

```

import statistics as stats
from pandas import *
from pandas_datareader import data
from matplotlib.pyplot import *

set_option ("display.max_rows", 20000)
set_option ("display.max_columns", 1000)
set_option ("display.width", 1000)

start_date = "2014-01-01"
end_date = "2020-02-22"
google_data = data.DataReader( "GOOG", "yahoo", start_date, end_date )

time_period = 20 # look back period to compute gains & losses
gain_history = [] # history of gains over look back period (0 if no gain, magnitude
of gain if gain)
loss_history = [] # history of losses over look back period (0 if no loss,
magnitude of loss if loss)

avg_gain_values = [] # track avg gains for visualization purposes
avg_loss_values = [] # track avg losses for visualization purposes
rsi_values = [] # track computed RSI values

last_price = 0 # current_price - last_price > 0 => gain // current_price -
last_price < 0 => loss.

for close_price in google_data ["Adj Close"]:

    if last_price == 0:
        last_price = close_price

    gain_history.append (max (0, close_price - last_price))
    loss_history.append (max (0, last_price - close_price))
    last_price = close_price

    if len (gain_history) > time_period: # maximum observations is equal to
lookback period
        del (gain_history[0])
        del (loss_history[0])

    avg_gain = stats.mean (gain_history) # average gain over lookback period
    avg_loss = stats.mean (loss_history) # average loss over lookback period
    avg_gain_values.append (avg_gain)
    avg_loss_values.append (avg_loss)

    rs = 0
    if avg_loss > 0: # to avoid division by 0, which is undefined
        rs = avg_gain / avg_loss
    rsi = 100 - (100 / (1 + rs))
    rsi_values.append (rsi)

```

```

google_data = google_data.assign(RelativeStrengthAvgGainOver20Days=
Series(avg_gain_values, index=google_data.index))
google_data = google_data.assign(RelativeStrengthAvgLossOver20Days=
Series(avg_loss_values, index=google_data.index))
google_data = google_data.assign(RelativeStrengthIndicatorOver20Days=
Series(rsi_values, index=google_data.index))
google_data = google_data.assign(Middle_line_50= Series(50.,
index=google_data.index))

```

```

close_price = google_data['Adj Close']
rs_gain = google_data['RelativeStrengthAvgGainOver20Days']
rs_loss = google_data['RelativeStrengthAvgLossOver20Days']
rsi = google_data['RelativeStrengthIndicatorOver20Days']
middle_line = google_data ["Middle_line_50"]

```

```

fig = figure()
ax1 = fig.add_subplot(311, ylabel='Google price in $')
close_price.plot(ax=ax1, color='black', lw=2., legend=True)
ax2 = fig.add_subplot(312, ylabel='RS')
rs_gain.plot(ax=ax2, color='g', lw=2., legend=True)
rs_loss.plot(ax=ax2, color='r', lw=2., legend=True)
ax3 = fig.add_subplot(313, ylabel='RSI')
rsi.plot(ax=ax3, color='b', lw=2., legend=True)
middle_line.plot (ax=ax3, color="black", lw=2., legend=True)
legend ()
show()

```