Exercise

Alone, in pairs, or in groups, discuss in what other ways graphics can deceive, either intentionally or not.

I argue that misinterpretation is the blame for deceit with graphics. This misinterpretation is a function of how the graph is presented and can be worsened by a presenter making claims that are not distinguished by the graphic.

One typical example of graphical deceit is stock price comparison over time. On the y-axis is the price of a stock, on the x-axis is the timeline, and there are multiple lines of different color indicating multiple companies. The deceit appears when trying to use such a graph to talk about comparing a return on investment between companies. This is wrong to do because the y-axis represents a magnitude measurement, not a rate measurement. Three graphics are presented below to show how a graph of stock prices can be adjusted to do a fair comparison on rates of return over time.

Figure 1 below show the price of nine stocks at the end of the day Friday (4:30 PM EST) every two weeks from October 12th, 2012 to October 12th, 2018. This figure has the same structure as explained above to show the same deceit, which is the difference in magnitudes on the y-axis suppressing the trends of smaller priced stocks to look flat.

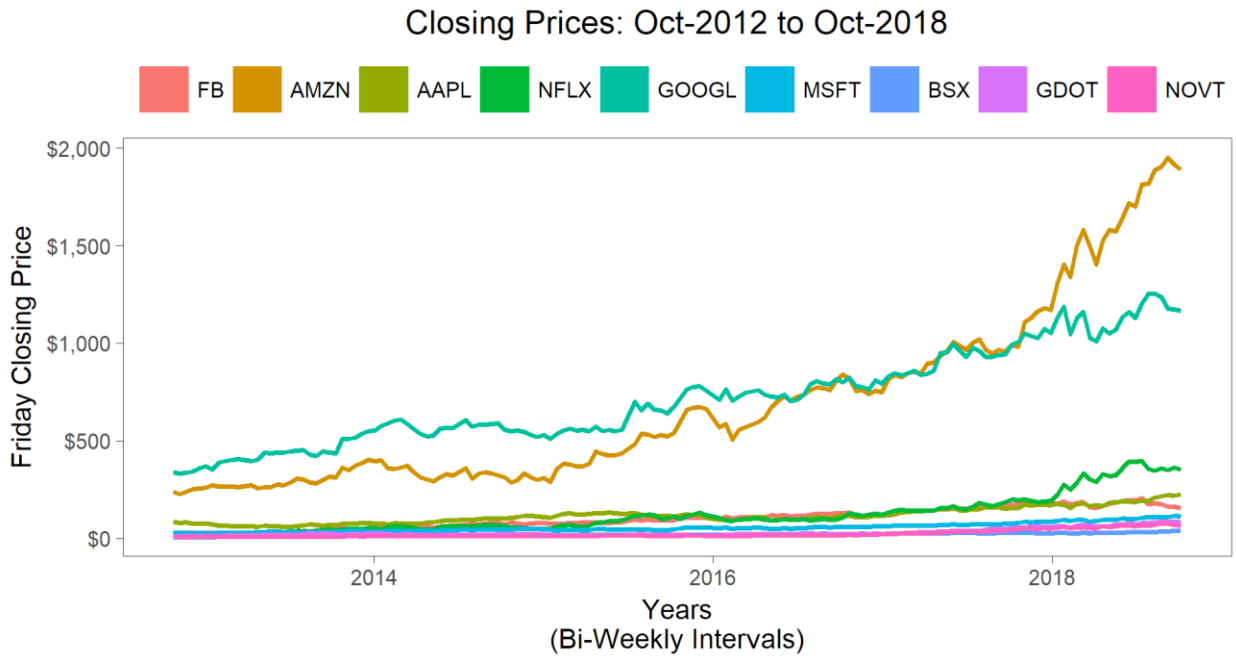

Figure 1: Six Years of Bi-Weekly Closing Prices for Nine Stocks

Two logarithmic properties allow for a convenient way to convert prices into returns. Before the properties are presented, some mathematical notation is required.

$$r_t: return\ at\ time\ t \qquad p_t: price\ at\ time\ t$$

The first property shows the logarithmic relationship between return and price over an investment length of k discrete time periods.

$$log(1 + r_t) = log\left(\frac{p_t}{p_{t-k}}\right) = log(p_t) - log(p_{t-k})$$

We can solve for the rate of return like so.

$$exp(log(1 + r_t)) - 1 = r_t$$

The second property shows that the summation of a sequence of log differences converges to a single log difference for the entire set of sequences. Let N represent the total number of time periods to evaluate.

$$\sum_{t=1}^{N} log(1 + r_t) = log(1 + r_1) + log(1 + r_2) + \cdots + log(1 + r_N) = log(p_N) - log(p_1)$$

The first property allows us to create log differences between successive bi-weekly Friday closing prices. The second property allows us to compute a set of summations that start from the first Friday up to each following Friday until the last Friday.

Figure 2 below shows the growing rate of return for nine stocks. The y-axis now has a measurement on it suitable for comparing rates of return between companies regardless of their stock price. There is still deceit, and this is shown by the extraordinary performance of Netflix (NFLX) in green. The graph shows Netflix has outperformed other stocks by orders of magnitude, which flattens the trends of the other stocks with lower return rates. This calls for the removal of Netflix from the comparison, because it is an anomaly.

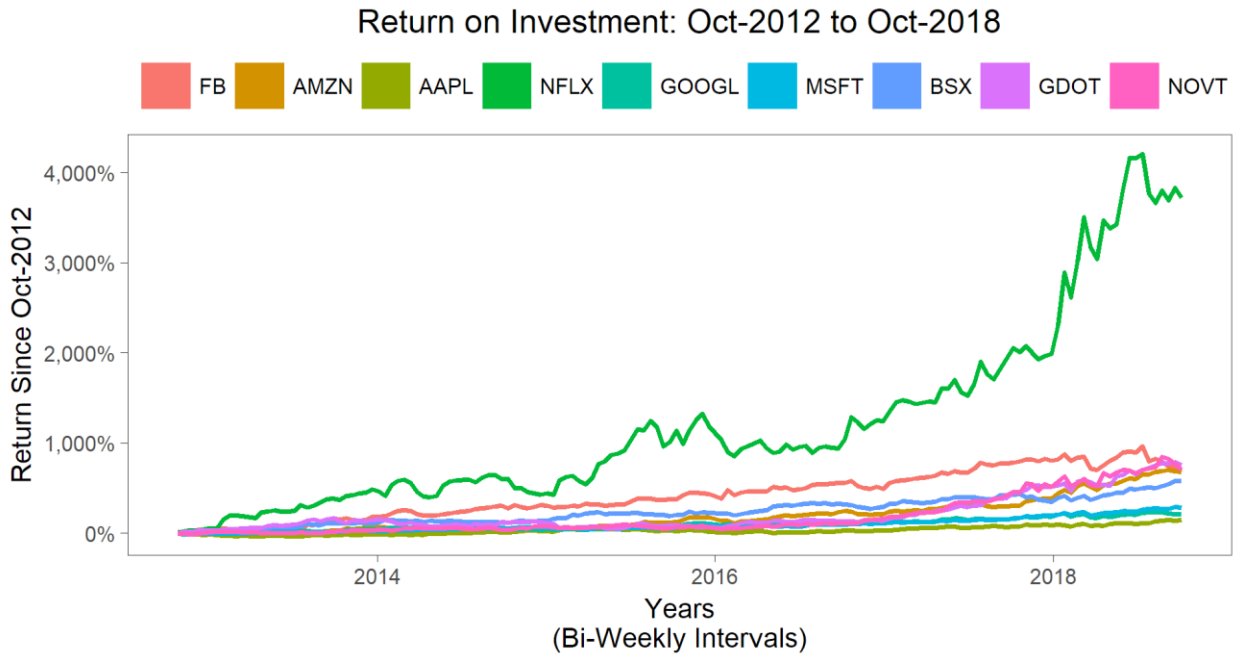

Figure 2: Rate of Return for Nine Stocks over Six Years

Figure 3 below shows the comparison of eight stocks that all vary in price per share as of October 22, 2018. This graph shows that lower priced stocks BSX ($36), GDOT ($76), and NOVT ($65) outperform AAPL ($220), GOOGL ($1112), and MSFT ($109) while performing like AMZN ($1792), and catching up to FB ($154).

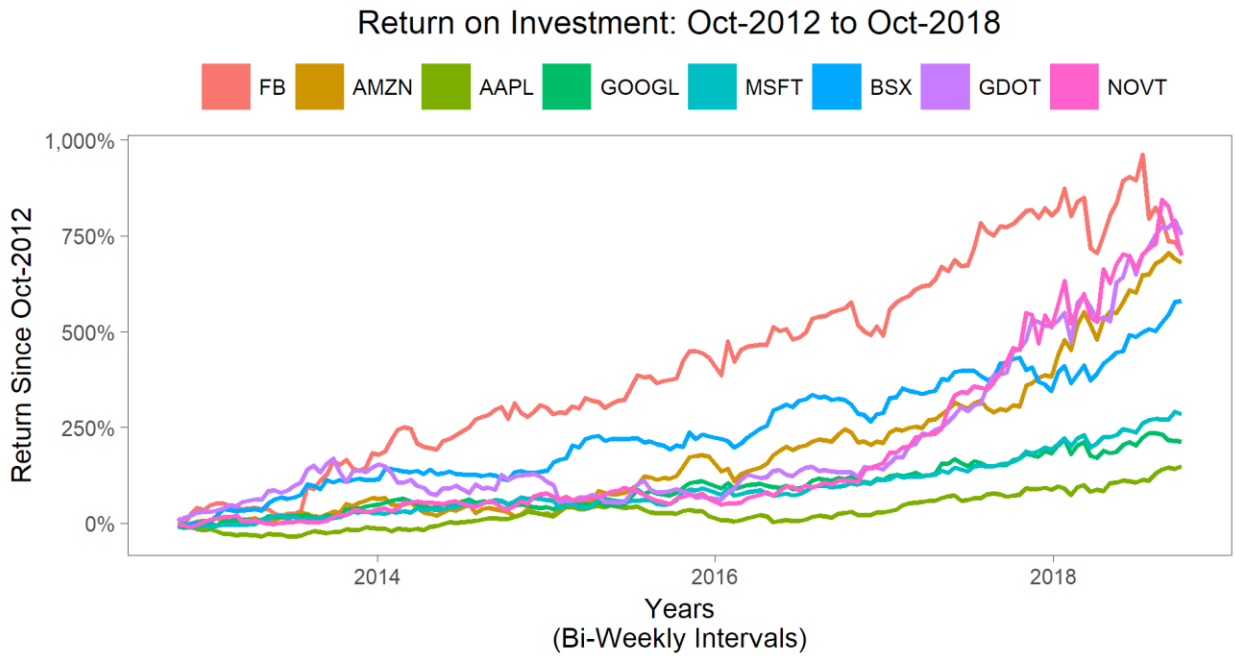

Figure 3: Rate of Return for Eight Stocks over Six Years

The program for this analysis is supplied on the following page for reproducibility of the approach. The program was written in the open source statistical computer language, R.

Program

```
# you will need to install these packages just once on your computer
# install.packages(“ggplot2”)
# install.packages(“scales”)
# install.packages(“data.table”)
# install.packages(“quantmod”)

# load these packages for the script to work
require(ggplot2)
require(scales)
require(data.table)
require(quantmod)

# disable the yahoo warning message
options("getSymbols.yahoo.warning" = FALSE)
options("getSymbols.warning4.0" = FALSE)

# only keep closing prices at the end of the week, every 2 weeks, for the past 4 years
keep.dates = seq.Date(from = as.Date("2012-10-12"), to = as.Date("2018-10-12"), by = "2 weeks")

# get stock prices for a set of companies
stock.prices = lapply(c("FB", "AMZN", "AAPL", "NFLX", "GOOGL", "MSFT", "BSX", "GDOT", "NOVT"),
function(s)
{
  # get the closing prices for symbol s
  DT = data.frame(Cl(getSymbols(s, src = "yahoo", auto.assign = FALSE)))

  # make the row names into a Date column for DT
  DT = cbind("Date" = as.Date(row.names(DT)), data.table(DT))

  # rename the columns of DT
  setnames(DT, c("Date", "Close"))

  # make a Symbol column for DT
  DT = cbind("Symbol" = s, DT)

  # only keep the dates of interest
  DT = DT[Date %in% keep.dates]

  # compute the difference between log prices
  DT[, diffLog := c(NA, diff(log(Close)))]

  # remove missing values from DT
  DT = na.omit(DT)
```

```
  # compute a rolling sum of diffLog at each date of interest
  rolling.sum = sapply(1:nrow(DT), function(i) sum(DT$diffLog[1:i]))

  # convert the rolling sum into percent return values
  rolling.return = exp(rolling.sum) - 1

  # add return values to DT
  DT[, Return := rolling.return]

  # return DT without missing values
  return(na.omit(DT))
})

# combine the list of tables into one table
stock.prices = rbindlist(stock.prices)

# plot Return
return.plot = ggplot(data = stock.prices[Symbol != "NFLX"], aes(x = Date, y = Return, color = Symbol)) +
  # geom_point() +
  geom_line(size = 2) +
  scale_y_continuous(label = percent) +
  ggtitle("Return on Investment: Oct-2012 to Oct-2018") +
  labs(x = "Years\n(Bi-Weekly Intervals)", y = "Return Since Oct-2012", color = "") +
  theme_bw(25) +
  theme(legend.position = "top",
        legend.key.size = unit(0.25, "in"),
        plot.title = element_text(hjust = 0.5),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank()) +
  guides(color = guide_legend(override.aes = list(size = 20, linetype = 1, alpha = 1), nrow = 1))

return.plot
```