

---

## Table of Contents

.....	1
Initial Values .....	1
Main .....	1
Graphs .....	4
Print Results .....	8
Function Definitions .....	8

```
% Author: Andrew Dwyer
% Final Project Algorithm 1
```

```
% This algorithm decides to buy or sell stocks by taking the moving average
% of n number of days and comparing the square of the difference between
% the current stock price and the moving average. The square of the
% difference is used to make dramatic changes impact the algorithm more. If
% the square of the difference is greater or less than a certain percent
% difference, the stock is bought or sold. The stocks are looped through in
% a randomized order to prevent certain stocks from always being bought or
% sold and others not impacted. The algorithm also checks that the bank
% account has sufficient funds to buy the stock, or the portfolio has the
% shares needed to sell. Only one share of a stock is ever bought or sold
% on one day.
```

```
clear, clc, close all
format bank
```

```
load StockData.mat
```

## Initial Values

```
bank_initial=1000;
bank = bank_initial;
portfolio = zeros(length(Companies),1);
delta_d = 14; % the step of days to step each iteration
num_days_avg = 14; % the number of days to look at the moving average over
% this is the percent off the moving average that will cause the stock to be
    bought or sold
percent_window = .12;
portfolio_days=[];
net_worth=bank;
```

## Main

```
i=0;

% set initial points on the graph
figure(1), hold on
plot(Dates(1),bank, 'k.')
figure(2), hold on
```

---

```

plot(Dates(1), bank, "r.")

% loop over the days with a step of delta_d
for d=(1+num_days_avg):delta_d:length(Dates)
    i=i+1;
    % randomize the order the companies are chosen
    rand_comp=randperm(length(Companies));
    day_prices = zeros(length(Companies),1);
    % loop over the companies
    for c=1:length(Companies)
        % the index of the current company
        curr_comp_index = rand_comp(c);
        % get the current price of the company's stock
        price = Stocks.(Companies(curr_comp_index,4))(d);

        % if the price is NaN then skip the company and continue on to the
        % next company
        if(isnan(price))
            continue
        end

        % run the algorithm to determine whether to buy or sell
        trans_type=logit(Stocks.(Companies(curr_comp_index,4)), d, ...
            num_days_avg, percent_window);

        % from the algorithm perform the transaction
        [portfolio, bank]=transaction(price, ...
            curr_comp_index,trans_type,bank,portfolio);

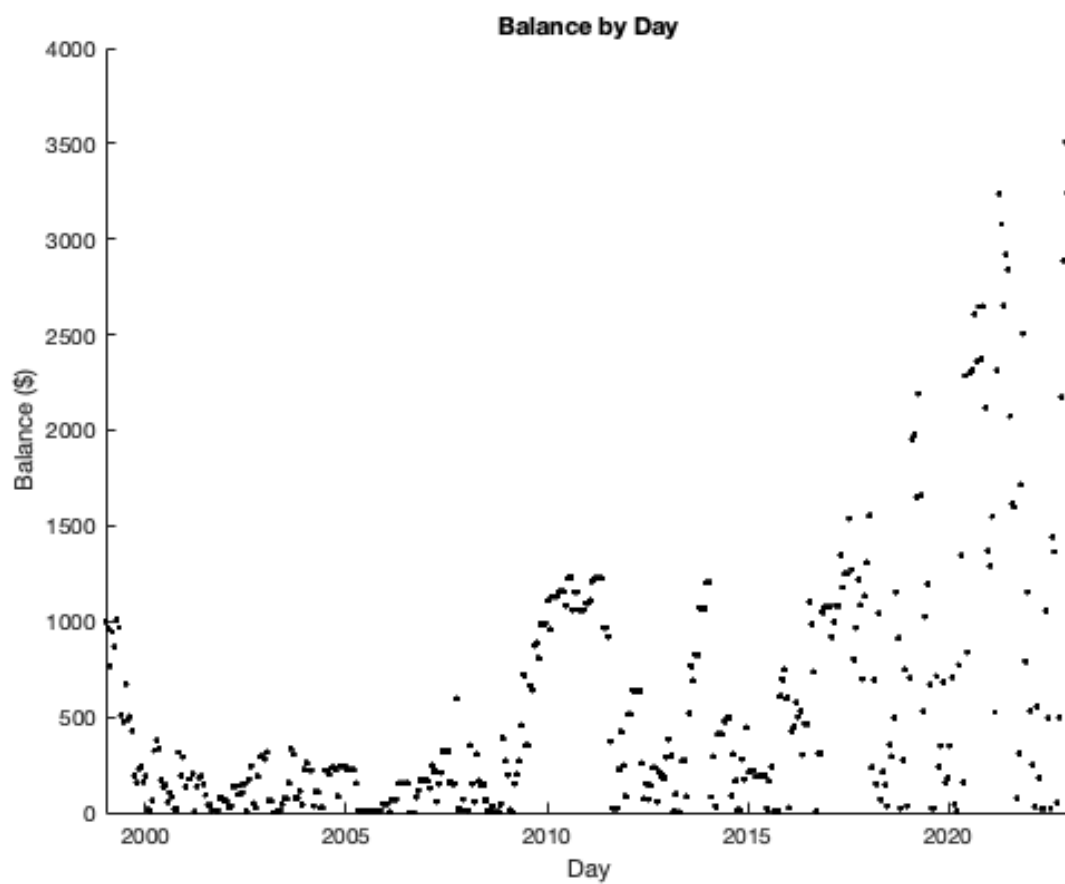
        % save the values to be graphed later
        day_prices(curr_comp_index) = price;
        portfolio_days(curr_comp_index,i) = portfolio(curr_comp_index);
    end

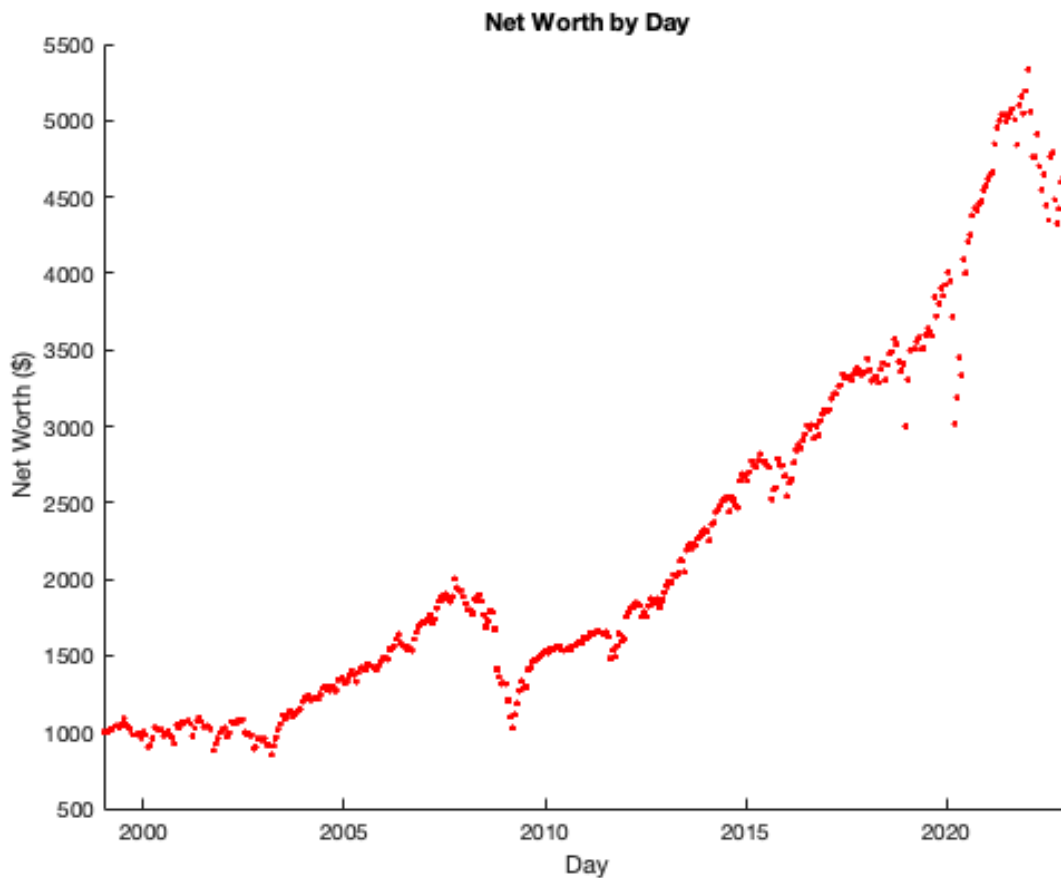
    % Graphs
    % graph the bank balance at the end of the day
    figure(1), hold on
    plot(Dates(d),bank, 'k.')
    xlabel("Day"),ylabel("Balance ($)"), title("Balance by Day")

    % graph the networth at the end of the day
    figure(2), hold on
    net_worth = sum(portfolio .* day_prices) + bank;
    plot(Dates(d), net_worth, "r.")
    xlabel("Day"),ylabel("Net Worth ($)"), title("Net Worth by Day")
end

```

---



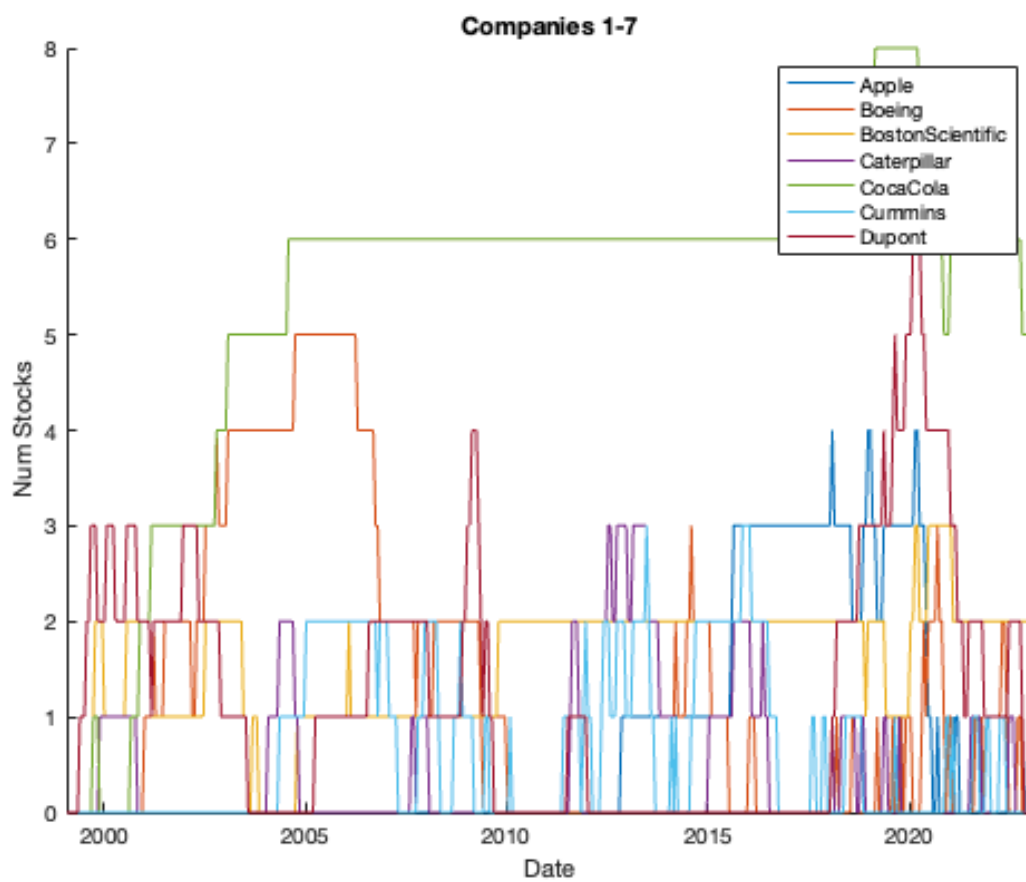


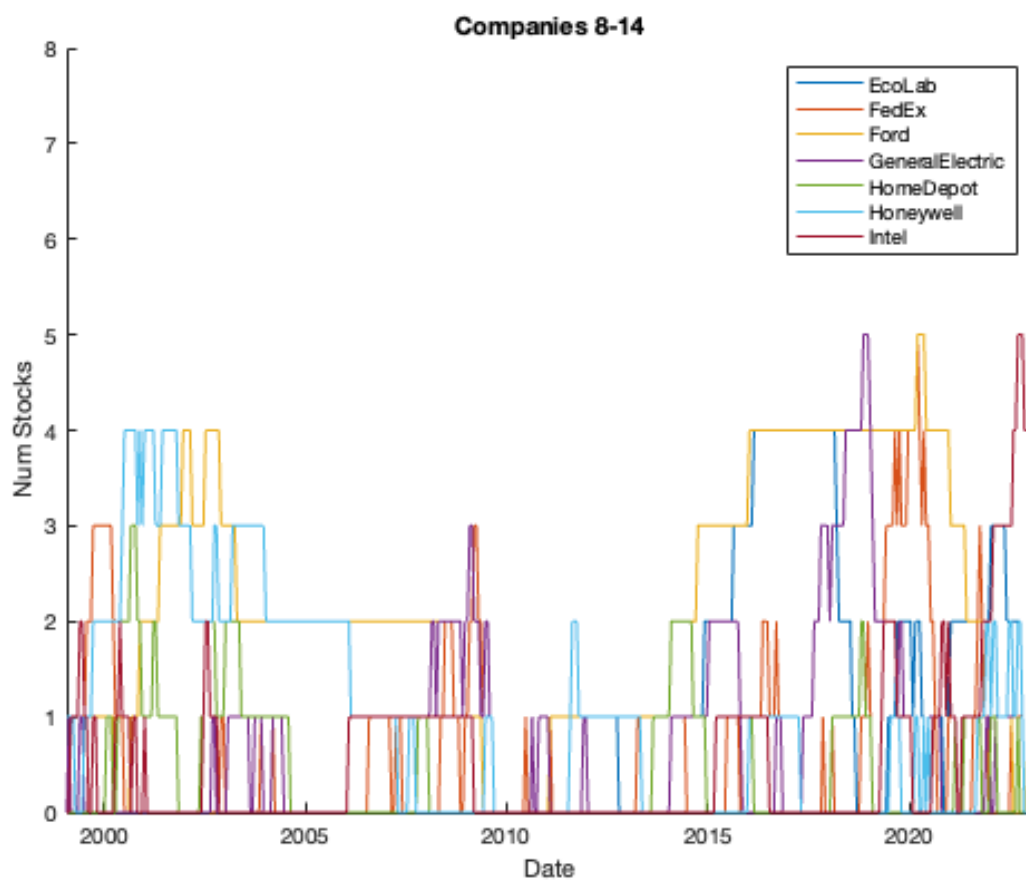
## Graphs

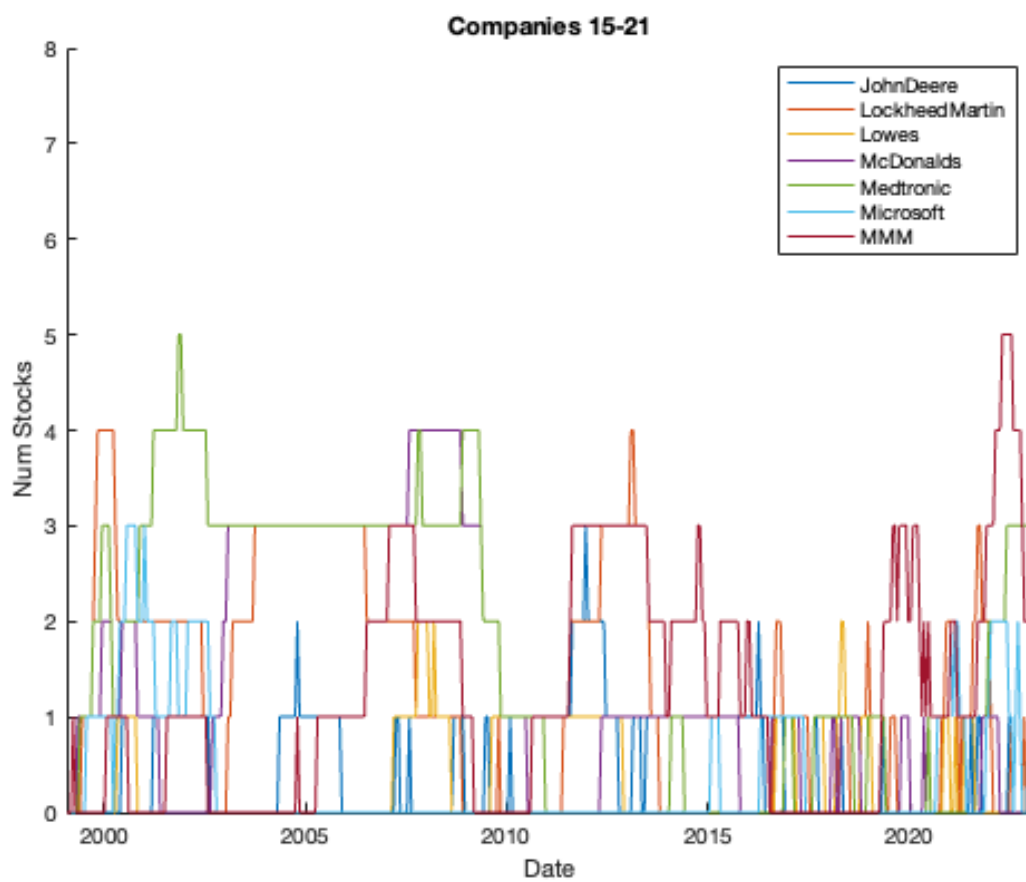
graph the number of stocks in the portfolio over time creating a seperate graph for each company

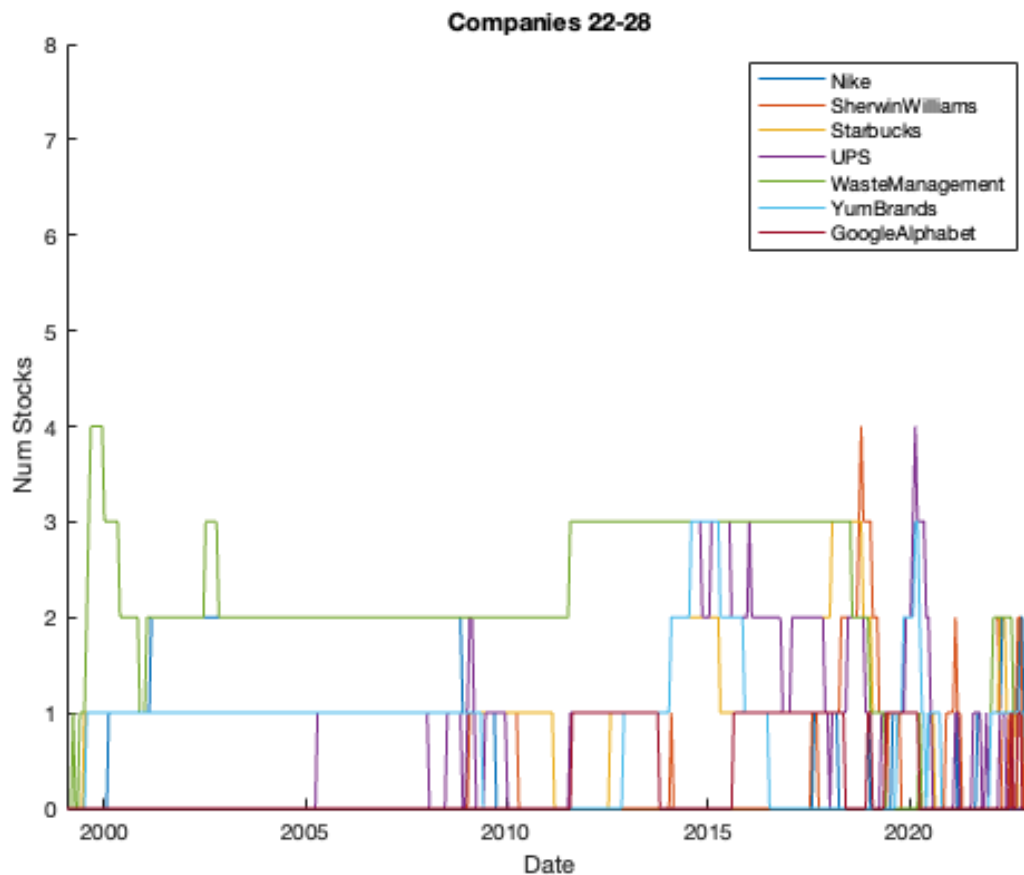
```
for i=1:length(Companies)
    figure(ceil(i/7)+2), hold on
    d=(1+num_days_avg):delta_d:length(Dates);
    plot(Dates(d), portfolio_days(i,:))
    ylabel('Num Stocks'), xlabel('Date');
end

figure(3), title("Companies 1-7"), legend(Companies(1:7,1)), ylim([0
    max(portfolio_days, [], 'all')])
figure(4), title("Companies 8-14"), legend(Companies(8:14,1)), ylim([0
    max(portfolio_days, [], 'all')])
figure(5), title("Companies 15-21"), legend(Companies(15:21,1)), ylim([0
    max(portfolio_days, [], 'all')])
figure(6), title("Companies 22-28"), legend(Companies(22:28,1)), ylim([0
    max(portfolio_days, [], 'all')])
```









## Print Results

```
fprintf("Bank: $%5.2f\n", bank)
fprintf("Net Worth: $%5.2f\n", net_worth)
fprintf("Num Stocks: %i\n", sum(portfolio))
fprintf("%% Gain: %3.2f%%\n", (net_worth/bank_initial - 1)*100)
```

```
Bank: $3642.12
Net Worth: $4654.91
Num Stocks: 17
% Gain: 365.49%
```

## Function Definitions

```
function [trans_type, avg]=logic(prices, idate, num_days_avg, percent_window)
% Determines if the stock for a company should be bought or sold based
% on whether the square of the difference between the n day moving
% average is outside a window around the moving mean.
%
% company is the current company vector
% idate is the day index and must be greater than num_days_avg and less
% than the length of the stock prices vector.
```



---

```

% calculate the moving average over the last n days
avg=mean(prices(idate-num_days_avg:idate));

% calculate the square of the difference of the current date vs the ma
d_price = avg - prices(idate); % this is the difference in price
d2_price = sign(d_price)*d_price^2;

% compare the average vs percent of the moving mean
trans_type=0;

if(d2_price < -percent_window*avg) % then sell
    trans_type=-1;
elseif(d2_price > percent_window*avg) % then buy
    trans_type=1;
end
end

function
[portfolio,bank]=transaction(stock_price,icompany,trans_type,bank,portfolio)
    % based on the trans_type determine if it is possible based on balance
    % update the balance and the portfolio
    % only buys or sells one stock at a time

    % check if trans_type is sell and the portfolio has the stock to sell
    if(trans_type < 0) && (portfolio(icompany) > 0) % then sell
        portfolio(icompany) = portfolio(icompany) - 1; % remove from the
portfolio
        bank = bank + stock_price; % add funds to the bank
    % if trans_type is buy and the bank has the funds to buy
    elseif (trans_type > 0) && (bank >= stock_price) % then buy
        portfolio(icompany) = portfolio(icompany) + 1; % add to the protfolio
        bank = bank - stock_price; % remove funds from the bank
    end
    % else do nothing because the resource to preform the action are not
    % available
end

```

*Published with MATLAB® R2022b*