**MainFunction.m** % A function that returns a column vector of buy signals and a column

% vector of sell signals for days 1..n based on technical analysis

% of a security's 14 day price history.

%

% Inputs:

% high - a column vector of highest recorded price for days 1..n

% low - a column vector of lowest recorded price for days 1..n

% close - a column vector of the closing price for days 1..n

% alpha - a constant used to determine how much to discount information

% as you go futher back. An alpha of 0.9 means that you weigh

% yesterday's values 90% of the weight given to the current

% day. (0 < alpha < 1)

% sensitivity - a constant used to determine how sensitive the trigger

% for the buy/sell signal is. Values range from 1 to 10.

% Values closer to one reflect short term price movements

% while values closer to ten are more representative of a

% long run trend.

% Outputs:

% buy - binary vector where 1 represents a buy recommendation and

% 0 represents a sell recommendation

% sell - binary vector where 1 represents a sell recommendation and

% 0 represents a buy recommendation.

%

function [buy, sell]= trender(high,low,close,alpha,sensitivity)

n = length(close);

buy = zeros(n,1);

sell = zeros(n,1);

% calculates the weights needed for a 14 day exponential moving average

p = 1:14;

raw = alpha.^p;

wt = raw/(sum(raw));

% calculates the Exponential Moving Average of the Midpoint (for t > 15)

MP = (high+low)/2;

EMAvg\_MP = filter(wt,1,MP);

EMAvg\_MP = [zeros(15,1); EMAvg\_MP(16:n)];

% calculates the Exponential Moving Average of the True Range (for t > 15)

yesterday\_close = [close(1); close(1:n-1)];

TR = max(high-low, max(abs(high - yesterday\_close), abs(low - yesterday\_close)));

EMAvg\_TR = filter(wt,1,TR);

EMAvg\_TR = [zeros(15,1); EMAvg\_TR(16:n)];

% calculates the standard deviation of EMavg\_TR for past 14 days (t > 29)

std\_dev\_TR = zeros(n,1);

for t= 30:n

partition = EMAvg\_TR(t-14:t-1);

std\_dev\_TR(t) = std(partition);

end

% Computes the Trender\_Up values for days 16..n.

% If a price closes above the Trender Up value for the day,

% then a BUY signal is triggered.

Trender\_UP = EMAvg\_MP + 0.5\*EMAvg\_TR + sensitivity\*std\_dev\_TR;

% Computes the Trender\_DOWN values for days 16..n.

% If a price closes below the Trender Down value for the day,

% then a SELL signal is triggered.

Trender\_DOWN = EMAvg\_MP - 0.5\*EMAvg\_TR - sensitivity\*std\_dev\_TR;

% Sets both Trender\_UP and Trender\_DOWN for t < 30 to the

% average price from 2..t. This ensures each day prior to t=30

% will have a BUY/SELL signal generated.

for t=2:29

Trender\_UP(t) = mean(close(1:t));

Trender\_DOWN(t) = mean(close(1:t));

end

% Constructs the output vectors buy and sell.

% Note: both buy(1) and sell(1) will be 0 since one day of price history

% is not sufficient to predict a trend.

for t=2:n

% a new buy signal - close at time t crosses above Trender\_UP

if(Trender\_UP(t) < close(t))

buy(t) = 1;

% a new sell signal - close at time t crosses below Trender\_DOWN

elseif(Trender\_DOWN(t) > close(t))

sell(t) = 1;

% no new signal - maintain previous buy/sell recommendation

else

buy(t) = buy(t-1);

sell(t) = sell(t-1);

end

end

**Investment.m** function that returns a column vector of buy signals and a column

% vector of sell signals for days 1..n based on technical analysis

% of a security's 14 day price history.

%

% Inputs:

% high - a column vector of highest recorded price for days 1..n

% low - a column vector of lowest recorded price for days 1..n

% close - a column vector of the closing price for days 1..n

% alpha - a constant used to determine how much to discount information

% as you go futher back. An alpha of 0.9 means that you weigh

% yesterday's values 90% of the weight given to the current

% day. (0 < alpha < 1)

% sensitivity - a constant used to determine how sensitive the trigger

% for the buy/sell signal is. Values range from 1 to 10.

% Values closer to one reflect short term price movements

% while values closer to ten are more representative of a

% long run trend.

% Outputs:

% buy - binary vector where 1 represents a buy recommendation and

% 0 represents a sell recommendation

% sell - binary vector where 1 represents a sell recommendation and

% 0 represents a buy recommendation.

%

function [buy, sell]= trender(high,low,close,alpha,sensitivity)

n = length(close);

buy = zeros(n,1);

sell = zeros(n,1);

% calculates the weights needed for a 14 day exponential moving average

p = 1:14;

raw = alpha.^p;

wt = raw/(sum(raw));

% calculates the Exponential Moving Average of the Midpoint (for t > 15)

MP = (high+low)/2;

EMAvg\_MP = filter(wt,1,MP);

EMAvg\_MP = [zeros(15,1); EMAvg\_MP(16:n)];

% calculates the Exponential Moving Average of the True Range (for t > 15)

yesterday\_close = [close(1); close(1:n-1)];

TR = max(high-low, max(abs(high - yesterday\_close), abs(low - yesterday\_close)));

EMAvg\_TR = filter(wt,1,TR);

EMAvg\_TR = [zeros(15,1); EMAvg\_TR(16:n)];

% calculates the standard deviation of EMavg\_TR for past 14 days (t > 29)

std\_dev\_TR = zeros(n,1);

for t= 30:n

partition = EMAvg\_TR(t-14:t-1);

std\_dev\_TR(t) = std(partition);

end

% Computes the Trender\_Up values for days 16..n.

% If a price closes above the Trender Up value for the day,

% then a BUY signal is triggered.

Trender\_UP = EMAvg\_MP + 0.5\*EMAvg\_TR + sensitivity\*std\_dev\_TR;

% Computes the Trender\_DOWN values for days 16..n.

% If a price closes below the Trender Down value for the day,

% then a SELL signal is triggered.

Trender\_DOWN = EMAvg\_MP - 0.5\*EMAvg\_TR - sensitivity\*std\_dev\_TR;

% Sets both Trender\_UP and Trender\_DOWN for t < 30 to the

% average price from 2..t. This ensures each day prior to t=30

% will have a BUY/SELL signal generated.

for t=2:29

Trender\_UP(t) = mean(close(1:t));

Trender\_DOWN(t) = mean(close(1:t));

end

% Constructs the output vectors buy and sell.

% Note: both buy(1) and sell(1) will be 0 since one day of price history

% is not sufficient to predict a trend.

for t=2:n

% a new buy signal - close at time t crosses above Trender\_UP

if(Trender\_UP(t) < close(t))

buy(t) = 1;

% a new sell signal - close at time t crosses below Trender\_DOWN

elseif(Trender\_DOWN(t) > close(t))

sell(t) = 1;

% no new signal - maintain previous buy/sell recommendation

else

buy(t) = buy(t-1);

sell(t) = sell(t-1);

end

end

**Markowitz.m** function that returns a column vector of buy signals and a column

% vector of sell signals for days 1..n based on technical analysis

% of a security's 14 day price history.

%

% Inputs:

% high - a column vector of highest recorded price for days 1..n

% low - a column vector of lowest recorded price for days 1..n

% close - a column vector of the closing price for days 1..n

% alpha - a constant used to determine how much to discount information

% as you go futher back. An alpha of 0.9 means that you weigh

% yesterday's values 90% of the weight given to the current

% day. (0 < alpha < 1)

% sensitivity - a constant used to determine how sensitive the trigger

% for the buy/sell signal is. Values range from 1 to 10.

% Values closer to one reflect short term price movements

% while values closer to ten are more representative of a

% long run trend.

% Outputs:

% buy - binary vector where 1 represents a buy recommendation and

% 0 represents a sell recommendation

% sell - binary vector where 1 represents a sell recommendation and

% 0 represents a buy recommendation.

%

function [buy, sell]= trender(high,low,close,alpha,sensitivity)

n = length(close);

buy = zeros(n,1);

sell = zeros(n,1);

% calculates the weights needed for a 14 day exponential moving average

p = 1:14;

raw = alpha.^p;

wt = raw/(sum(raw));

% calculates the Exponential Moving Average of the Midpoint (for t > 15)

MP = (high+low)/2;

EMAvg\_MP = filter(wt,1,MP);

EMAvg\_MP = [zeros(15,1); EMAvg\_MP(16:n)];

% calculates the Exponential Moving Average of the True Range (for t > 15)

yesterday\_close = [close(1); close(1:n-1)];

TR = max(high-low, max(abs(high - yesterday\_close), abs(low - yesterday\_close)));

EMAvg\_TR = filter(wt,1,TR);

EMAvg\_TR = [zeros(15,1); EMAvg\_TR(16:n)];

% calculates the standard deviation of EMavg\_TR for past 14 days (t > 29)

std\_dev\_TR = zeros(n,1);

for t= 30:n

partition = EMAvg\_TR(t-14:t-1);

std\_dev\_TR(t) = std(partition);

end

% Computes the Trender\_Up values for days 16..n.

% If a price closes above the Trender Up value for the day,

% then a BUY signal is triggered.

Trender\_UP = EMAvg\_MP + 0.5\*EMAvg\_TR + sensitivity\*std\_dev\_TR;

% Computes the Trender\_DOWN values for days 16..n.

% If a price closes below the Trender Down value for the day,

% then a SELL signal is triggered.

Trender\_DOWN = EMAvg\_MP - 0.5\*EMAvg\_TR - sensitivity\*std\_dev\_TR;

% Sets both Trender\_UP and Trender\_DOWN for t < 30 to the

% average price from 2..t. This ensures each day prior to t=30

% will have a BUY/SELL signal generated.

for t=2:29

Trender\_UP(t) = mean(close(1:t));

Trender\_DOWN(t) = mean(close(1:t));

end

% Constructs the output vectors buy and sell.

% Note: both buy(1) and sell(1) will be 0 since one day of price history

% is not sufficient to predict a trend.

for t=2:n

% a new buy signal - close at time t crosses above Trender\_UP

if(Trender\_UP(t) < close(t))

buy(t) = 1;

% a new sell signal - close at time t crosses below Trender\_DOWN

elseif(Trender\_DOWN(t) > close(t))

sell(t) = 1;

% no new signal - maintain previous buy/sell recommendation

else

buy(t) = buy(t-1);

sell(t) = sell(t-1);

end

end

**Trend\_matrices.m** % A function that returns a column vector of buy signals and a column

% vector of sell signals for days 1..n based on technical analysis

% of a security's 14 day price history.

%

% Inputs:

% high - a column vector of highest recorded price for days 1..n

% low - a column vector of lowest recorded price for days 1..n

% close - a column vector of the closing price for days 1..n

% alpha - a constant used to determine how much to discount information

% as you go futher back. An alpha of 0.9 means that you weigh

% yesterday's values 90% of the weight given to the current

% day. (0 < alpha < 1)

% sensitivity - a constant used to determine how sensitive the trigger

% for the buy/sell signal is. Values range from 1 to 10.

% Values closer to one reflect short term price movements

% while values closer to ten are more representative of a

% long run trend.

% Outputs:

% buy - binary vector where 1 represents a buy recommendation and

% 0 represents a sell recommendation

% sell - binary vector where 1 represents a sell recommendation and

% 0 represents a buy recommendation.

%

function [buy, sell]= trender(high,low,close,alpha,sensitivity)

n = length(close);

buy = zeros(n,1);

sell = zeros(n,1);

% calculates the weights needed for a 14 day exponential moving average

p = 1:14;

raw = alpha.^p;

wt = raw/(sum(raw));

% calculates the Exponential Moving Average of the Midpoint (for t > 15)

MP = (high+low)/2;

EMAvg\_MP = filter(wt,1,MP);

EMAvg\_MP = [zeros(15,1); EMAvg\_MP(16:n)];

% calculates the Exponential Moving Average of the True Range (for t > 15)

yesterday\_close = [close(1); close(1:n-1)];

TR = max(high-low, max(abs(high - yesterday\_close), abs(low - yesterday\_close)));

EMAvg\_TR = filter(wt,1,TR);

EMAvg\_TR = [zeros(15,1); EMAvg\_TR(16:n)];

% calculates the standard deviation of EMavg\_TR for past 14 days (t > 29)

std\_dev\_TR = zeros(n,1);

for t= 30:n

partition = EMAvg\_TR(t-14:t-1);

std\_dev\_TR(t) = std(partition);

end

% Computes the Trender\_Up values for days 16..n.

% If a price closes above the Trender Up value for the day,

% then a BUY signal is triggered.

Trender\_UP = EMAvg\_MP + 0.5\*EMAvg\_TR + sensitivity\*std\_dev\_TR;

% Computes the Trender\_DOWN values for days 16..n.

% If a price closes below the Trender Down value for the day,

% then a SELL signal is triggered.

Trender\_DOWN = EMAvg\_MP - 0.5\*EMAvg\_TR - sensitivity\*std\_dev\_TR;

% Sets both Trender\_UP and Trender\_DOWN for t < 30 to the

% average price from 2..t. This ensures each day prior to t=30

% will have a BUY/SELL signal generated.

for t=2:29

Trender\_UP(t) = mean(close(1:t));

Trender\_DOWN(t) = mean(close(1:t));

end

% Constructs the output vectors buy and sell.

% Note: both buy(1) and sell(1) will be 0 since one day of price history

% is not sufficient to predict a trend.

for t=2:n

% a new buy signal - close at time t crosses above Trender\_UP

if(Trender\_UP(t) < close(t))

buy(t) = 1;

% a new sell signal - close at time t crosses below Trender\_DOWN

elseif(Trender\_DOWN(t) > close(t))

sell(t) = 1;

% no new signal - maintain previous buy/sell recommendation

else

buy(t) = buy(t-1);

sell(t) = sell(t-1);

end

end

**Stats.m** function that returns a column vector of buy signals and a column

% vector of sell signals for days 1..n based on technical analysis

% of a security's 14 day price history.

%

% Inputs:

% high - a column vector of highest recorded price for days 1..n

% low - a column vector of lowest recorded price for days 1..n

% close - a column vector of the closing price for days 1..n

% alpha - a constant used to determine how much to discount information

% as you go futher back. An alpha of 0.9 means that you weigh

% yesterday's values 90% of the weight given to the current

% day. (0 < alpha < 1)

% sensitivity - a constant used to determine how sensitive the trigger

% for the buy/sell signal is. Values range from 1 to 10.

% Values closer to one reflect short term price movements

% while values closer to ten are more representative of a

% long run trend.

% Outputs:

% buy - binary vector where 1 represents a buy recommendation and

% 0 represents a sell recommendation

% sell - binary vector where 1 represents a sell recommendation and

% 0 represents a buy recommendation.

%

function [buy, sell]= trender(high,low,close,alpha,sensitivity)

n = length(close);

buy = zeros(n,1);

sell = zeros(n,1);

% calculates the weights needed for a 14 day exponential moving average

p = 1:14;

raw = alpha.^p;

wt = raw/(sum(raw));

% calculates the Exponential Moving Average of the Midpoint (for t > 15)

MP = (high+low)/2;

EMAvg\_MP = filter(wt,1,MP);

EMAvg\_MP = [zeros(15,1); EMAvg\_MP(16:n)];

% calculates the Exponential Moving Average of the True Range (for t > 15)

yesterday\_close = [close(1); close(1:n-1)];

TR = max(high-low, max(abs(high - yesterday\_close), abs(low - yesterday\_close)));

EMAvg\_TR = filter(wt,1,TR);

EMAvg\_TR = [zeros(15,1); EMAvg\_TR(16:n)];

% calculates the standard deviation of EMavg\_TR for past 14 days (t > 29)

std\_dev\_TR = zeros(n,1);

for t= 30:n

partition = EMAvg\_TR(t-14:t-1);

std\_dev\_TR(t) = std(partition);

end

% Computes the Trender\_Up values for days 16..n.

% If a price closes above the Trender Up value for the day,

% then a BUY signal is triggered.

Trender\_UP = EMAvg\_MP + 0.5\*EMAvg\_TR + sensitivity\*std\_dev\_TR;

% Computes the Trender\_DOWN values for days 16..n.

% If a price closes below the Trender Down value for the day,

% then a SELL signal is triggered.

Trender\_DOWN = EMAvg\_MP - 0.5\*EMAvg\_TR - sensitivity\*std\_dev\_TR;

% Sets both Trender\_UP and Trender\_DOWN for t < 30 to the

% average price from 2..t. This ensures each day prior to t=30

% will have a BUY/SELL signal generated.

for t=2:29

Trender\_UP(t) = mean(close(1:t));

Trender\_DOWN(t) = mean(close(1:t));

end

% Constructs the output vectors buy and sell.

% Note: both buy(1) and sell(1) will be 0 since one day of price history

% is not sufficient to predict a trend.

for t=2:n

% a new buy signal - close at time t crosses above Trender\_UP

if(Trender\_UP(t) < close(t))

buy(t) = 1;

% a new sell signal - close at time t crosses below Trender\_DOWN

elseif(Trender\_DOWN(t) > close(t))

sell(t) = 1;

% no new signal - maintain previous buy/sell recommendation

else

buy(t) = buy(t-1);

sell(t) = sell(t-1);

end

end

**Trender.m** function that returns a column vector of buy signals and a column

% vector of sell signals for days 1..n based on technical analysis

% of a security's 14 day price history.

%

% Inputs:

% high - a column vector of highest recorded price for days 1..n

% low - a column vector of lowest recorded price for days 1..n

% close - a column vector of the closing price for days 1..n

% alpha - a constant used to determine how much to discount information

% as you go futher back. An alpha of 0.9 means that you weigh

% yesterday's values 90% of the weight given to the current

% day. (0 < alpha < 1)

% sensitivity - a constant used to determine how sensitive the trigger

% for the buy/sell signal is. Values range from 1 to 10.

% Values closer to one reflect short term price movements

% while values closer to ten are more representative of a

% long run trend.

% Outputs:

% buy - binary vector where 1 represents a buy recommendation and

% 0 represents a sell recommendation

% sell - binary vector where 1 represents a sell recommendation and

% 0 represents a buy recommendation.

%

function [buy, sell]= trender(high,low,close,alpha,sensitivity)

n = length(close);

buy = zeros(n,1);

sell = zeros(n,1);

% calculates the weights needed for a 14 day exponential moving average

p = 1:14;

raw = alpha.^p;

wt = raw/(sum(raw));

% calculates the Exponential Moving Average of the Midpoint (for t > 15)

MP = (high+low)/2;

EMAvg\_MP = filter(wt,1,MP);

EMAvg\_MP = [zeros(15,1); EMAvg\_MP(16:n)];

% calculates the Exponential Moving Average of the True Range (for t > 15)

yesterday\_close = [close(1); close(1:n-1)];

TR = max(high-low, max(abs(high - yesterday\_close), abs(low - yesterday\_close)));

EMAvg\_TR = filter(wt,1,TR);

EMAvg\_TR = [zeros(15,1); EMAvg\_TR(16:n)];

% calculates the standard deviation of EMavg\_TR for past 14 days (t > 29)

std\_dev\_TR = zeros(n,1);

for t= 30:n

partition = EMAvg\_TR(t-14:t-1);

std\_dev\_TR(t) = std(partition);

end

% Computes the Trender\_Up values for days 16..n.

% If a price closes above the Trender Up value for the day,

% then a BUY signal is triggered.

Trender\_UP = EMAvg\_MP + 0.5\*EMAvg\_TR + sensitivity\*std\_dev\_TR;

% Computes the Trender\_DOWN values for days 16..n.

% If a price closes below the Trender Down value for the day,

% then a SELL signal is triggered.

Trender\_DOWN = EMAvg\_MP - 0.5\*EMAvg\_TR - sensitivity\*std\_dev\_TR;

% Sets both Trender\_UP and Trender\_DOWN for t < 30 to the

% average price from 2..t. This ensures each day prior to t=30

% will have a BUY/SELL signal generated.

for t=2:29

Trender\_UP(t) = mean(close(1:t));

Trender\_DOWN(t) = mean(close(1:t));

end

% Constructs the output vectors buy and sell.

% Note: both buy(1) and sell(1) will be 0 since one day of price history

% is not sufficient to predict a trend.

for t=2:n

% a new buy signal - close at time t crosses above Trender\_UP

if(Trender\_UP(t) < close(t))

buy(t) = 1;

% a new sell signal - close at time t crosses below Trender\_DOWN

elseif(Trender\_DOWN(t) > close(t))

sell(t) = 1;

% no new signal - maintain previous buy/sell recommendation

else

buy(t) = buy(t-1);

sell(t) = sell(t-1);

end

end