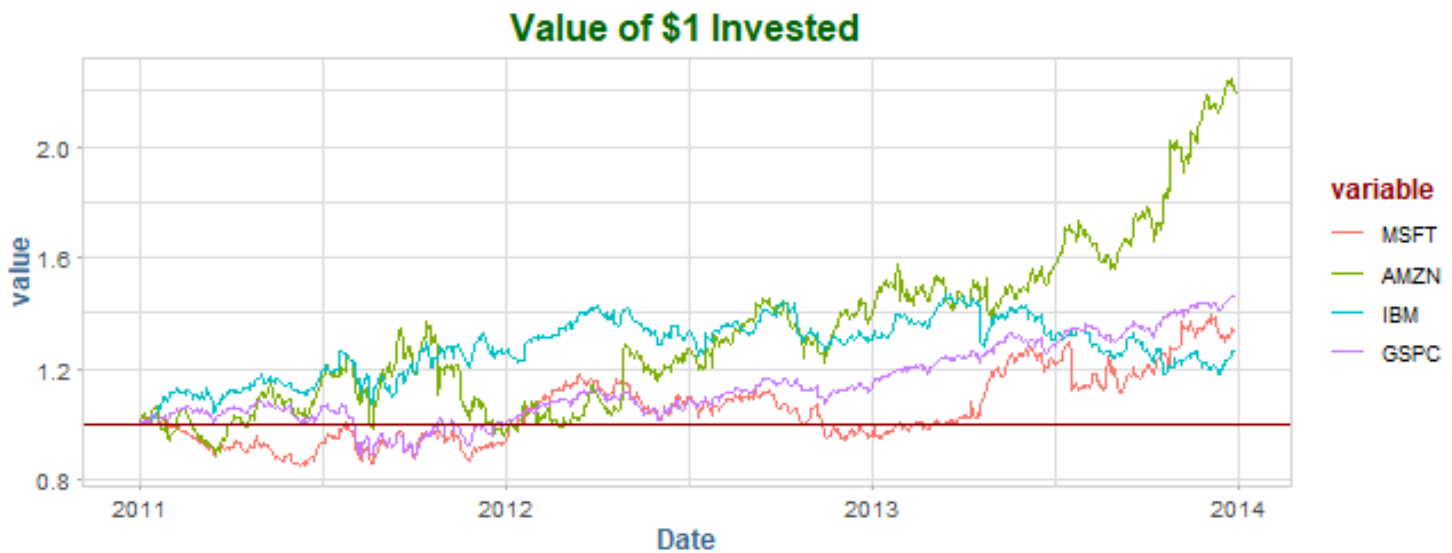
data.returns <- melt(combined.returns, id.vars = "Date")

ggplot(data.returns, aes(Date, value, group = variable)) +
  geom_line(aes(col = variable))
```

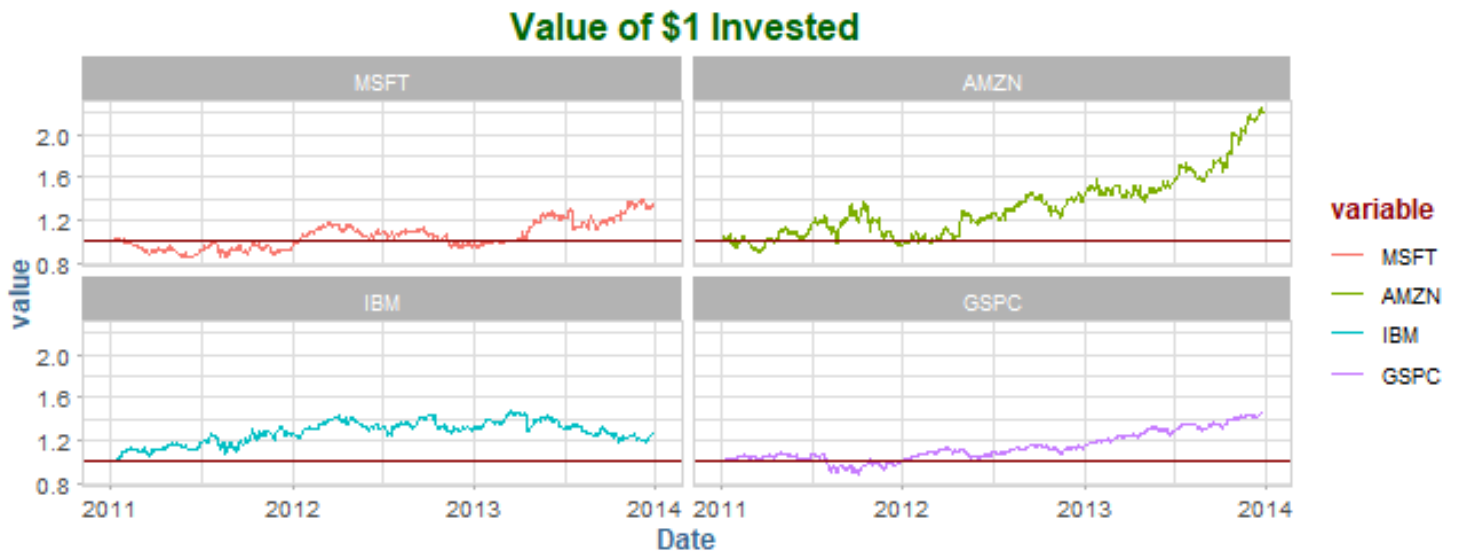


```
combined.appreciation <- data.table(Date = combined$Date, apply(combined[, 2:5], 2, function(x) {
  data.appreciation <- melt(combined.appreciation, id.vars = "Date")

  ggplot(data.appreciation, aes(Date, value, group = variable)) +
    geom_line(aes(col = variable)) +
    geom_hline(yintercept = 1, col = "darkred") +
    labs(title = "Value of $1 Invested")
}))
```



```
ggplot(data.appreciation, aes(Date, value)) +
  geom_line(aes(col = variable)) +
  geom_hline(yintercept = 1, col = "darkred") +
  facet_wrap(~variable) +
  labs(title = "Value of $1 Invested")
```



```
gg_color_hue <- function(n) {
  hues = seq(15, 375, length = n + 1)
  hcl(h = hues, l = 65, c = 100)[1:n]
}

names <- unique(data.appreciation$variable)

n <- length(names)

colors <- gg_color_hue(n)

plots <- gList()

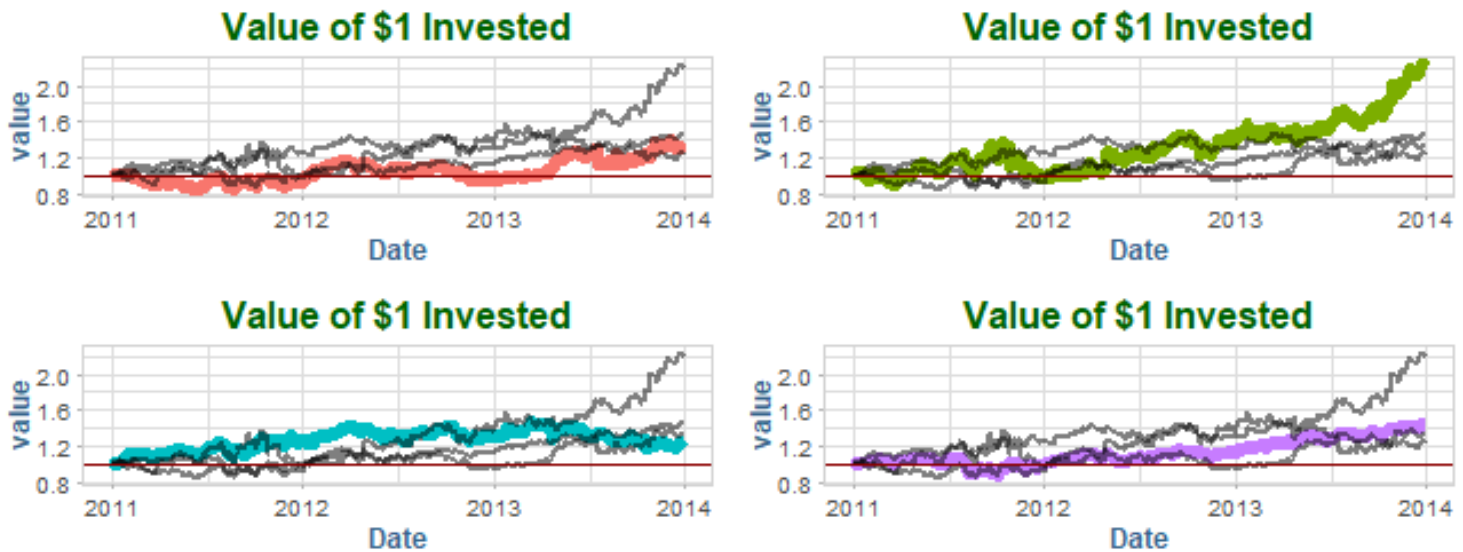
for(index in 1:length(names))
{
  current <- names[index]

  plot <- local({
    ggplot() +
      geom_line(data = data.appreciation[variable == current], aes(Date, value), col = colors[index]) +
      geom_line(data = data.appreciation[variable != current], aes(Date, value, group = variable)) +
      geom_hline(yintercept = 1, col = "darkred") +
      labs(title = "Value of $1 Invested")

  })

  plots[[index]] <- plot
}
```

```
do.call("grid.arrange", c(plots, ncol=2))
```



## Technical Analysis

### Simple Moving Average

```
plot_sma <- function(data, name) {
  prices <- data.table(Date = data$Date, Close = data[, get(name)])

  prices$sma50 <- rollmeanr(prices$Close, k=50, fill = NA)
  prices$sma200 <- rollmeanr(prices$Close, k=200, fill = NA)

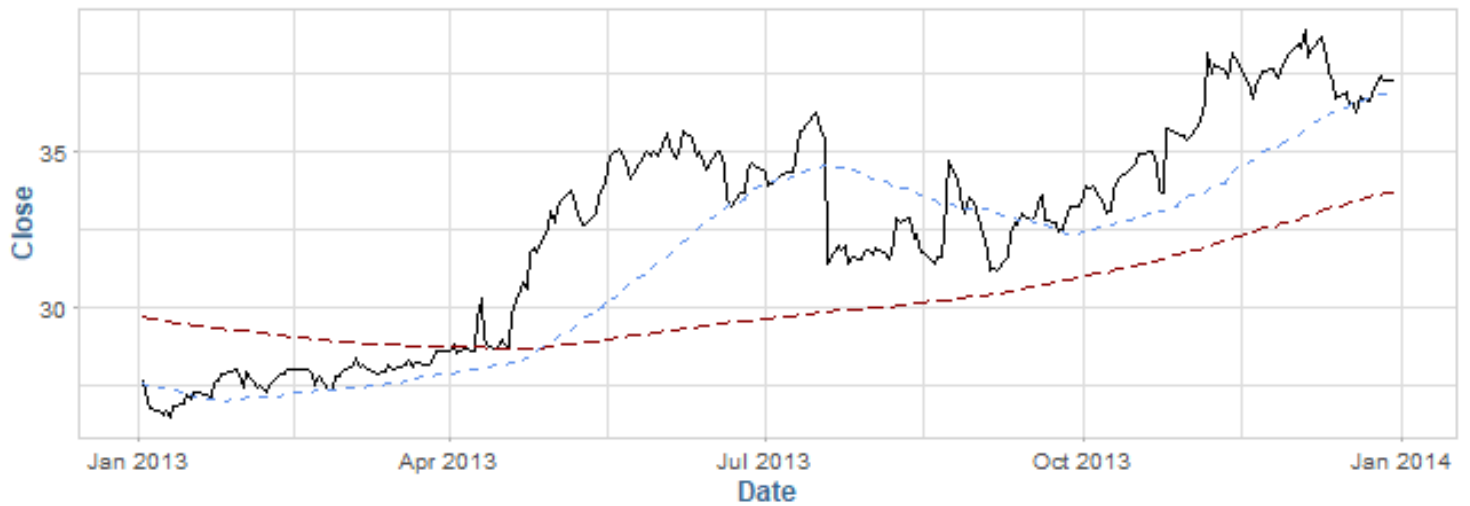
  ggplot(prices[Date >= "2013-1-1"]) +
    geom_line(aes(Date, Close)) +
    geom_line(aes(Date, sma50), col = "cornflowerblue", lty = 2) +
    geom_line(aes(Date, sma200), col = "darkred", lty = 5) +
    labs(title = paste(name, "SMA"))
}

plot_sma(combined, "AMZN")
```



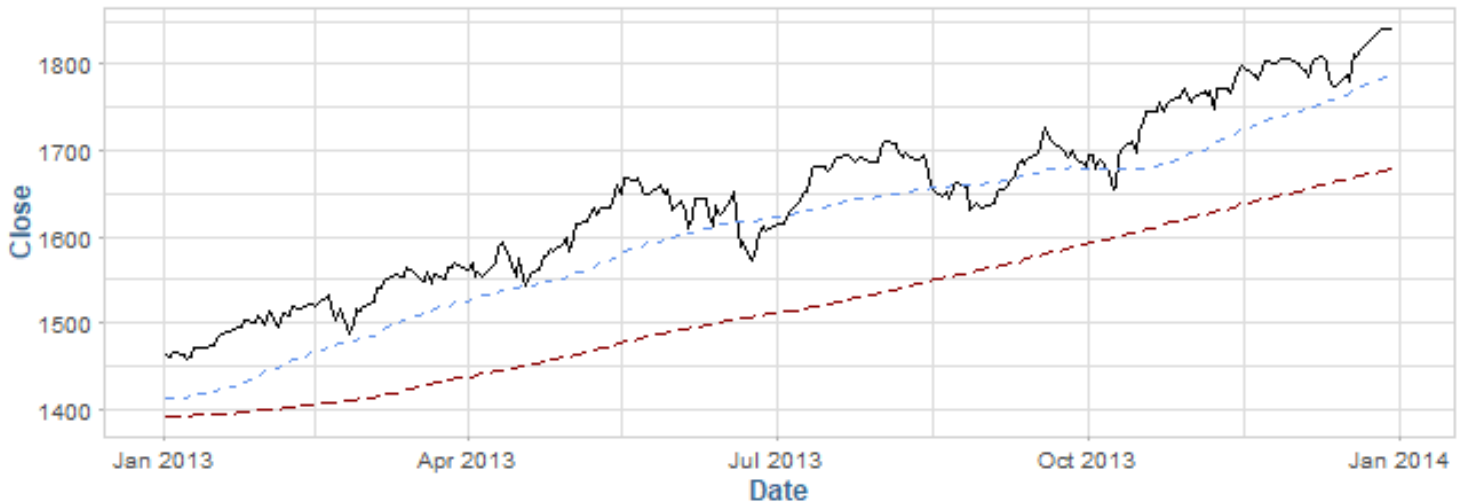
**AMZN SMA**

```
plot_sma(combined, "MSFT")
```

**MSFT SMA**

```
plot_sma(combined, "GSPC")
```

### GSPC SMA



```
plot_sma(combined, "IBM")
```

### IBM SMA



## Bollinger Bands

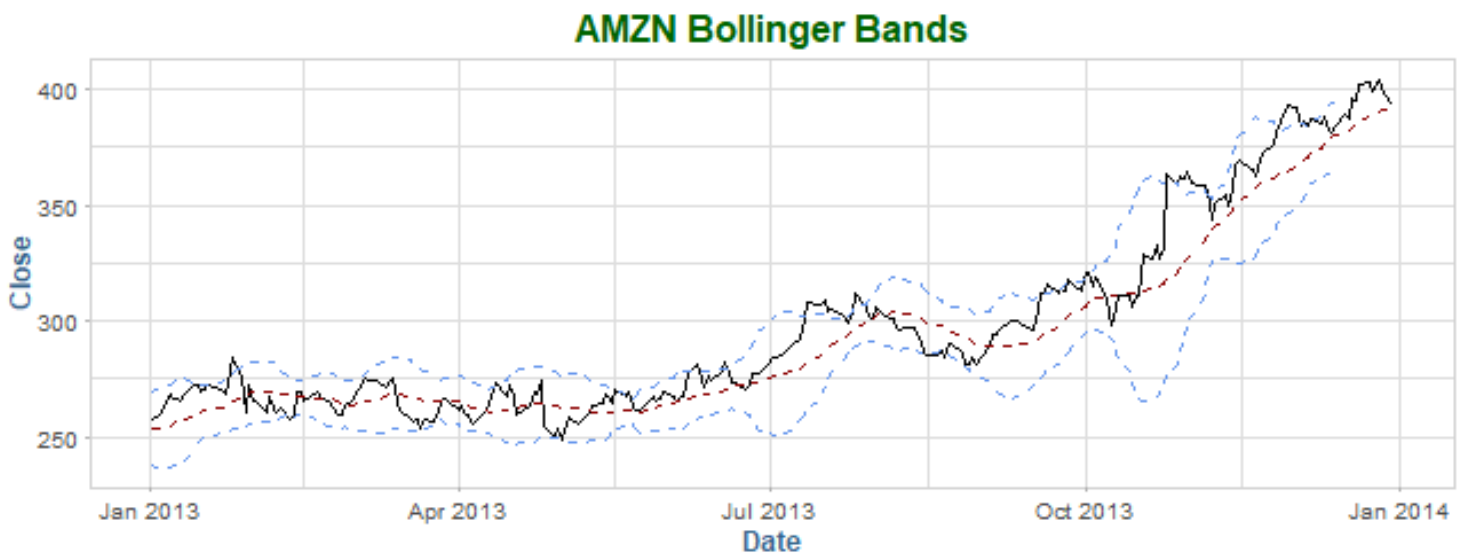
```
plot_bb <- function(data, name) {
  prices <- data.table(Date = data$Date, Close = data[, get(name)])
  prices$avg <- rollmeanr(prices$Close, k=20, fill = NA)
  prices$sd <- rollapply(prices$Close, width = 20, FUN = sd, fill = NA)
  prices$upper <- prices$avg + 2*prices$sd
  prices$lower <- prices$avg - 2*prices$sd

  ggplot(prices[Date >= "2013-1-1"]) +
```

```
geom_line(aes(Date, Close)) +  
geom_line(aes(Date, avg), col = "darkred", lty = 2) +  
geom_line(aes(Date, upper), col = "cornflowerblue", lty = 2) +  
geom_line(aes(Date, lower), col = "cornflowerblue", lty = 2) +  
labs(title = paste(name, "Bollinger Bands"))  
}  
  
plot_bb(combined, "AMZN")
```

Warning: Removed 10 row(s) containing missing values (geom\_path).

Warning: Removed 10 row(s) containing missing values (geom\_path).

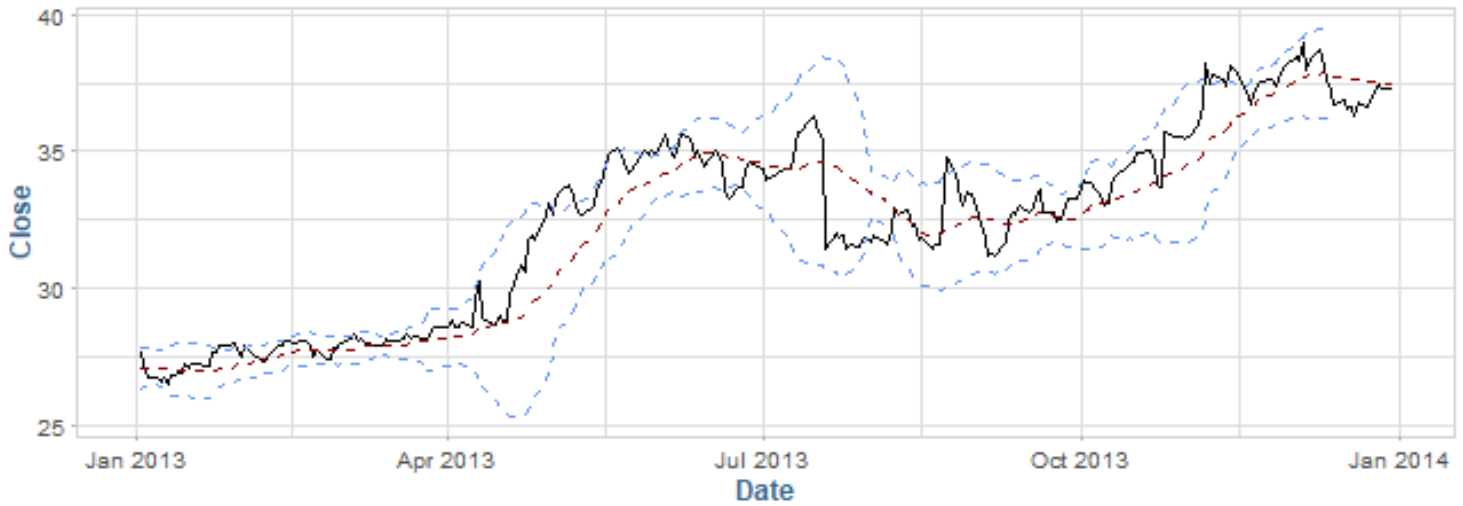


```
plot_bb(combined, "MSFT")
```

Warning: Removed 10 row(s) containing missing values (geom\_path).

Warning: Removed 10 row(s) containing missing values (geom\_path).

### MSFT Bollinger Bands



```
plot_bb(combined, "IBM")
```

Warning: Removed 10 row(s) containing missing values (geom\_path).

Warning: Removed 10 row(s) containing missing values (geom\_path).

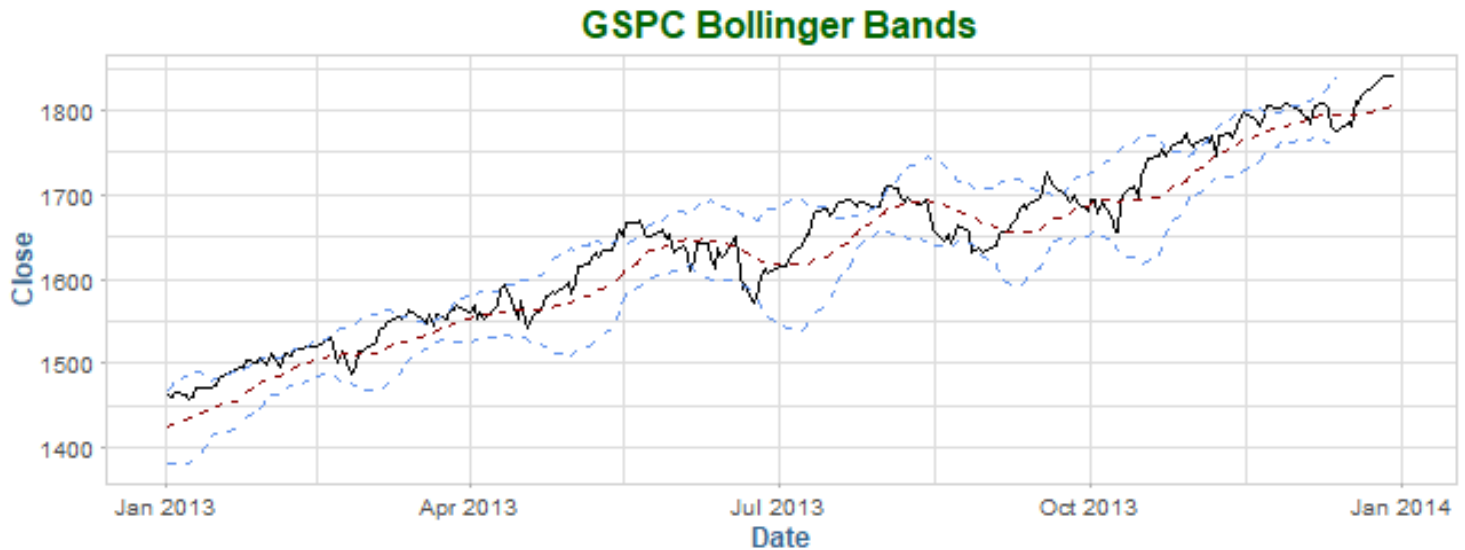
### IBM Bollinger Bands



```
plot_bb(combined, "GSPC")
```

Warning: Removed 10 row(s) containing missing values (geom\_path).

Warning: Removed 10 row(s) containing missing values (geom\_path).



## Momentum

### Relative Strength Index

```
plot_rsi <- function(data, name) {
  prices <- data.table(Date = data$Date, Close = data[, get(name)])
  prices$delta <- c(NA, diff(prices$Close))
  prices$up <- ifelse(prices$delta > 0, 1, 0)
  prices$down <- ifelse(prices$delta < 0, -1, 0)
  prices$up.val <- prices$up * prices$delta
  prices$down.val <- prices$down * prices$delta
  prices$up.first.avg <- rollmeanr(prices$up.val, k = 14, fill = NA)
  prices$down.first.avg <- rollmeanr(prices$down.val, k = 14, fill = NA)

  up.val <- as.numeric(prices$up.first.avg)
  down.val <- as.numeric(prices$down.first.avg)

  prices$up.avg <- prices$up.first.avg
  for(i in 15:nrow(prices)) {
    prices$up.avg[i] <- ((up.val[i-1]*13 + up.val[i])/15)
  }

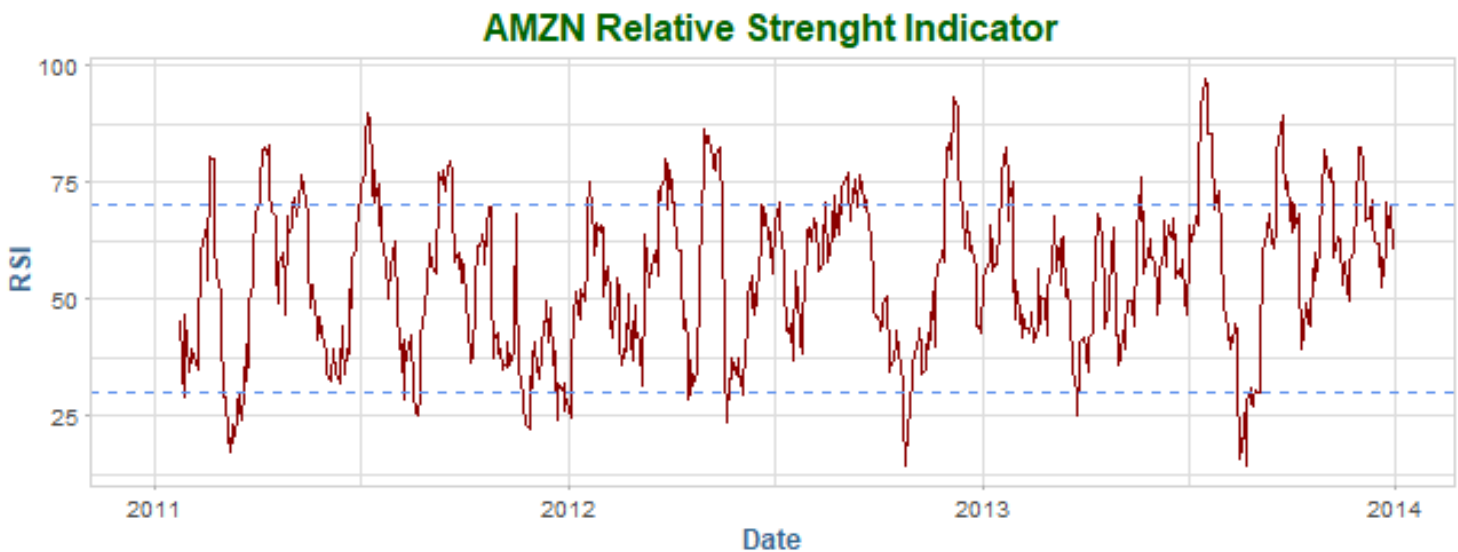
  prices$down.avg <- prices$down.first.avg
  for(i in 15:nrow(prices)) {
    prices$down.avg[i] <- ((down.val[i-1]*13 + down.val[i])/15)
  }

  prices$RS <- prices$up.avg / prices$down.avg
  prices$RSI <- 100 - (100/(1 + prices$RS))
}
```

```
ggplot(prices, aes(Date, RSI)) +  
  geom_line(col = "darkred") +  
  geom_hline(yintercept = 30, col = "cornflowerblue", lty = 2) +  
  geom_hline(yintercept = 70, col = "cornflowerblue", lty = 2) +  
  labs(title = paste(name, "Relative Strenght Indicator"))  
}
```

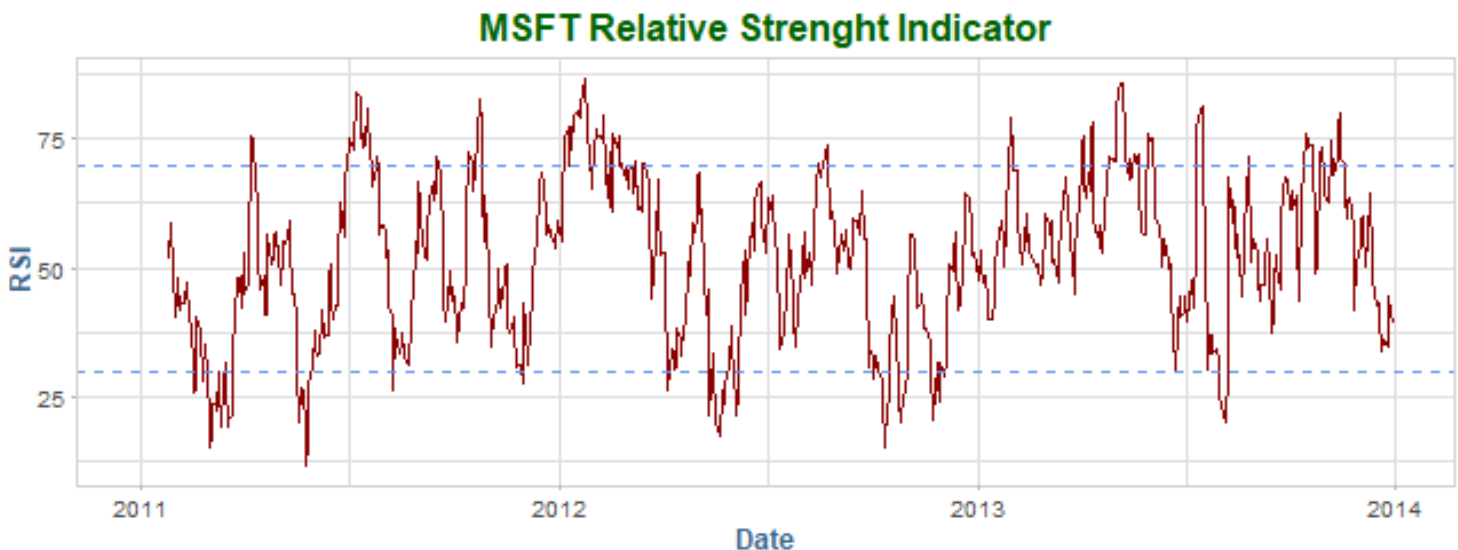
```
plot_rsi(combined, "AMZN")
```

Warning: Removed 15 row(s) containing missing values (geom\_path).



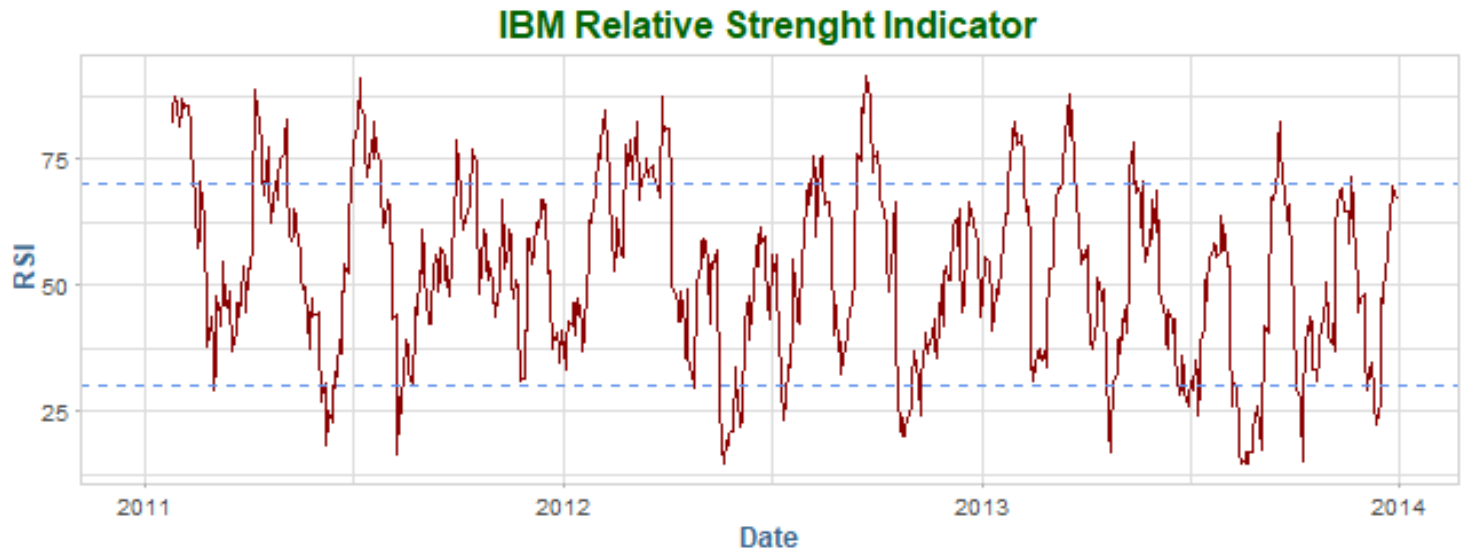
```
plot_rsi(combined, "MSFT")
```

Warning: Removed 15 row(s) containing missing values (geom\_path).



```
plot_rsi(combined, "IBM")
```

Warning: Removed 15 row(s) containing missing values (geom\_path).



```
plot_rsi(combined, "GSPC")
```

Warning: Removed 15 row(s) containing missing values (geom\_path).

