ECON 4360: Empirical Finance Final Project

Table of Contents

By: Nick Bruno

PerfStat

Import Data

```
[countrydata, countryheader] = xlsread('/Users/nickbruno/Documents/
ECON 4360/Project/newdataset.xlsx','Country');
countrydata(:,1) = []; % deletes the date column
countryheader(:,1) = []; % also deletes the date column


% Now I will repeat this for all 5 different datasets
[factordata, factorheader] = xlsread('/Users/nickbruno/Documents/ECON
 4360/Project/newdataset.xlsx','Factor_BR_AQR');
factordata(:,1) = [];
factorheader(:,1) = [];

[sectordata, sectorheader] = xlsread('/Users/nickbruno/Documents/ECON
 4360/Project/newdataset.xlsx','Sector');
sectordata(:,1) = [];
```

```
sectorheader(:,1) = [];

[styledata, styleheader] = xlsread('/Users/nickbruno/Documents/ECON
 4360/Project/newdataset.xlsx','Style');
styledata(:,1) = [];
styleheader(:,1) = [];

[assetdata, assetheader] = xlsread('/Users/nickbruno/Documents/ECON
 4360/Project/newdataset.xlsx','Asset Class');
assetdata(:,1) = [];
assetheader(:,1) = [];

% setting up the data
workingdata = countrydata; % allows the user to change the working
 data for each dataset
workingheader = countryheader; % same for header

% Important to note that I will use the 'country' dataset for the
 following
% examples, but changing the other datasets equal to 'M' will run the
 same
% statistics for the different datasets

% finding the statistics
[T,k] = size(workingdata);
monthly_return = 100*(prod(1+workingdata/100).^(1/T)-1);
annual_return = 100*(prod(1+workingdata/100).^(12/T)-1);
monthly_risk = std(workingdata)*sqrt(12);
annual_risk = std(workingdata);
sharpe_ratio = annual_return./annual_risk;
skew = skewness(workingdata);
kurtosis = kurtosis(workingdata);

% Now I need to invert the statistics found to create a talbe
country = (workingheader)'; % transposes the header so that now it is
 a column

% Commented out below are the different workingheaders for the
 different
% datasets

% factor = (workingheader)';
% sector = (workingheader)';
% style = (workingheader)';
% asset = (workingheader)';

% Inverting statistics to make each a column for a table that will be
% created later
monthly_return = (monthly_return)';
mean_return = (annual_return)';
monthly_risk = (monthly_risk)';
risk = (annual_risk)';
sharpe = (sharpe_ratio)';
skew = (skew)';
```

```
kurtosis = (kurtosis)';

% Creating the tables
% The table shows the mean return, risk, sharpe ration, skew, and
% kurtosis of each country in the dataset

countrytable = table(country, mean_return, risk, sharpe, skew,
 kurtosis)
% factortable = table(factor, annual_return, risk, sharpe, skew,
 kurtosis)
% sectortable = table(sector, annual_return, risk, sharpe, skew,
 kurtosis)
% styletable = table(style, annual_return, risk, sharpe, skew,
 kurtosis)
% assettable = table(asset, annual_return, risk, sharpe, skew,
 kurtosis)

% Sharpe ratio
countrysharpe = table(country, sharpe)

% The table ashows the mean return, risk, sharpe ration, skew, and
% kurtosis of each country in the dataset
clear;clc;
```

countrytable =

20×6 table

country	mean_return	risk	sharpe	skew	kurtosis
'us'	6.0386	4.1991	1.4381	-0.56487	4.1565
'japan'	3.9667	5.1632	0.76827	-0.30726	3.8786
'UK'	4.9478	3.8872	1.2728	-0.62553	3.7087
'germany'	5.3853	6.0305	0.893	-0.46153	5.127
'france'	5.3479	4.9551	1.0793	-0.43704	3.6967
'italy'	1.0804	5.6368	0.19167	-0.13391	3.6066
'canada'	8.0006	4.1228	1.9406	-0.67712	5.1139
'australia'	10.064	3.6619	2.7483	-0.54881	3.0993
'spain'	5.2042	5.8205	0.89412	-0.13735	3.834
'switzerland'	3.9042	3.8681	1.0093	-0.5377	3.6261

'EU'	4.694	4.3595	1.0767	-0.50431	4.0196
'china'	8.8565	8.4358	1.0499	0.40298	6.7293
'brazil'	10.486	10.575	0.99155	-0.02194	3.927
'india'	15.474	7.249	2.1346	-0.11275	4.1202
'russia'	15.625	11.279	1.3854	0.65018	6.6173
'mexico'	10.994	6.8312	1.6094	-0.40392	4.4577
'turkey'	24.194	12.335	1.9613	1.6662	11.34
'indonesia'	18.275	7.7041	2.3721	-0.054872	4.717
'korea'	11.662	7.1691	1.6267	0.38808	4.7378
'southAfrica'	16.668	5.0143	3.324	-0.036871	3.2186

countrysharpe =

20×2 table

country	sharpe
'us'	1.4381
'japan'	0.76827
'UK'	1.2728
'germany'	0.893
'france'	1.0793
'italy'	0.19167
'canada'	1.9406
'australia'	2.7483
'spain'	0.89412
'switzerland'	1.0093
'EU'	1.0767
'china'	1.0499
'brazil'	0.99155
'india'	2.1346
'russia'	1.3854
'mexico'	1.6094
'turkey'	1.9613
'indonesia'	2.3721
'korea'	1.6267
'southAfrica'	3.324

Financial Crisis Data from November 2007 to June 2009

Import Factor data

```
[factordata, factorheader] = xlsread('/Users/nickbruno/Documents/ECON
 4360/Project/newdataset.xlsx','Factor_BR_AQR');
factordata(:,1) = [];
factorheader(:,1) = [];
crisis_factordata = factordata(107:126, :);


% Now I will look at the crisis factor data
workingdata = crisis_factordata; % changes per data set
workingheader = factorheader; % changes per data set

% finding the sharpe ratio statistics
[T,k] = size(workingdata);
annual_return = 100*(prod(1+workingdata/100).^(12/T)-1);
annual_risk = std(workingdata);
sharpe_ratio = annual_return./annual_risk;


% Invert the statistics found to create a talbe
factor = (workingheader)';
sharpe = (sharpe_ratio)';


% Creating the Factor tables
crisis_factor_sharpe_table = table(factor, sharpe)
    % Looks at a table with the factors and their corresponding Sharpe
    % ratio

% Comparing crisis and Post-crisis Factor data
% reload factor dataset and change dataset from dates Jul 09 to Dec 17
 to
% represent the post-crisis data
```

```
crisis_factor_sharpe_table =

  12×2 table

        factor            sharpe
    _________________    ________

    'RealRates'           0.72083
    'Inflation'            0.7944
    'Credit'              -1.8475
    'Economic'            -4.1263
    'EmergingMarkets'    -0.95313
    'Liquidity'           0.62245
```

```
'BAB'                -3.2701
'MKT'                -3.3746
'SMB'               -0.67483
'HML'                -1.9894
'UMD'                -1.9205
'RF'                  13.736
```

Post-Crisis Data July 2009 to December 2017

```
[factordata, factorheader] = xlsread('/Users/nickbruno/Documents/ECON
 4360/Project/newdataset.xlsx','Factor_BR_AQR');
factordata(:,1) = [];
factorheader(:,1) = [];
post_crisis_factordata = factordata(127:end, :);

% set working data set
workingdata = post_crisis_factordata;
workingheader = factorheader;

% finding the sharpe ratio statistics
[T,k] = size(workingdata);
annual_return = 100*(prod(1+workingdata/100).^(12/T)-1);
annual_risk = std(workingdata);
sharpe_ratio = annual_return./annual_risk;

% Invert statistics
factor = (workingheader)';
sharpe = (sharpe_ratio)';

% Sharpe table
post_crisis_factor_sharpe_table = table(factor, sharpe)

% Sharpe ratios post-crisis were higher compared to crisis data for
 the
% 'factor' portfolio

clear;clc;
```

```
post_crisis_factor_sharpe_table =

  12×2 table

       factor             sharpe
    _________________    _________

    'RealRates'             3.5503
    'Inflation'           -0.14294
    'Credit'                4.2266
    'Economic'              2.9166
    'EmergingMarkets'    0.0042838
    'Liquidity'             3.0923
```

```
'BAB'                    9.5026
'MKT'                     3.084
'SMB'                   0.54362
'HML'                   0.20606
'UMD'                    2.9531
'RF'                     8.4137
```

Portfolio Selection, Optimization, and Attribution

Import and select data

```
country = readtable('/Users/nickbruno/Documents/ECON 4360/Project/
newdataset.xlsx','Sheet','Country');
date = datetime(country.Date);
M = [country.us, country.italy, country.australia, country.brazil,
 country.india, country.southAfrica];
countryselectedVariables =
 {'US';'Italy';'australia';'Brazil';'India';'South Africa'};

% Importing other datasets
    % Important to note that changing P,Q, and R equal to M will allow
 for
    % the code to work for each dataset
sector = readtable('/Users/nickbruno/Documents/ECON 4360/Project/
newdataset.xlsx','Sheet','Sector');
% date = datetime(sector.Date); (uncomment when looking at sector
 data)
P = [sector.Energy, sector.IT, sector.Materials,
 sector.Consumer_Discretionary, sector.Financials, sector.TeleComm,
 sector.Industrials, sector.Health_Care, sector.Consumer_Staples,
 sector.Utilities];
sectorselectedVariables = {'Energy';'IT';'Materials';'Consumer
 Discretionary';'Financials';'TeleComm';'Industrials';'Health
 Care';'Consumer Staples';'Utilities'};

style = readtable('/Users/nickbruno/Documents/ECON 4360/Project/
newdataset.xlsx','Sheet','Style');
% date = datetime(style.Date);
Q = [style.MSCI_US_MOM, style.MSCI_US_QUAL, style.MSC_US_SC,
 style.R2000_GROWTH, style.R2000_VALUE, style.MSCI_WORLD_MIN_VOL];
styleselectedVariables = {'MSCI US MOM';'MSCI US QUAL';'MSC US
 SC';'R200GROWTH';'R2000VALUE';'MSCI WORLD MIN VOL'};


asset = readtable('/Users/nickbruno/Documents/ECON 4360/Project/
newdataset.xlsx','Sheet','Asset Class');
% date = datetime(asset.Date);
R = [asset.MSCI_WORLD_X_US, asset.MSCI_US_RE,
 asset.SP_GSCI_Tot_Return_Indx, asset.CBOE_SP_500_PutWrite,
 asset.US_Credit, asset.CREDIT_SUISSE_LS];
```

```
assetselectedVariables = {'MSCIE WORLD X US';'MSCI US RE';'S&P GSCI
 Tot Return Indx';'CBOE S&P 500 PutWrite';'U.S. Credit';'CREDIT SUISSE
 LS'};


% Setting working data and header
% Country
workingdata = M; % allows the user to change the working data for each
 dataset
workingheader = countryselectedVariables;

% Find size and mean return
[T,k] = size(M);
mean_return = 100*(prod(1+M/100).^(1/T)-1);

% Sector
% workingdata = P;
% workingheader = sectorselectedVariables;

% Style
% workingdata = Q;
% workingheader = styleselectedVariables;

% Asset
% workingdata = R;
% workingheader = asssetselectedVariables;

% finding the statistics
[T,k] = size(workingdata);
monthly_return = 100*(prod(1+workingdata/100).^(1/T)-1);
annual_return = 100*(prod(1+workingdata/100).^(12/T)-1);
    % Same as monthly return, but annualizing it
monthly_risk = std(workingdata)*sqrt(12);
annual_risk = std(workingdata);
sharpe_ratio = annual_return./annual_risk;
skew = skewness(workingdata);
kurtosis = kurtosis(workingdata);

% Invert statistics
country = (workingheader); % will change based off of dataset used
monthly_return = (monthly_return)';
annual_return = (annual_return)';
monthly_risk = (monthly_risk)';
risk = (annual_risk)';
sharpe = (sharpe_ratio)';
skew = (skew)';
kurtosis = (kurtosis)';

% Creating the tables
countrytable = table(country, annual_return, risk, sharpe, skew,
 kurtosis)
% sectortable = table(factor, annual_return, risk, sharpe, skew,
 kurtosis)
```

```
% styletable = table(style, annual_return, risk, sharpe, skew,
 kurtosis)
% assettable = table(asset, annual_return, risk, sharpe, skew,
 kurtosis)
```

```
Warning: Variable names were modified to make them valid MATLAB
 identifiers. The
original names are saved in the VariableDescriptions property.
Warning: Variable names were modified to make them valid MATLAB
 identifiers. The
original names are saved in the VariableDescriptions property.

countrytable =

  6×6 table

     country          annual_return     risk      sharpe       skew
   kurtosis
    ______________    _____________    ______    _______    _________
   ________

    'US'              6.0386            4.1991     1.4381     -0.56487
   4.1565
    'Italy'           1.0804            5.6368    0.19167     -0.13391
   3.6066
    'australia'       10.064            3.6619     2.7483     -0.54881
   3.0993
    'Brazil'          10.486            10.575    0.99155     -0.02194
    3.927
    'India'           15.474             7.249     2.1346     -0.11275
   4.1202
    'South Africa'    16.668            5.0143      3.324    -0.036871
   3.2186
```

Portfolio Optimization Constraints & Options

Statistics

```
[T,k] = size(M); % again, M can be changed to P, Q, or R depending on
 which dataset is used
    % or just set the data for the workingdata equal to M
mean_return = 100*(prod(1+M/100).^(1/T)-1);
nAssets = numel(mean_return); r = mean(mean_return);     % number of
 assets and desired return
Aeq = ones(1,nAssets); beq = 1;              % equality Aeq*x = beq
Aineq = -mean_return; bineq = -1.5*r;           % inequality Aineq*x
 <= bineq
lb = zeros(nAssets,1); ub = ones(nAssets,1); % bounds lb <= x <= ub
c = zeros(nAssets,1);                        % objective has no linear
 term; set it to zero

% Select Options for Quadprog
options = optimset('Algorithm','interior-point-convex');
```

```
options = optimset(options,'Display','off','TolFun',1e-10);
```

Optimize monthly using 36 month moving window, saving statistics (weights,returns,risk analysis)

```
for t = 36:T
    V = cov(M(t-35:t,:)); % substitute M with P, Q, or R for other
 datasets
    [wtemp1,fval1] = (quadprog(V,c,Aineq,bineq,Aeq,beq,lb,ub,
[],options));  %targeted return port
    [wtemp2,fval2] = (quadprog(V,c,[],[],Aeq,beq,lb,ub,[],options));
  %Min Var portfolio
    wrp = (rsk_parity(mean_return',V,lb,ub,0));
    wtarg(:,t) = (wtemp1);
    wmv(:,t) = (wtemp2);
    wrp(:,t) = (wrp);
    Rp_mv(t) = (M(t,:)*wmv(:,t));
    Rp_targ(t) = (M(t,:)*wtarg(:,t));
    Rp_rp(t) = (M(t,:)*wrp(:,t));
    risk_wtarg(t) = (sqrt(wtarg(:,t)'*V*wtarg(:,t)));
    risk_wmv(t) = (sqrt(wmv(:,t)'*V*wmv(:,t)));
    risk_rp(t) = (sqrt(wrp(:,t)'*V*wrp(:,t)));
    rho(t) = (sqrt(wtarg(:,t)'*V*wmv(:,t))/
(risk_wtarg(t)*risk_wmv(t)));
    mctr_targ(:,t) = (V*wtarg(:,t)/risk_wtarg(t));
    mctr_mv(:,t) = (V*wmv(:,t)/risk_wmv(t));
    mctr_rp(:,t) = (V*wrp(:,t)/risk_rp(t));
    riskBud_targ(:,t) = (mctr_targ(:,t).*wtarg(:,t));
    riskBud_mv(:,t) = (mctr_mv(:,t).*wmv(:,t));
    riskBud_rp(:,t) = (mctr_rp(:,t).*wrp(:,t));
end
% Make a Table for time t (e.g, t = 117 is Lehman Bros. collapse
 September 2008)
% So 117 + 36 = 153 (adding in the 36 month look-back window).
t = 153;

% Performance table consturction
Performance =
 table(round(wmv(:,t),3),round(wtarg(:,t),3),round(mctr_targ(:,t),3),round(mctr_mv
    round(riskBud_targ(:,t),3),round(riskBud_mv(:,t),3),...
    'VariableNames',
{'wmv','wtarg','mctr_targ','mctr_mv','riskBud_targ','riskBud_mv'},'RowNames',count
    % Note: 'countryselectedVariables' should change to correspond
 with the
    % dataset

% Table of weights, returns, and risks
MiscStats = table(risk_wmv(t), risk_wtarg(t),rho(t),'VariableNames',
{'risk_mv','risk_targ','correlation'})
```

```
% Plots for predicted returns for the targeted return and risk parity
% portfolios
subplot(2,1,1); plot(date(36:end),Rp_mv(36:end)); title('Predicted
 Return - MV Port'); grid on;
subplot(2,1,2); plot(date(36:end),Rp_targ(36:end)); title('Predicted
 Return - Targ Ret Port'); grid on;

% country weights over time in the portfolio
handleFigure = figure;
plot(date(36:end),wtarg(:,36:end));legend(countryselectedVariables);
 title('Cross-section of Port Wts')
```

Performance =

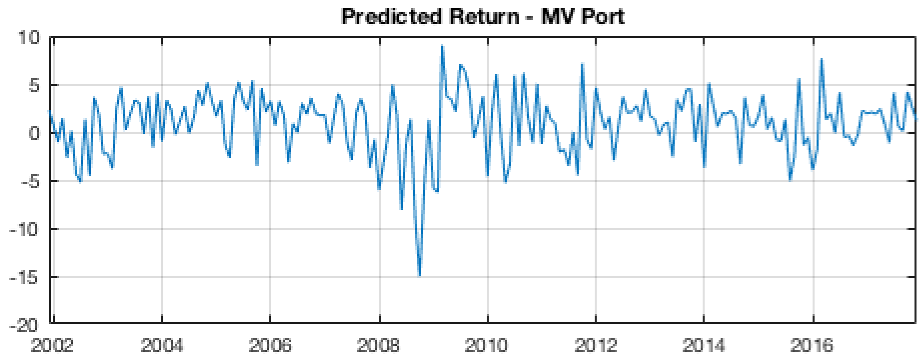
6×6 table

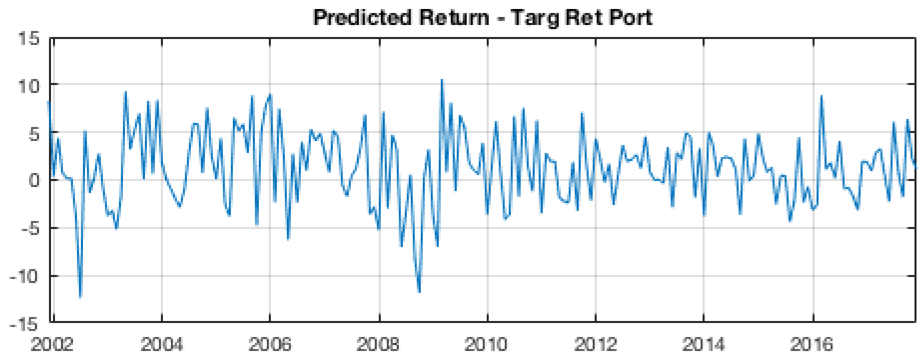
	wmv	wtarg	mctr_targ	mctr_mv	riskBud_targ	riskBud_mv
US	0	0	5.156	5.565	0	0
Italy	0	0	4.748	5.519	0	0
australia	0.621	0.233	3.752	4.405	0.873	2.734
Brazil	0	0	7.782	8.293	0	0
India	0	0	6.251	6.305	0	0
South Africa	0.379	0.767	4.901	4.405	3.761	1.671

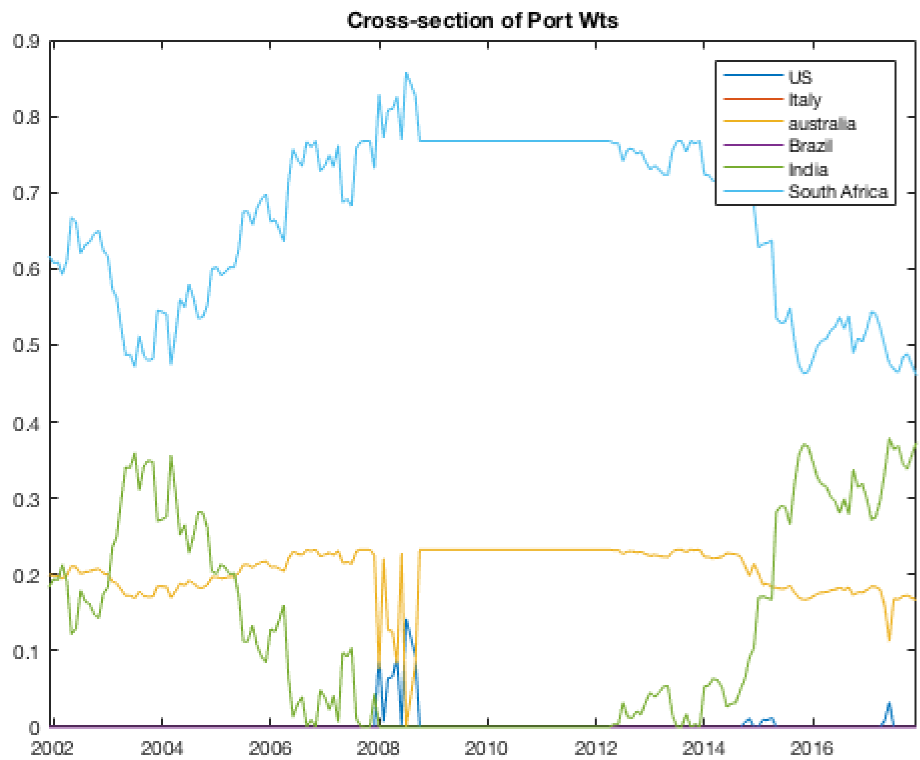
MiscStats =

1×3 table

risk_mv	risk_targ	correlation
4.4051	4.6336	0.21582


Predicted Return - MV Port
10
5
0
-5
-10
-15
-20
2002
2004
2006
2008
2010
2012
2014
2016


Predicted Return - Targ Ret Port
15
10
5
0
-5
-10
-15
2002
2004
2006
2008
2010
2012
2014
2016


Cross-section of Port Wts
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
2002
2004
2006
2008
2010
2012
2014
2016
US
Italy
australia
Brazil
India
South Africa

Monthly Returns Attribution - select factors

```
factors = readtable('/Users/nickbruno/Documents/ECON 4360/Project/
OriginalDataSets.xlsx','Sheet','Factor_BR_AQR');
BR =
 [factors.RealRates,factors.Inflation,factors.Credit,factors.Economic,factors.Emer
AQR = [factors.BAB, factors.MKT,factors.SMB,factors.HML,factors.UMD];

% Select regressors (BR or AQR)
X = [ones(T,1),BR]; Y = Rp_targ';
```

Monthly estimates through 36 month moving window regressions (betas, tstats)

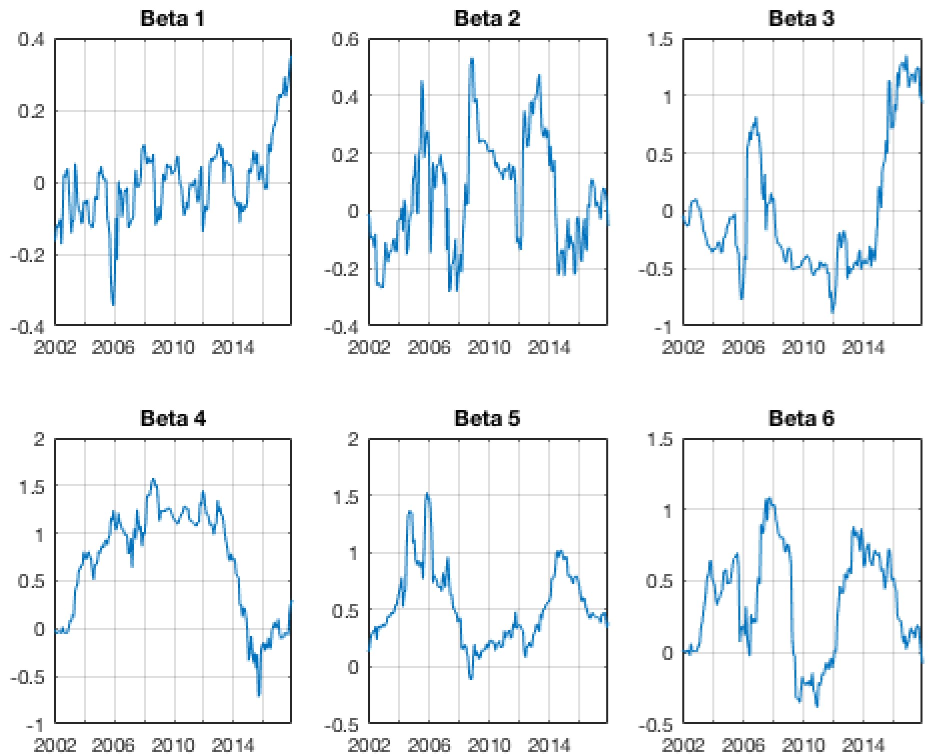
```
for t = 36:T
    b(:,t) = inv(X(t-35:t,:)'*X(t-35:t,:))*X(t-35:t,:)'*Y(t-35:t);
    yhat = X(t-35:t,:)*b(:,t);
    ESS = sum((yhat-Y(t-35:t)).^2);
    SSb = ESS/(T-k)*inv(X(t-35:t,:)'*X(t-35:t,:));
    sb = sqrt(diag(SSb));
    tstats(:,t) = (b(:,t)./sb);
end
```

Calculate Returns Attribution across factors

```
temp = b'.*X; % Individual factor returns
alpha = Y-sum(temp,2); % intercept plus residual (specific) return
ReturnsAtt =
 table(date,Y,alpha,temp(:,1),temp(:,2),temp(:,3),temp(:,4),temp(:,5),temp(:,6),'V

 {'Date','Rport','alpha','realRates','inflation','credit','economic','EM','liquidi
ReturnsAtt = ReturnsAtt(36:end,:); % looks at the 36 month window
```

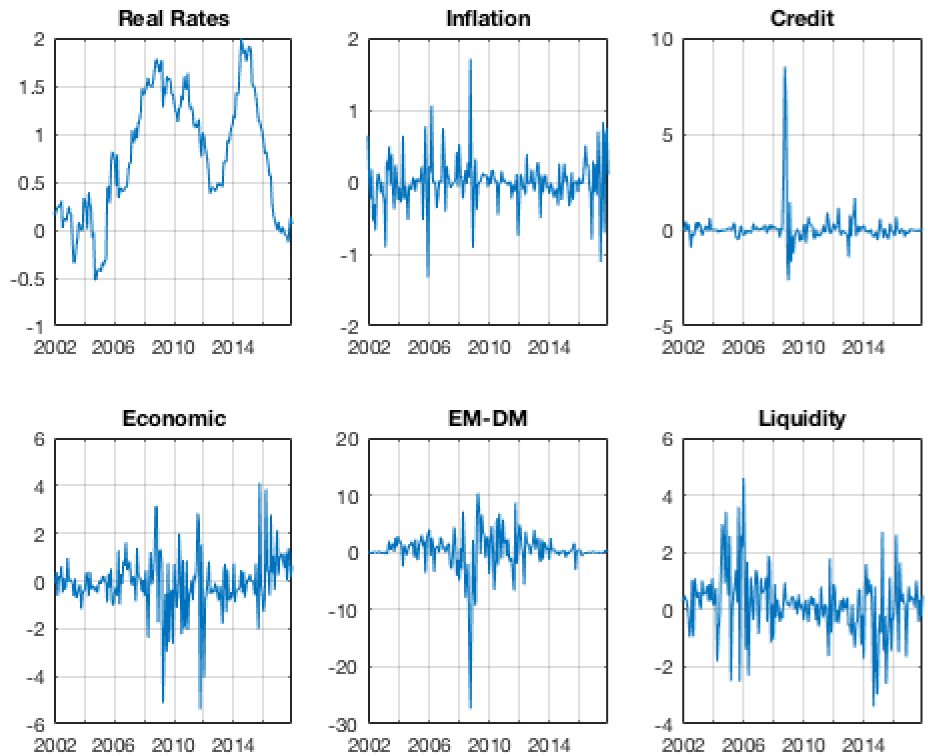
Time Series of Betas to the Black Rock Factors

```
subplot(2,3,1); plot(date(36:end),b(2,36:end)); title('Beta 1');
 grid on;
subplot(2,3,2); plot(date(36:end),b(3,36:end)); title('Beta 2');
 grid on;
subplot(2,3,3); plot(date(36:end),b(4,36:end)); title('Beta 3');
 grid on;
subplot(2,3,4); plot(date(36:end),b(5,36:end)); title('Beta 4');
 grid on;
subplot(2,3,5); plot(date(36:end),b(6,36:end)); title('Beta 5');
 grid on;
subplot(2,3,6); plot(date(36:end),b(7,36:end)); title('Beta 6');
 grid on;
```



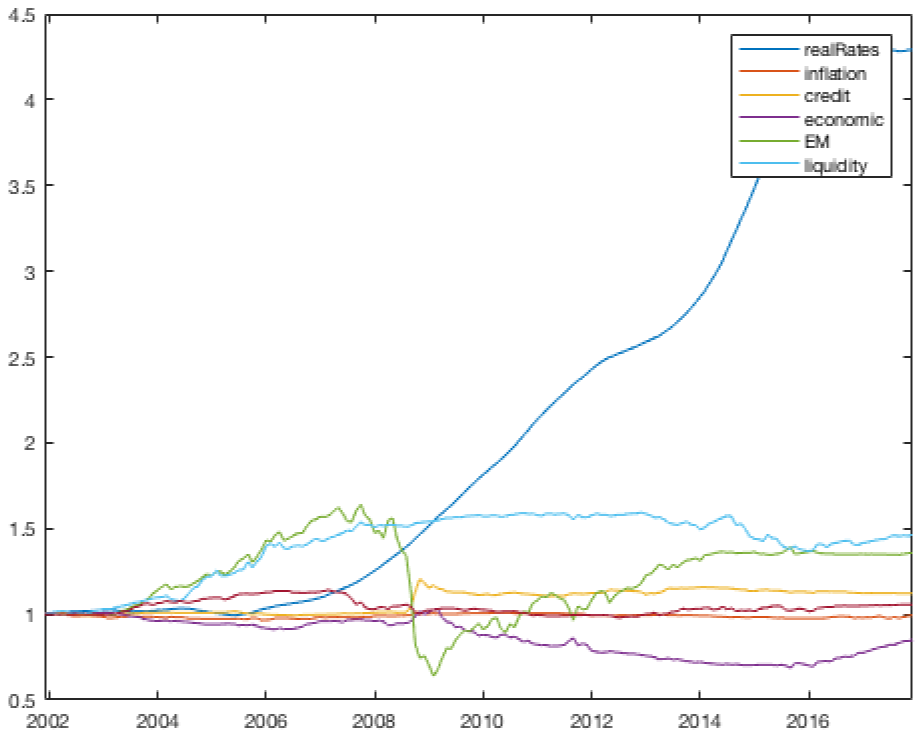
Sample Attribution Time Series Plots

```
subplot(2,3,1); plot(ReturnsAtt.Date,ReturnsAtt.realRates);
 title('Real Rates'); grid on;
subplot(2,3,2); plot(ReturnsAtt.Date,ReturnsAtt.inflation);
 title('Inflation'); grid on;
subplot(2,3,3); plot(ReturnsAtt.Date,ReturnsAtt.credit);
 title('Credit'); grid on;
subplot(2,3,4); plot(ReturnsAtt.Date,ReturnsAtt.economic);
 title('Economic'); grid on;
subplot(2,3,5); plot(ReturnsAtt.Date,ReturnsAtt.EM); title('EM-DM');
 grid on;
subplot(2,3,6); plot(ReturnsAtt.Date,ReturnsAtt.liquidity);
 title('Liquidity'); grid on;

% Note: interpretation on plots is for the previous 3-year period
```



Plot Cumulative Returns Contributions by Factor through time

```
CumulativeReturn = cumprod(1+temp(36:end,:)/100);
handleFigure = figure;
plot(date(36:end),CumulativeReturn(:,1:7));legend('realRates','inflation','credit'

clear;clc;
```



Monte Carlo Simulation

Import Country Data

```
country = readtable('/Users/nickbruno/Documents/ECON 4360/Project/
newdataset.xlsx','Sheet','Country');
date = datetime(country.Date);
M = [country.us, country.italy, country.australia, country.brazil,
 country.india, country.southAfrica]/100;
selectedVariables = {'US';'Italy';'Australia';'Brazil';'India';'South
 Africa'};

[T,k] = size(M);
mean_return = (prod(1+M).^(1/T)-1); % do we multiply by 100 here

nAssets = numel(mean_return); r = mean(mean_return); % number of
 assets and desired return
Aeq = ones(1,nAssets); beq = 1; % equality Aeq*x = beq
Aineq = -mean_return; bineq = -1.5*r; % inequality Aineq*x <= bineq
lb = zeros(nAssets,1); ub = ones(nAssets,1); % bounds lb <= x <= ub
c = zeros(nAssets,1); % objective has no linear term; set it to zero
% Select Options for Quadprog
options = optimset('Algorithm','interior-point-convex');
options = optimset(options,'Display','off','TolFun',1e-10);

for t = 36:T
```

```
 V = cov(M(t-35:t,:));
 Vcov(:,:,t) = V; Vcond(t) = cond(V);
end
    % calculates the Covariance Matrix
% Base Case
Vbase = Vcov(:,:,end); % Data from December 2017

% Stress Case
Vstress = Vcov(:,:,117); % Data from September 2008
```

Monte Carlo Simulation, Comparing Base and Stress Case

```
r = (100*(prod(1+M/100).^(1/T) -1 )); % Calculates expected monthly
 returns
N = 1:1000;
T = 60;
arithmetic_base = r + 0.5*Vbase;
artithmetic_stress = r + 0.5*Vstress;

s = [1, 5, 10];
x = diag(arithmetic_base);

% Base Case Monte Carlo simulation
for n = 1:1000
    R = (repmat(mean_return - 0.5*diag(Vbase)', 60, 1) +
 randn(60,6)*chol(Vbase));
    [wtemp1,fval1] = (quadprog(Vbase,c,Aineq,bineq,Aeq,beq,lb,ub,
[],options));  %targeted return port
    [wtemp2,fval2] = (quadprog(Vbase,c,[],[],Aeq,beq,lb,ub,
[],options));  %Min Var portfolio
    wrp = (rsk_parity(mean_return', Vbase, lb, ub, 0));
    wtarg = (wtemp1);
    wmv = (wtemp2);
    wrp = (wrp);
    Rp_mv(n) = (prod(1+R*wmv)^((12/60) - 1)); % finds minimum variance
    Rp_targ(n) = (prod(1+R*wtarg)^((12/60) - 1)); % finds target
    Rp_rp(n) = (prod(1+R*wrp)^((12/60) - 1)); % finds risk parity
end

base = [Rp_mv',Rp_targ',Rp_rp']; % creates a 1000x3 matrix

Varbase = prctile(base, s); % returns percentiles

VarbaseTable = table(Varbase(:,1), Varbase(:,2), Varbase(:,3),...,
    'VariableNames',{'Minimum_Variance','Target','Risk_Parity'},...,
    'RowNames', {'VAR 1%','VAR 5%','VAR 10%'})

% Stress Case Monte Carlo Simulation
for n = 1:1000
    R = repmat(mean_return - 0.5*diag(Vstress)',60,1) +
 randn(60,6)*chol(Vstress);
```

```
    [wtemp1,fval1] = quadprog(Vstress,c,Aineq,bineq,Aeq,beq,lb,ub,
[],options);  %targeted return port
    [wtemp2,fval2] = quadprog(Vstress,c,[],[],Aeq,beq,lb,ub,
[],options);  %Min Var portfolio
    wrp = rsk_parity(mean_return', Vstress, lb, ub, 0);
    wtarg = wtemp1;
    wmv = wtemp2;
    wrp = wrp;
    Rp_mv(n) = prod(1+R*wmv)^((12/60) - 1); % find minimum variance
    Rp_targ(n) = prod(1+R*wtarg)^((12/60) - 1); % finds target
    Rp_rp(n) = prod(1+R*wrp)^((12/60) - 1); % finds risk parity
end

stress = [Rp_mv',Rp_targ',Rp_rp']; % creates a 1000x3 matrix

Varstress = prctile(stress, s); % returns percentiles

VarstressTable = table(Varstress(:,1), Varstress(:,2),
 Varstress(:,3),...,
    'VariableNames',{'Minimum_Variance','Target','Risk_Parity'},...,
    'RowNames', {'VAR 1%','VAR 5%','VAR 10%'})

% VarstressTable shows that there was a higher minimum variance for
 each
% Value at Risk level in September 2008 compared to the data from
 December
% 2017. The tables also show that there is a higher risk parity in the
% December 2017 data compared to September 2008.

clear;clc;
```

```
VarbaseTable =

  3×3 table

               Minimum_Variance    Target     Risk_Parity
               ________________    _______    ___________

    VAR 1%     0.49573             0.39991    0.47786
    VAR 5%     0.54622             0.44475     0.5535
    VAR 10%    0.57484             0.48041    0.58172


VarstressTable =

  3×3 table

               Minimum_Variance    Target     Risk_Parity
               ________________    _______    ___________

    VAR 1%     0.50473             0.32089    0.40744
    VAR 5%     0.58714             0.39529    0.51245
    VAR 10%    0.63149              0.4381    0.55137
```

Option Straddle

Import Country Data

```
country = readtable('/Users/nickbruno/Documents/ECON 4360/Project/
newdataset.xlsx','Sheet','Country');
date = datetime(country.Date);
M = [country.us, country.italy, country.australia, country.brazil,
 country.india, country.southAfrica]/100;
selectedVariables = {'US';'Italy';'Australia';'Brazil';'India';'South
 Africa'};

[T,k] = size(M);
mean_return = (prod(1+M).^(1/T)-1);

% Option Straddle Simulation
r = mean(mean_return) + 0.05; % do I add 0.05 here?
sigma = [0.05, 0.1, 0.15, 0.2]; % volatility levels
S = 100;
n = 1000;

for vol = sigma(:,:)
    for t=1:n
        P(t,:) = S * exp(r - 0.5*(sigma.^2) + (sigma * randn *
 sqrt(1)));
        call = max(P - S, 0)/(1 + r);
        put = max(S - P, 0)/(1 + r);
    end
        average_call = mean(call);
        average_put = mean(put);
        std_call = std(call);
        std_put = std(put);
        skew_call = skewness(call);
        skew_put = skewness(put);

   if vol == sigma(1)
        vol_five_percent = table(average_call', std_call',...,
            skew_call', average_put',std_put', skew_put',...,
            'VariableNames',
{'average_call','std_call','skew_call',...,
            'average_put','std_put','skew_put'},'RowNames',{'VAR
 5%',...,
            'VAR 10%','VAR 15%','VAR 20%'})
   end

   if vol == sigma(2)
        vol_ten_percent = table(average_call', std_call',...,
            skew_call', average_put',std_put', skew_put',...,
            'VariableNames',
{'average_call','std_call','skew_call',...,
```

```
            'average_put','std_put','skew_put'},'RowNames',{'VAR
 5%',...,
            'VAR 10%','VAR 15%','VAR 20%'})
   end

   if vol == sigma(3)
        vol_fifteen_percent = table(average_call', std_call',...,
            skew_call', average_put',std_put', skew_put',...,
            'VariableNames',
{'average_call','std_call','skew_call',...,
            'average_put','std_put','skew_put'},'RowNames',{'VAR
 5%',...,
            'VAR 10%','VAR 15%','VAR 20%'})
   end

   if vol == sigma(4)
        vol_twenty_percent = table(average_call', std_call',...,
            skew_call', average_put',std_put', skew_put',...,
            'VariableNames',
{'average_call','std_call','skew_call',...,
            'average_put','std_put','skew_put'},'RowNames',{'VAR
 5%',...,
            'VAR 10%','VAR 15%','VAR 20%'})
   end

end

% This code prints four tables, the first looking at the volatility
 level
% at 5%, second with volatility at 10%, third with volatility at 15%,
 and
% the last table represents the statistics with volatility at 20%.

clear;clc;
```

vol_five_percent =

4×6 table

	average_call	*std_call*	*skew_call*	*average_put*	*std_put*	*skew_put*
VAR 5%	*5.7543*	*4.6036*	*0.712*	*0.28204*	*0.95623*	*4.33*
VAR 10%	*7.0777*	*8.0894*	*1.3067*	*1.7607*	*3.4614*	*2.2551*
VAR 15%	*8.7614*	*11.6*	*1.6532*	*3.5941*	*5.9542*	*1.7592*
VAR 20%	*10.542*	*15.314*	*1.9148*	*5.5173*	*8.3299*	*1.5001*

vol_ten_percent =

4×6 table

	average_call	std_call	skew_call	average_put	std_put	skew_put
VAR 5%	5.9388	4.6654	0.78843	0.27452	1.0054	4.5723
VAR 10%	7.3558	8.3357	1.4094	1.6493	3.4638	2.519
VAR 15%	9.135	12.076	1.786	3.3745	5.9161	1.9728
VAR 20%	11.024	16.074	2.0803	5.1948	8.2526	1.6841

vol_fifteen_percent =

4×6 table

	average_call	std_call	skew_call	average_put	std_put	skew_put
VAR 5%	5.8915	4.589	0.65241	0.35317	1.176	4.1936
VAR 10%	7.338	8.0235	1.3075	1.877	3.8325	2.3963
VAR 15%	9.0848	11.528	1.7159	3.686	6.4353	1.9149
VAR 20%	10.938	15.237	2.0625	5.5863	8.881	1.6545

vol_twenty_percent =

4×6 table

	average_call	std_call	skew_call	average_put	std_put	skew_put
VAR 5%	5.9887	4.4768	0.52796	0.32894	1.166	4.4226
VAR 10%	7.3871	7.8817	1.0523	1.6964	3.7336	2.6561
VAR 15%	9.1231	11.249	1.3436	3.4011	6.2429	2.1321

```
    VAR 20%      10.95           14.75       1.5577       5.1985
 8.6135      1.838
```

Dynamic Delta Hedge

Set parameters

```
Kp = 100;
Kc = 115;
sigma = 0.18;
sigma_index = 0.15;
rho = rand;
r = 0.02;
Week = [0:52]';
u1 = 0.07;
u2 = 0.06;
arithmetic_mean = [u1 + (sigma^2)/2, u2 + (sigma^2)/2];

% Set Time
Time = (repmat(52,length(Week),1)-Week)/52;
Time(53) = (52 - 51.5)/52; % ensures that the last value for time does
 not equal zero

% Creates empty tables to append to later
put_values = [];
call_values = [];
pnl = [];

% Creating C, S, and V
C = [1, rho; rho,1];
S = [sigma, 0; 0, sigma_index];
V = S'*C*S;

% Conducting the Monte Carlo Simulation
for n = 1:100

    M =[100,100; zeros(52,2)];
    for t = 2:length(Time)
        M(t, :) = M(t-1,:).*exp(arithmetic_mean/52 +
 randn(1,2)*chol(V)/sqrt(52));
    end

    % Generate d1c, d2c, d1p, and d2p
    d1c = (log(M(:,2)/
Kc)+repmat(r,length(Time),1)+0.5*repmat(sigma_index^2,length(Time),1).*Time)./
(sigma_index*sqrt(Time));
    d2c = d1c - (repmat(sigma_index, length(Time),1).*sqrt(Time));
    d1p = (log(M(:,2)/
Kp)+repmat(r,length(Time),1)+0.5*repmat(sigma_index^2,length(Time),1).*Time)./
(sigma_index*sqrt(Time));
    d2p = d1p - (repmat(sigma_index,length(Time),1).*sqrt(Time));
```

Price Options and Computing the Delta

```
    put = Kp*exp(-r*Time).*normcdf(-d2p) - M(:,2).*normcdf(-d1p);
    call = M(:,2).*normcdf(d1c) - exp(-r*Time)*Kc.*normcdf(d1c-
sigma_index*sqrt(Time));
    delta = normcdf(d1c);
```

Delta Hedging

```
    X = [delta(1), 100*delta(1), (100*delta(1)*(r/52)); zeros(52,3)];
    for t = 2:length(Time)
        X(t,1) = delta(t) - delta(t-1);
        X(t,2) = X(t-1,2) + (X(t,1)*M(t,2))+X(t-1,3);
        X(t,3) = X(t,2)*(r/52);
    end
```

Delta Hedge Account Table

Setting Index, Stock, shares purchased, and interest cost based off of the values from X and M

```
    Index = M(:,1);
    Stock = M(:,2);
    shares_purch = X(:,1);
    cost = X(:,2);
    interest_cost = X(:,3);

    % Creating the table
    DeltaHedgeAcct = table(Week, Time, Index, Stock, d1c, d2c, d1p,
 d2p, put, call, delta, shares_purch, cost, interest_cost);
```

Finding put, call, and percent gain/loss values

Find put value

```
    put_value = max(Kp-Stock(53),0)-put(1);
    put_values = [put_values; put_value];

    % Find call value
    hedge = cost(53);
    call_value = (Stock(53) > Kc)*Kc - hedge + call(1);
    call_values = [call_values ; call_value];

    % Find net profit
    port_profit = Index(53)-Index(1)+call_value+put_value;
    net_pl = ((Index(53)-Index(1)+put_value+call_value)/Index(1));
    pnl = [pnl; net_pl];

end
```

Evaluating the mean put, mean call, and mean profit

```
mean_put = mean(put_values);
mean_call = mean(call_values);
mean_pnl = mean(pnl);

Delta_Hedge_Account_Table = DeltaHedgeAcct(:,:)
    % Shows one of the DeltaHedgeAcct tables from the 100 created in
 the
    % earlier for-loop
```

Delta_Hedge_Account_Table =

53×14 table

Week	Time	Index	Stock	d1c	d2c	d1p	d2p	put	call	delta	shares_purch	cost	interest_cost
0	1	100	100	-0.72341	-0.87341	0.20833	0.058333	4.9817	1.9166	0.23471	0.23471	23.471	0.0090274
1	0.98077	101.62	99.398	-0.77255	-0.9211	0.16828	0.019733	5.1995	1.7285	0.21989	-0.01482	22.007	0.0084643
2	0.96154	102.97	99.019	-0.80774	-0.95482	0.14246	-0.0046255	5.328	1.5975	0.20962	-0.010272	20.999	0.0080764
3	0.94231	106.74	101.89	-0.62097	-0.76658	0.33887	0.19326	4.1169	2.221	0.26731	0.057688	26.885	0.01034
4	0.92308	101.76	98.91	-0.83503	-0.97915	0.13476	-0.009357	5.2985	1.4777	0.20185	-0.06546	20.42	0.007854
5	0.90385	97.777	98.653	-0.86358	-1.0062	0.11648	-0.026131	5.3751	1.3798	0.19391	-0.0079399	19.645	0.0075557
6	0.88462	97.222	95.018	-1.1406	-1.2817	-0.14994	-0.29102	7.1992	0.77324	0.12702	-0.066889	13.297	0.0051142
7	0.86538	96.152	94.331	-1.2068	-1.3463	-0.20518	-0.34471	7.5626	0.66023	0.11376	-0.013261	12.051	0.004635
8	0.84615	98.782	91.708	-1.4263	-1.5643	-0.41338	-0.55136	9.1821	0.39442	0.076892	-0.036868	8.6746	0.0033364

9	0.82692	104.51	90.676	-1.5274	-1.6638
-0.50273	-0.63914	9.8655	0.30454	0.063336	
-0.013556	7.4487	0.0028649			
10	0.80769	101.47	89.892	-1.6115	-1.7463
-0.57472	-0.70953	10.405	0.24324	0.053538	
-0.0097976	6.5709	0.0025273			
11	0.78846	100.54	89.409	-1.673	-1.8062
-0.62372	-0.75691	10.742	0.20457	0.04716	
-0.0063783	6.0031	0.0023089			
12	0.76923	99.536	90.302	-1.6199	-1.7515
-0.55758	-0.68913	10.067	0.23009	0.052624	
0.0054639	6.4988	0.0024995			
13	0.75	96.501	89.184	-1.7381	-1.868
-0.66224	-0.79214	10.877	0.16676	0.041094	
-0.011529	5.4731	0.002105			
14	0.73077	98.794	86.506	-2.0003	-2.1285
-0.91036	-1.0386	12.995	0.080035	0.022734	
-0.018361	3.8868	0.0014949			
15	0.71154	97.365	89.479	-1.7618	-1.8883
-0.65723	-0.78376	10.618	0.15077	0.039051	
0.016317	5.3484	0.0020571			
16	0.69231	92.686	90.468	-1.6998	-1.8246
-0.57999	-0.70479	9.8588	0.17379	0.044584	
0.0055331	5.851	0.0022504			
17	0.67308	90.306	89.156	-1.8444	-1.9675
-0.7087	-0.83177	10.832	0.11592	0.032562	
-0.012022	4.7814	0.001839			
18	0.65385	94.822	90.223	-1.775	-1.8963
-0.62269	-0.74398	9.9964	0.13666	0.037951	
0.0053894	5.2695	0.0020267			
19	0.63462	90.432	88.061	-2.0064	-2.1259
-0.83683	-0.95632	11.676	0.070534	0.022405	
-0.015547	3.9025	0.001501			
20	0.61538	90.506	85.059	-2.3342	-2.4519
-1.1465	-1.2641	14.235	0.025778	0.0097926	
-0.012612	2.8312	0.0010889			
21	0.59615	91.112	84.83	-2.3967	-2.5125
-1.1899	-1.3057	14.444	0.020589	0.0082725	
-0.0015201	2.7034	0.0010398			
22	0.57692	91.971	84.262	-2.4971	-2.6111
-1.2704	-1.3844	14.965	0.014447	0.0062601	
-0.0020125	2.5348	0.00097494			
23	0.55769	94.516	84.776	-2.4875	-2.5995
-1.2398	-1.3518	14.506	0.014463	0.0064323	
0.00017229	2.5504	0.00098093			
24	0.53846	92.815	85.552	-2.4507	-2.5608
-1.181	-1.2911	13.812	0.015794	0.0071282	
0.00069583	2.6109	0.0010042			
25	0.51923	90.6	88.897	-2.1429	-2.251
-0.84983	-0.95791	10.915	0.039998	0.016061	
0.008933	3.406	0.00131			
26	0.5	96.769	87.434	-2.3421	-2.4482
-1.0245	-1.1305	12.147	0.020899	0.0095866	
-0.0064746	2.8413	0.0010928			

27	0.48077	98.639	85.403	-2.6166	-2.7206	-1.2728	-1.3768	13.964	0.0081357	0.0044407	-0.0051459	2.4029	0.00092418	
28	0.46154	97.324	84.606	-2.7646	-2.8665	-1.3931	-1.495	14.711	0.004647	0.0028494	-0.0015913	2.2692	0.00087276	
29	0.44231	96.756	86.471	-2.6077	-2.7075	-1.2067	-1.3065	13.001	0.0076864	0.0045574	0.001708	2.4177	0.0009299	
30	0.42308	101.35	85.8	-2.7484	-2.846	-1.3159	-1.4135	13.622	0.0044684	0.0029943	-0.0015631	2.2845	0.00087867	
31	0.40385	103.57	87.64	-2.5928	-2.6881	-1.1266	-1.2219	11.947	0.0073164	0.0047606	0.0017663	2.4402	0.00093855	
32	0.38462	101.2	88.622	-2.5394	-2.6324	-1.037	-1.13	11.069	0.0083051	0.0055528	0.00079217	2.5114	0.00096591	
33	0.36538	104.46	88.177	-2.6633	-2.7539	-1.1218	-1.2125	11.459	0.0050698	0.0038694	-0.0016833	2.3639	0.00090919	
34	0.34615	104.68	87.888	-2.7759	-2.8642	-1.1922	-1.2805	11.719	0.0031424	0.0027524	-0.001117	2.2666	0.00087179	
35	0.32692	105.16	89.424	-2.6569	-2.7426	-1.0273	-1.113	10.332	0.0044734	0.0039437	0.0011912	2.374	0.00091309	
36	0.30769	105.45	91.56	-2.4574	-2.5406	-0.77771	-0.86091	8.4753	0.0082842	0.0069966	0.0030529	2.6545	0.001021	
37	0.28846	106.01	89.002	-2.8925	-2.973	-1.1577	-1.2382	10.695	0.0015181	0.001911	-0.0050855	2.2029	0.00084726	
38	0.26923	104.2	86.209	-3.4064	-3.4842	-1.6107	-1.6885	13.335	0.00014705	0.00032911	-0.0015819	2.0673	0.00079513	
39	0.25	106.99	86.064	-3.5604	-3.6354	-1.6969	-1.7719	13.497	5.4796e-05	0.00018514	-0.00014398	2.0558	0.00079067	
40	0.23077	107.53	84.262	-4.0024	-4.0744	-2.0628	-2.1349	15.295	1.9652e-06	3.1354e-05	-0.00015378	2.0436	0.00078599	
41	0.21154	107.63	85.443	-3.9818	-4.0508	-1.956	-2.025	14.157	-7.5026e-07	3.4193e-05	2.8393e-06	2.0446	0.00078639	
42	0.19231	112.28	87.036	-3.8986	-3.9643	-1.7739	-1.8396	12.613	-4.635e-06	4.8381e-05	1.4188e-05	2.0466	0.00078717	
43	0.17308	114.36	87.993	-3.9378	-4.0002	-1.6982	-1.7606	11.697	-8.5801e-06	4.1117e-05	-7.2641e-06	2.0468	0.00078722	
44	0.15385	113.06	90.116	-3.775	-3.8338	-1.3995	-1.4583	9.6457	-2.2322e-05	8.0002e-05	3.8885e-05	2.0511	0.00078887	

```
   45        0.13462    113.32    92.887     -3.4893     -3.5444
 -0.94983       -1.0049    7.0314    -7.9408e-05     0.0002421
0.0001621     2.0669    0.00079497
   46        0.11538    112.72    92.641     -3.8251     -3.8761
  -1.0822       -1.1331    7.2314    -3.5406e-05     6.535e-05
-0.00017676     2.0513    0.00078898
   47       0.096154    105.95    90.043     -4.8065      -4.853
  -1.8017       -1.8482    9.7658    -6.3468e-07    7.6805e-07
-6.4582e-05     2.0463    0.00078704
   48       0.076923    103.76    87.909     -5.9556     -5.9972
  -2.5962       -2.6378    11.935    -1.3567e-09    1.2952e-09
-7.6675e-07      2.047    0.00078732
   49       0.057692    108.21    83.689     -8.2482     -8.2842
   -4.369        -4.405    16.195     -9.903e-17    8.0421e-17
-1.2952e-09     2.0478    0.00078762
   50       0.038462     108.7    83.847     -10.045     -10.075
  -5.2943       -5.3237    16.076    -6.6632e-24    4.8173e-24
-8.0421e-17     2.0486    0.00078793
   51       0.019231    104.84    82.228     -15.154     -15.175
   -8.435       -8.4558    17.734    -5.4067e-52     3.569e-52
-4.8173e-24     2.0494    0.00078823
   52      0.0096154    105.73    85.884     -18.481     -18.496
  -8.9788       -8.9935    14.097    -2.4306e-76    1.4741e-76
-3.569e-52     2.0502    0.00078853
```

Creating a table summarizing the mean put, mean call, and mean gain/loss

```
summary_statistics = table(mean_put, mean_call, mean_pnl)
```

```
summary_statistics = 

  1×3 table

    mean_put    mean_call    mean_pnl 
    ________    _________    _________

    -1.1648     -7.2889      0.0071955
```

Published with MATLAB® R2017b