Coka-Cola price in past 2 years report

Ziyu Huang

2025-06-20

Introduction

In this report, we analyze the closing prices of Coca-Cola(KO) over the past two years. Our goal is to fit a smooth curve through the observed prices. We will use polynomial regression and Newton-Raphson's method to locate key turning points.

Data Retrieval and Preprocessing

To begin, we collect Coca-Cola's daily closing stock prices from the past two years using publicly available data from Yahoo Finance. These daily prices give us a detailed picture of how the stock moved over time. Before analyzing patterns, we smooth out some of the day-to-day fluctuations using a moving average technique to better observe longer-term trends.

1. Load required packages and set working directory

2. Fetch Coca-Cola closing price data from Yahoo Finance and plot it. To get the trading days data of the past two years, we ask the function to count back 730 days.

```
par(mfrow = c(1,1))

getSymbols('KO', src= 'yahoo', from=Sys.Date()-730, auto.assign=T)
```

```
## [1] "KO"
```

```
ko = Cl(KO)
Y = as.numeric(ko)
n = length(Y)
```

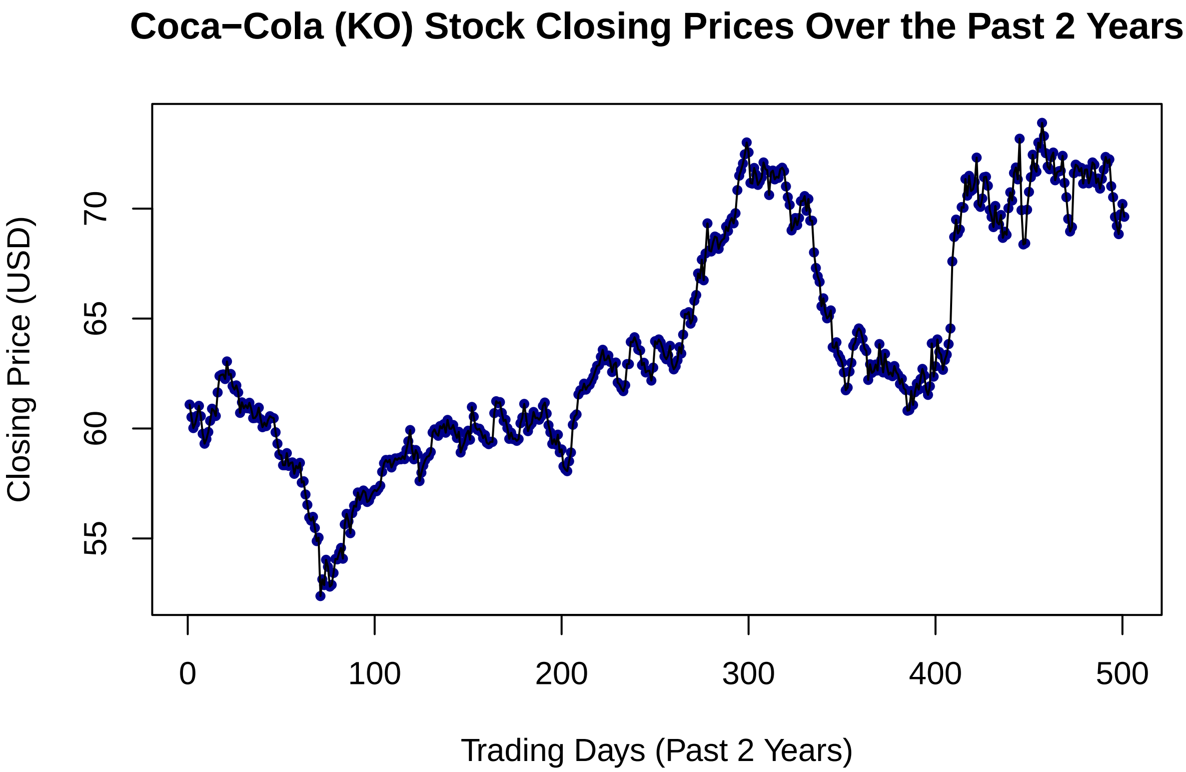
3. Load and apply custom moving average smoothing function(my_tsma)

```
source('my_tsma.R')
M = my_tsma(Y, 14)
dates = index(KO)
ydf = data.frame(Date = dates, Close = Y, MA = M$approx)
```

4. Plot raw data again with smoothed version

```
source('mlr.R')
tt = seq(1, n, 1)
plot(
  tt, Y,
```

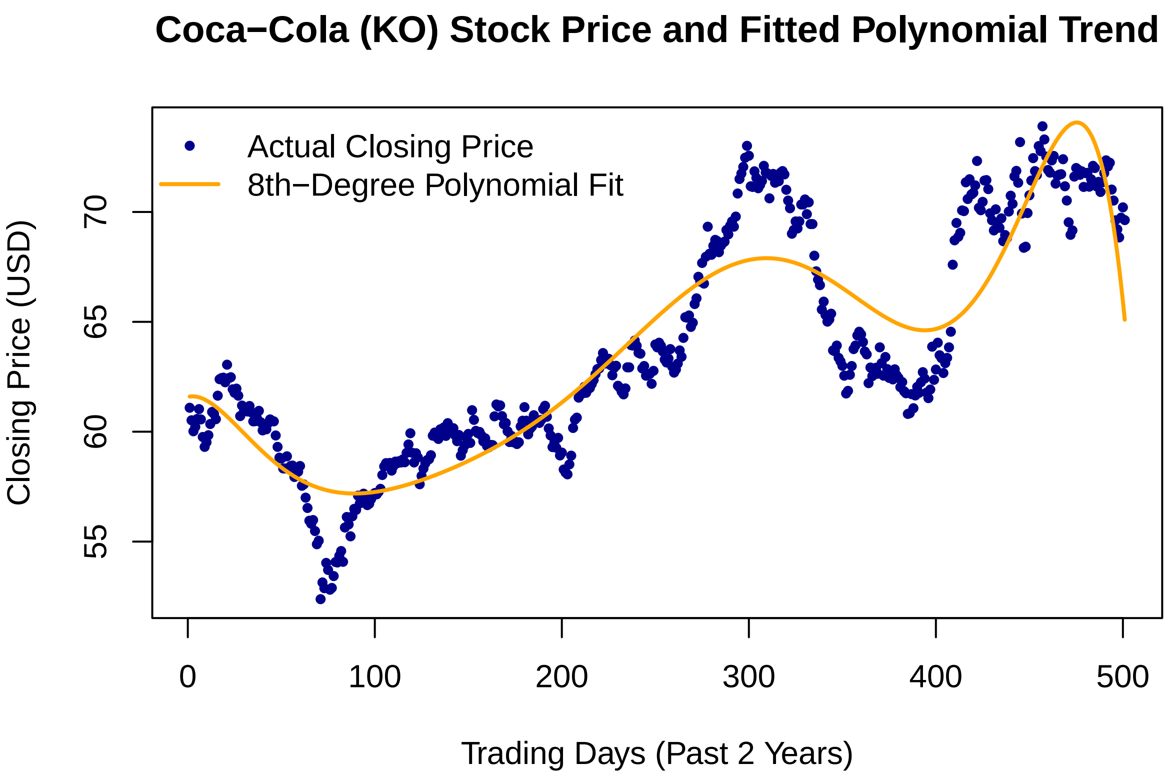
```r
  pch = 16,
  col = 'darkblue',
  cex = 0.7,
  xlab = "Trading Days (Past 2 Years)",
  ylab = "Closing Price (USD)",
  main = "Coca-Cola (KO) Stock Closing Prices Over the Past 2 Years"
)
lines(tt, Y)
```



Polynomial Regression Fit 8th-degree polynomial model, since it is the best fit after trying out different orders of polynomials

```r
ttz = (tt - mean(tt)) / sd(tt)

Mp = mlr(cbind(1, ttz, ttz^2, ttz^3, ttz^4, ttz^5, ttz^6, ttz^7, ttz^8), Y)
bhat = Mp$bhat
yhat = Mp$Yhat

plot(
  tt, Y, pch = 16, col = 'darkblue',cex = 0.7,
  xlab = "Trading Days (Past 2 Years)",
  ylab = "Closing Price (USD)",
  main = "Coca-Cola (KO) Stock Price and Fitted Polynomial Trend"
)

lines(tt, yhat, col = 'orange', lwd = 2)

legend("topleft",
       legend = c("Actual Closing Price", "8th-Degree Polynomial Fit"),
```

```
       col = c("darkblue", "orange"),
       pch = c(16, NA),
       lty = c(NA, 1),
       lwd = c(NA, 2),
       pt.cex = c(0.7, NA),
       bty = "n")
```



Derivative and Turning Point Detection

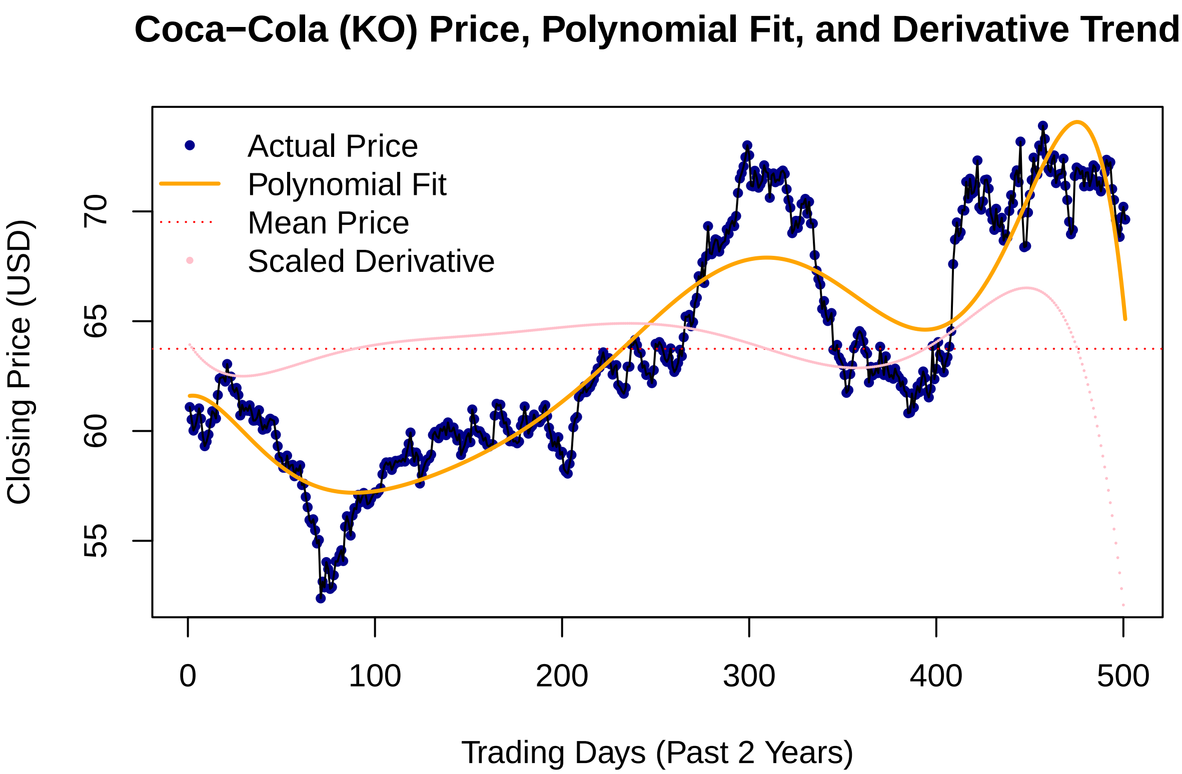
Compute derivative coefficients and define function. This section allows us to see the trend of how the slope changes in the original graph.

```
dbhat = bhat[2:9] * 1:8

# Define the derivative of the fitted polynomial
dpx = function(x){
  x = (x - mean(tt)) / sd(tt)
  val = dbhat[1] + dbhat[2]*x + dbhat[3]*x^2 + dbhat[4]*x^3 +
        dbhat[5]*x^4 + dbhat[6]*x^5 + dbhat[7]*x^6 +dbhat[8]*x^7
  return(val)
}
```

Plot the original data, fitted curve, and scaled derivative to observe the pattern in the graph

```
tt = seq(1, n, 1)
plot(
  tt, Y,
  pch = 16,
```

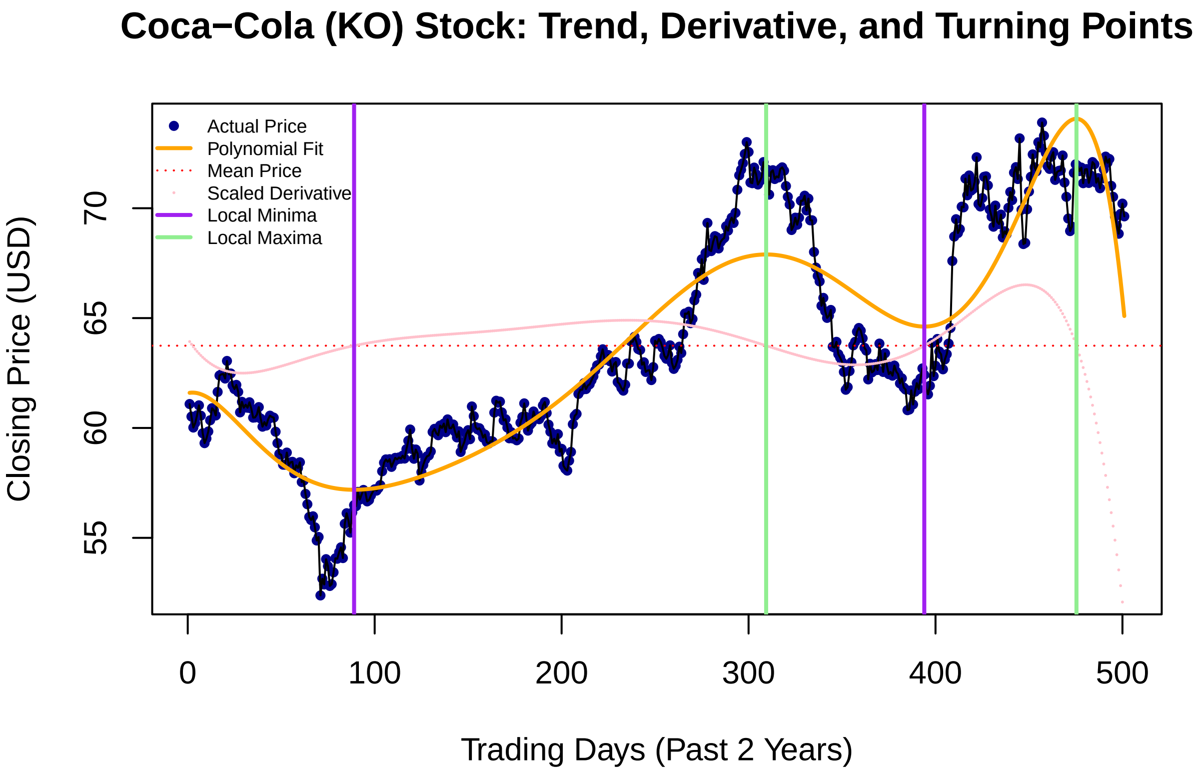
```
  col = 'darkblue',
  cex = 0.7,
  xlab = "Trading Days (Past 2 Years)",
  ylab = "Closing Price (USD)",
  main = "Coca-Cola (KO) Price, Polynomial Fit, and Derivative Trend"
)
lines(tt, Y)
lines(tt, yhat, col = 'orange', lwd = 2)
abline(h = mean(Y), lty = 3, col = 'red')

# Overlay scaled derivative
points(tt, dpx(tt) * 0.1 + mean(Y), pch = 16, col = 'pink', cex = 0.2)

# Add legend
legend("topleft",
       legend = c("Actual Price", "Polynomial Fit", "Mean Price", "Scaled Derivative"),
       col = c("darkblue", "orange", "red", "pink"),
       pch = c(16, NA, NA, 16),
       lty = c(NA, 1, 3, NA),
       lwd = c(NA, 2, 1, NA),
       pt.cex = c(0.7, NA, NA, 0.5),
       bty = "n")
```



Using Newton-Raphson method to find the minimum and maximum. From the pink derivative line we derived above, we observed that there are 5 intersections with the mean, and we suspect there is a turning point at the stock price curve. To avoid missing any possible maxima, we choose to have 6 guesses for Newton-Raphson method.

```r
# Plot the data and fitted curve with enhancements
tt = seq(1, n, 1)
plot(
  tt, Y,
  pch = 16,
  col = 'darkblue',
  cex = 0.7,
  xlab = "Trading Days (Past 2 Years)",
  ylab = "Closing Price (USD)",
  main = "Coca-Cola (KO) Stock: Trend, Derivative, and Turning Points"
)
lines(tt, Y)
lines(tt, yhat, col = 'orange', lwd = 2)
abline(h = mean(Y), lty = 3, col = 'red')

# Overlay scaled derivative
points(tt, dpx(tt) * 0.1 + mean(Y), pch = 16, col = 'pink', cex = 0.2)

source('newrap.R')

# Define second derivative
ddpx = function(x){
  x_std = (x - mean(tt)) / sd(tt)
  val = dbhat[2] + 2*dbhat[3]*x_std + 3*dbhat[4]*x_std^2 +
        4*dbhat[5]*x_std^3 + 5*dbhat[6]*x_std^4 + 6*dbhat[7]*x_std^5 + 7*dbhat[8]*x_std^6
  return(val / sd(tt)^2)
}

# Find turning points
guesses = c(80, 130, 200, 300, 380, 450)
roots = c()

for (g in guesses) {
  result = newrap(dpx, g)
  if (result$converged) {
    roots = c(roots, round(result$solution, 1))
  }
}

roots = sort(unique(roots))

# draw turning points
for (r in roots) {
  if (ddpx(r) > 0) {
    abline(v = r, col = 'purple', lwd = 2)      # local min
  } else if (ddpx(r) < 0) {
    abline(v = r, col = 'lightgreen', lwd = 2)  # local max
  }
}

# legend
legend("topleft",
       legend = c("Actual Price", "Polynomial Fit", "Mean Price", "Scaled Derivative",
```

```
                 "Local Minima", "Local Maxima"),
       col = c("darkblue", "orange", "red", "pink", "purple", "lightgreen"),
       pch = c(16, NA, NA, 16, NA, NA),
       lty = c(NA, 1, 3, NA, 1, 1),
       lwd = c(NA, 2, 1, NA, 2, 2),
       pt.cex = c(0.7, NA, NA, 0.2, NA, NA),
       cex = 0.58,
       bty = "n")
```



Conclusion

In this paper, we analyzed the closing prices of Coca-Cola (KO) over the past two years using polynomial regression and Newton-Raphson's method. By fitting an 8th-degree polynomial to the stock price data, we were able to create a smooth curve that captures the general trends and fluctuations in price. We then computed the first derivative of the fitted model and used the Newton-Raphson method to locate where the derivative is zero.

From our analysis, we identified six turning points (local maxima and minima) across approximately 730 trading days (about 2 years of data). These points correspond to significant shifts in the stock's direction—either from increasing to decreasing or vice-versa.

After applying Newton-Raphson, we obtained a refined list of actual turning points:

```
print(roots)
```

```
## [1]  89.0 309.4 394.0 475.4
```

To understand how frequently Coca-Cola's stock price trends change direction, we compute the differences between successive turning points and take their average:

```
intervals = diff(roots)
average_interval = mean(intervals)
intervals
```

```
## [1] 220.4  84.6  81.4
```

```r
average_interval
```

```
## [1] 128.8
```

```r
round(average_interval)
```

```
## [1] 129
```

This output tells us The intervals between turning points (in trading days) and the average interval, which represents how often the price trend repeats. Thus we find that Coca-Cola's price trend reverses direction every 129 trading days, which corresponds to approximately every 3.5 to 4 months.