

MPRA

Munich Personal RePEc Archive

Multiperiod Black Litterman Asset Allocation Model

Karan Malhotra

December 2012

Online at <http://mpra.ub.uni-muenchen.de/44806/>

MPRA Paper No. 44806, posted 8. March 2013 02:28 UTC

Multiperiod Black Litterman Asset Allocation Model

Karan Malhotra

STAT 240P

Professor Tze Lai, Stanford University

Data Setup

To make the computation easier and more transparent I exported all the data from the text file into excel and performed basic data cleanup in excel. The following things were done in excel

- I subtracted the risk free rate of return (10 year monthly treasury rates) from the returns given to obtain the excess returns for each time period for each portfolio. **Columns 2:7**
- The risk free rate of return for each time period is also present in the excel backup. The risk free rate of return is later added back to the equilibrium risk premiums obtained to get the predicted return for that asset class (equilibrium return). **Columns 34**
- The predictive variables (explained in detail later) were lagged by 1 time period for the regression. **Columns 8:27**
- AR1 and AR2 terms for each of the six portfolios was introduced as predictive variables in excel. **Columns 16:27**
- Market weights of each portfolio were calculated in excel for each time period. This was performed as follows –
a) get market weight as (number of firms x average firm size) for each portfolio. Then add all the market weights for all the portfolios. And then divide the market weight for that asset class by the total to get the %weight. **Columns 28:33**
- We lose observations due to the AR terms and the analysis period starts in April 1992.

Below is a snapshot of the excel file for reference purposes, with all the 34 columns.

DATE	SMLD	SMME	SMHI	BILD	BIME	BHIL	VIX	IDYLIST	OIL	AAA	BAA	IPY	GBP	SPX	SMLD AR1	SMME AR1	SMHI AR1	BILD AR1	BIME AR1	BHIL AR1	SMLD AR2	SMME AR2	SMHI AR2	BILD AR2	BIME AR2	BHIL AR2	WSMLD	WSMME	WSMHI	WBILD	WBIME	WBHIL	RF	
30-Apr-92	-0.07	-0.03	-0.02	-0.01	0.04	0.03	-0.03	0.04	0.04	0.01	0.00	0.03	-0.01	-0.02	-0.07	-0.02	-0.02	-0.04	-0.02	-0.02	0.00	0.02	0.06	-0.01	0.00	0.06	0.04	0.04	0.03	0.49	0.32	0.09	0.006	
29-May-92	-0.02	0.00	0.02	0.00	-0.01	-0.01	-0.04	0.01	0.07	0.00	0.00	0.00	0.02	0.03	-0.07	-0.03	-0.02	-0.01	0.04	0.03	-0.07	-0.02	-0.02	-0.04	-0.02	-0.02	0.03	0.04	0.03	0.48	0.34	0.09	0.006	
30-Jun-92	-0.07	-0.04	-0.04	-0.03	-0.02	0.00	-0.11	-0.03	0.06	-0.01	-0.01	-0.04	0.03	0.00	-0.02	0.00	0.02	0.00	-0.01	-0.01	-0.07	-0.03	-0.02	-0.01	0.04	0.03	0.03	0.04	0.03	0.48	0.33	0.09	0.006	
31-Jul-92	0.02	0.02	0.03	0.03	0.04	0.01	-0.04	-0.03	-0.02	-0.01	-0.01	-0.01	0.04	-0.02	-0.07	-0.04	-0.04	-0.03	-0.02	0.00	-0.02	0.00	0.02	0.00	-0.01	-0.01	0.04	0.03	0.02	0.43	0.36	0.12	0.005	
31-Aug-92	-0.05	-0.03	-0.04	-0.02	-0.03	-0.06	-0.01	-0.06	0.01	-0.02	-0.02	0.01	0.01	0.04	0.02	0.02	0.03	0.03	0.04	0.01	-0.07	-0.04	-0.04	-0.03	-0.02	0.00	0.04	0.03	0.02	0.43	0.36	0.12	0.005	
30-Sep-92	0.00	0.01	0.01	0.01	0.00	0.00	0.03	-0.02	-0.02	-0.01	-0.02	-0.03	0.03	-0.02	-0.05	0.03	-0.04	-0.02	-0.03	-0.06	0.02	0.02	0.03	0.04	0.01	0.04	0.03	0.02	0.44	0.36	0.11	0.005		
30-Oct-92	0.03	0.01	0.01	-0.01	-0.02	0.00	0.05	-0.04	0.01	0.00	0.00	-0.02	-0.11	0.01	0.00	0.01	0.01	0.01	0.00	0.00	-0.05	-0.03	-0.04	-0.02	-0.03	-0.06	0.04	0.03	0.02	0.44	0.36	0.11	0.005	
30-Nov-92	0.09	0.05	0.06	0.04	0.02	0.04	0.13	0.07	-0.05	0.01	0.03	0.03	-0.12	0.00	0.03	0.01	0.01	0.01	0.01	0.00	0.00	0.00	0.01	0.01	0.00	0.00	0.04	0.03	0.02	0.45	0.35	0.11	0.006	
31-Dec-92	0.02	0.03	0.04	0.00	0.01	0.02	-0.19	0.02	-0.04	0.01	0.01	0.01	0.01	-0.03	0.03	0.09	0.05	0.06	0.04	0.02	0.04	0.03	0.01	0.01	0.01	-0.02	0.00	0.04	0.03	0.02	0.45	0.34	0.11	0.005
29-Jan-93	0.00	0.03	0.05	-0.02	0.00	0.04	-0.03	-0.04	-0.02	-0.01	-0.02	0.00	0.00	0.01	0.02	0.03	0.04	0.00	0.01	0.02	0.09	0.05	0.06	0.04	0.02	0.04	0.04	0.03	0.02	0.44	0.34	0.11	0.005	
28-Feb-93	-0.06	0.00	0.01	-0.02	0.04	0.03	-0.01	-0.05	0.04	-0.01	-0.02	0.00	-0.02	0.01	0.01	0.04	0.06	-0.01	0.01	0.05	0.03	0.04	0.05	0.01	0.02	0.03	0.04	0.03	0.02	0.43	0.35	0.12	0.005	
31-Mar-93	0.02	0.03	0.04	0.02	0.03	0.03	0.06	-0.05	0.01	-0.03	-0.03	-0.05	-0.04	0.01	-0.06	0.00	0.01	-0.02	0.04	0.03	0.01	0.04	0.06	-0.01	0.01	0.05	0.04	0.03	0.02	0.42	0.36	0.12	0.005	
30-Apr-93	-0.04	-0.02	-0.03	-0.05	-0.01	-0.01	-0.05	0.00	0.00	-0.02	-0.03	-0.03	0.06	0.02	0.02	0.03	0.04	0.02	0.03	0.03	-0.06	0.00	0.01	-0.02	0.04	0.03	0.04	0.03	0.03	0.42	0.36	0.12	0.005	

Part A

Aim: Calculation of Equilibrium Risk Premia

Code: Note the loop is incomplete in the code pasted below. The code only illustrates the calculation of the equilibrium risk premiums. The complete code is pasted in the Appendix.

```
# Final Project STAT 240P Karan Malhotra
# Read original data along with predictive variables
orig_data<-read.csv("C:/Users/RS/Desktop/final_project_raw_data_v2.csv", header=TRUE, stringsAsFactors=FALSE)

#declare libraries
library(MASS)

#declare matrix that will contain the predicted returns
predicted_mat=matrix(0,178,6)
#declare matrix that will contain the equilibrium risk premia
```

```

eq_riskprem_mat=matrix(0,178,6)
# declare matrix that will contain optimum portfolio weights, after maximizing for sharpe ratio across the efficient
frontier
w_opt<-matrix(0,178,6)
# matrix containing the dates on which the predictions are made
predicted_vec_date=matrix(0,178,1)
# declare risk aversion factor
risk_av<-2.5
weff<-matrix(0,178,6)

for (i in 1:178){

# Specify start and end row of the first 5 year window
start_row=i
end_row=i+59

# y_var is a matrix containing excess returns of the six portfolios in a rolling 5 year window
y_var=as.matrix(orig_data[start_row:end_row,2:7])
# x_var is a matrix containing the factors on which the predictive regression is going to be run in a rolling 5 year window
x_var=as.matrix(orig_data[start_row:end_row,8:27])
# W is a matrix containing the market weights of the six portfolios at time t
W<-as.matrix(orig_data[end_row,28:33])
#risk free rate of return
rf<-orig_data[i,34]

# degrees of freedom used in calculation of the error term of the predicted y- variable (used in the matrix omega in BL)
dof<-nrow(x_var)-ncol(x_var)-1

# To compute the covariance matrix an expanding window of data is used (instead of a 5 year rolling window)
y_var_exp<-as.matrix(orig_data[1:end_row, 2:7])
cov_y_var_exp<-cov(y_var_exp)
# Calculate the equilibrium risk premium for the excess returns
eq_riskprem<-t(risk_av*(cov_y_var_exp%*%t(W)))
# Add back the risk free rate of return to get the equilibrium risk premiums
eq_riskprem_mat[i,]<-eq_riskprem+rf
}

```

Comparison of Historical Risk Premium and Equilibrium Risk Premium:

Both historical risk premium (average of the excess returns using an expanding window, starting with 5 years) and equilibrium risk premiums (expanding window starting with 5 years of data) are shown below in the sample output. It is observed that generally the equilibrium risk premium is higher in value and also more stable than the historical risk premium. Note the output below is truncated:

	Equilibrium Risk Premium						Historical Risk Premium					
	SMLO	SMME	SMHI	BILO	BIME	BIHI	SMLO	SMME	SMHI	BILO	BIME	BIHI
30-Apr-97	0.84%	0.75%	0.75%	0.80%	0.77%	0.75%	0.10%	0.82%	1.04%	0.64%	0.95%	0.69%
30-May-97	0.83%	0.75%	0.75%	0.80%	0.77%	0.75%	0.04%	0.81%	1.01%	0.73%	0.99%	0.70%
30-Jun-97	0.86%	0.76%	0.76%	0.81%	0.77%	0.75%	0.28%	0.95%	1.12%	0.82%	1.07%	0.78%
31-Jul-97	0.76%	0.67%	0.66%	0.71%	0.68%	0.66%	0.33%	1.03%	1.20%	0.87%	1.12%	0.80%
29-Aug-97	0.77%	0.68%	0.67%	0.73%	0.69%	0.67%	0.39%	1.08%	1.27%	0.97%	1.24%	0.88%
30-Sep-97	0.76%	0.67%	0.67%	0.74%	0.70%	0.67%	0.44%	1.10%	1.29%	0.86%	1.16%	0.81%
31-Oct-97	0.77%	0.68%	0.68%	0.74%	0.71%	0.68%	0.56%	1.20%	1.38%	0.91%	1.23%	0.88%
28-Nov-97	0.88%	0.79%	0.78%	0.84%	0.81%	0.78%	0.46%	1.13%	1.32%	0.85%	1.16%	0.82%
31-Dec-97	0.77%	0.68%	0.68%	0.75%	0.71%	0.68%	0.40%	1.09%	1.30%	0.90%	1.18%	0.84%
30-Jan-98	0.76%	0.68%	0.67%	0.74%	0.71%	0.68%	0.36%	1.08%	1.30%	0.89%	1.20%	0.88%
27-Feb-98	0.76%	0.68%	0.67%	0.74%	0.71%	0.68%	0.35%	1.03%	1.26%	0.92%	1.15%	0.86%
31-Mar-98	0.78%	0.69%	0.68%	0.75%	0.72%	0.69%	0.45%	1.11%	1.34%	1.00%	1.24%	0.94%
30-Apr-98	0.77%	0.69%	0.68%	0.75%	0.72%	0.70%	0.49%	1.16%	1.38%	1.05%	1.28%	1.03%
29-May-98	0.77%	0.68%	0.68%	0.75%	0.72%	0.70%	0.49%	1.16%	1.38%	1.04%	1.29%	1.01%
30-Jun-98	0.77%	0.69%	0.68%	0.75%	0.72%	0.69%	0.38%	1.07%	1.32%	0.98%	1.24%	1.01%
31-Jul-98	0.77%	0.68%	0.68%	0.75%	0.71%	0.69%	0.39%	1.04%	1.29%	1.04%	1.24%	1.03%
31-Aug-98	0.68%	0.59%	0.58%	0.65%	0.62%	0.59%	0.27%	0.91%	1.18%	1.01%	1.18%	0.95%
30-Sep-98	0.80%	0.70%	0.68%	0.73%	0.72%	0.66%	-0.05%	0.65%	0.94%	0.81%	0.92%	0.79%
30-Oct-98	0.82%	0.70%	0.68%	0.74%	0.72%	0.66%	0.06%	0.72%	0.96%	0.89%	0.96%	0.86%
30-Nov-98	0.93%	0.81%	0.78%	0.86%	0.83%	0.77%	0.12%	0.75%	0.98%	0.98%	1.04%	0.90%
31-Dec-98	0.94%	0.81%	0.79%	0.86%	0.83%	0.77%	0.23%	0.80%	1.04%	1.05%	1.10%	0.94%
29-Jan-99	0.94%	0.81%	0.79%	0.87%	0.83%	0.77%	0.30%	0.82%	1.06%	1.14%	1.11%	0.96%
26-Feb-99	0.94%	0.81%	0.78%	0.87%	0.83%	0.77%	0.36%	0.82%	1.04%	1.20%	1.09%	0.93%
31-Mar-99	0.95%	0.81%	0.79%	0.88%	0.82%	0.76%	0.24%	0.72%	0.93%	1.13%	1.06%	0.89%
30-Apr-99	1.04%	0.91%	0.88%	0.97%	0.92%	0.86%	0.23%	0.71%	0.87%	1.16%	1.09%	0.89%
28-May-99	1.03%	0.90%	0.88%	0.96%	0.92%	0.86%	0.34%	0.81%	0.96%	1.16%	1.18%	0.96%
30-Jun-99	1.04%	0.91%	0.88%	0.97%	0.92%	0.86%	0.34%	0.82%	0.99%	1.11%	1.15%	0.95%
30-Jul-99	1.04%	0.91%	0.88%	0.97%	0.92%	0.86%	0.41%	0.87%	1.05%	1.17%	1.16%	0.92%
31-Aug-99	1.04%	0.90%	0.88%	0.98%	0.92%	0.85%	0.39%	0.85%	1.02%	1.12%	1.11%	0.87%
30-Sep-99	1.04%	0.91%	0.88%	0.98%	0.92%	0.85%	0.35%	0.80%	0.97%	1.10%	1.06%	0.83%
29-Oct-99	1.03%	0.90%	0.87%	0.97%	0.92%	0.85%	0.35%	0.77%	0.92%	1.06%	0.97%	0.76%
30-Nov-99	1.03%	0.89%	0.87%	0.98%	0.92%	0.86%	0.35%	0.75%	0.87%	1.12%	1.02%	0.80%
31-Dec-99	1.03%	0.90%	0.87%	0.98%	0.92%	0.86%	0.48%	0.80%	0.89%	1.15%	0.98%	0.77%
31-Jan-00	1.04%	0.90%	0.87%	0.98%	0.92%	0.85%	0.67%	0.88%	0.94%	1.22%	0.99%	0.78%
29-Feb-00	1.04%	0.90%	0.87%	0.99%	0.92%	0.86%	0.67%	0.87%	0.92%	1.14%	0.93%	0.72%
31-Mar-00	1.04%	0.90%	0.87%	0.99%	0.92%	0.86%	0.96%	1.01%	1.03%	1.14%	0.87%	0.63%
28-Apr-00	1.01%	0.89%	0.86%	1.00%	0.93%	0.87%	0.80%	0.95%	1.01%	1.21%	0.97%	0.75%
31-May-00	0.93%	0.80%	0.77%	0.91%	0.84%	0.78%	0.64%	0.88%	0.95%	1.15%	0.95%	0.77%
30-Jun-00	0.92%	0.79%	0.77%	0.91%	0.84%	0.77%	0.55%	0.81%	0.90%	1.11%	0.94%	0.76%
31-Jul-00	0.92%	0.79%	0.77%	0.90%	0.83%	0.76%	0.72%	0.89%	0.97%	1.14%	0.87%	0.70%

Below shows the average over time of the historical and equilibrium risk premiums and their SD over time. Equilibrium risk premium is certainly more stable over time.

	Equilibrium Risk Premium						Historical Risk Premium					
	SMLO	SMME	SMHI	BILO	BIME	BIHI	SMLO	SMME	SMHI	BILO	BIME	BIHI
Mean	0.95%	0.79%	0.78%	0.86%	0.81%	0.77%	0.29%	0.88%	1.00%	0.59%	0.74%	0.60%
SD	0.10%	0.08%	0.08%	0.10%	0.08%	0.08%	0.16%	0.12%	0.15%	0.29%	0.25%	0.19%

Part B

Aim: Build a model to obtain the view vector (predicted returns) at time $t+1$, along with the confidence of the view

Code: Note the code below is in continuation of the code above. I use a linear multifactor model to predict returns at time $t+1$. A rolling 5 year window was used for the linear regression to avoid the betas suffering from a structural change in relationship between the response variable and the explanatory variables. The following macroeconomic factors were used in the linear regression for each:

1. Return on the VIX Index
2. Return on the 10 year US treasury rate
3. Return on an Oil Index
4. Return on AAA Bond rates
5. Return on BBB Bond rates
6. Return on Japanese Yen
7. Return on the British Pound
8. Return on the SP-500 Index

In addition AR1 and AR2 terms for each of the 6 portfolios were used as regressors

In order to make the linear regression predictive, the macroeconomic variable returns were lagged by 1 time period. Therefore $Y(t)$ was regressed against $X(t-1)$. The coefficients obtained were multiplied by $X(t)$ (the value of the macroeconomic variables at time t) to obtain an estimate of $Y(t+1)$. The variance of the predicted return was obtained consistent with out of sample variance of a fitted value in linear regression.

```
#Omega to store the variances of the error terms of the predicted return
omega<-diag(6)
# P is the projection matrix
P<-diag(6)
# Tau is the calculated as below, decreasing in value as time window expands
tau<-1/(nrow(y_var_exp))

for (j in 1:6){

# Perform linear regression on all factors
initial_model<-lm(y_var[,j]~x_var[,1:20])
initial_coefs<-t(as.matrix(coef(initial_model)))

# Get data from the explanatory variables that will be multiplied with the coefficients of the regression fitted
x_input<-rbind(1,t(as.matrix(orig_data[end_row+1,8:27])))

# This is the predicted return for 1 asset in 1 time period
predicted_ret<-(initial_coefs%*%x_input)
date_predicted_ret<-orig_data[end_row+1,1]
y<-y_var[,j]
x<-cbind(1,x_var[,1:20])
eye<-diag(nrow(x))

# Compute the out of sample fitted value variance (error)
```

```

err_pred_ret<-((t(y)%*(eye-
x%*(solve(t(x)%*x))%*(t(x))%*y)/dof)*(t(x_input)%*solve((t(x)%*x))%*x_input)

# Store the error in the omega matrix to be used later in the BL process
omega[j,j]<-err_pred_ret

predicted_mat[i,j]<-predicted_ret
predicted_vec_date[i,1]<-date_predicted_ret

#end j
}

```

Part C

Aim: Complete the Black Litterman asset allocation process

Code: Note the code is in continuation of the code above. The comments in the code below explain the procedure step by step. First I calculate the mean returns vector and covariance matrix according to the Black Litterman formula. Subsequently, I write a function that generates portfolio return, standard deviation, sharpe ratio and portfolio weights given a particular target return. I run this function in a loop from 0 to 2% monthly target return and obtain the optimum portfolio, for which the Sharpe ratio across the efficient frontier is maximized. I multiply these optimum weights with the actual return of the six portfolios to get the realized returns of the BL asset allocation strategy.

Also, I take the log returns of these realized returns and then do a cumulative sum to get the cumulative return on this strategy (after converting the log returns back to simple returns)

```

# Compute the estimate of the mean returns according to the master Black Litterman formula
UBL<-
(solve(solve(tau*cov_y_var_exp)+t(P)%*(solve(omega))%*P))%*(((solve(tau*cov_y_var_exp))%*t(eq_riskprem))+
t(P)%*(solve(omega))%*predicted_mat[i,])
# Compute the estimate of the covariance matrix according to the master Black Litterman formula
CBL<-cov_y_var_exp+(solve(solve(tau*cov_y_var_exp)+t(P)%*(solve(omega))%*P))

ones<-as.matrix(rep(1,length(UBL)))

tUBL<-t(UBL)
invCBL<-solve(CBL)

A<-as.numeric(tUBL%*invCBL%*ones)
B<-as.numeric(tUBL%*invCBL%*UBL)
C<-as.numeric((t(ones))%*invCBL%*ones)
D<-B*C-A^2

# Define a function that inputs the target return and outputs portfolio return, st-dev of the return, sharpe ratio, and the
port. weights
funcports <- function (target){
weff<-(B*(invCBL%*ones)-A*(invCBL%*UBL)+target*(C*(invCBL%*UBL)-A*(invCBL%*ones)))/D
portret<-tUBL%*weff
vareff<-(B-2*target*A+(target^2)*C)/D
stdeff<-vareff^0.5
}

```

```

sharpe<-(portret/stdeff)*(sqrt(12))
output<-c(portret,stdeff, sharpe, t(weff))
return (output)
}

rets <- seq(0,0.02,by=0.0001)
final<-matrix(0,length(rets),9)

# Run a loop that generates the efficient frontier and stores all the output in a matrix
for (g in 1:length(rets)){
final[g,1]<-funcports(rets[g])[1]
final[g,2]<-funcports(rets[g])[2]
final[g,3]<-funcports(rets[g])[3]
final[g,4:9]<-funcports(rets[g])[4:9]
end
}

# Get the portfolio weights for which the sharp ratio is maximized
opt_index<-which.max(final[,3])
w_opt[i,]<-final[opt_index, 4:9]

# end i
}

# act_ret is the matrix of the actual returns on the six portfolios
act_ret<-orig_data[61:238,2:7]

# Declare a matrix ret_real that will contain the realized returns of the BL asset allocation process by multiplying port.
weights with the actual returns
ret_real<-matrix(0,178,1)
for (h in 1:178){
ret_real[h,1]<-w_opt[h,]%*%t(act_ret[h,])
}

# Cumulative Returns #

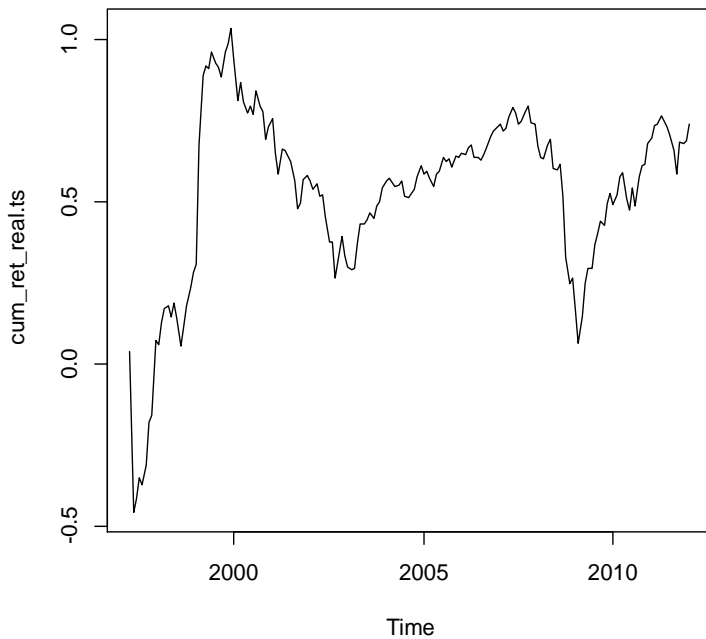
# Convert the returns to log return to support additivity
ret_real_ln<-log(1+ret_real)

# Do a cumulative sum on the log returns to get the total return over time
cum_ret_real<-cumsum(ret_real_ln)

#Gives you the cumulative return on the strategy from April 1997 to Jan 2012
str_percent_ret<-exp(cum_ret_real[length(cum_ret_real)])-1

```


The cumulative log returns of the strategy are pasted below:



The Sharpe ratio of the strategy was computed to be **0.3260312**. This was slightly better than an equally weighted portfolio which produced a sharpe ratio of **0.3107669**.

In comparison, the Markowitzian optimization using expanding window starting from 5 years to compute the mean and variance of the portfolio resulted in a sharpe ratio of **0.9336546**. The reason MVO has a higher sharpe ratio than black litterman is that MVO produces extreme corner solutions and does not produce a balanced portfolio. Also, since the optimization is unconstrained the MVO solution ends up producing extreme weights as shown below as example. The MVO method always shorts 1st, 5th and 6th portfolio by huge amounts and goes long the other portfolios.

```
[1,] -1.1649676 0.9374615 1.199678943 0.4781608 0.42149583 -0.8718295
[2,] -1.0790842 0.8997424 1.141743754 0.4044966 0.40865975 -0.7755582
[3,] -1.0698081 0.9540912 1.112375598 0.3792171 0.42786901 -0.8037448
[4,] -1.0139947 0.9926896 0.915171368 0.3935455 0.33067202 -0.6180838
[5,] -0.9938194 1.0878185 0.783887637 0.3965308 0.21502652 -0.4894439
[6,] -0.9758717 1.1371054 0.824041540 0.2919846 0.25818954 -0.5354494
[7,] -0.9503966 1.1442489 0.742120555 0.3117732 0.20779701 -0.4555431
[8,] -0.9837378 1.1423194 0.794066482 0.3381829 0.17742465 -0.4682556
[9,] -0.9908857 1.1158223 0.805885285 0.3676416 0.16008881 -0.4585523
[10,] -0.9816328 1.1155372 0.783843775 0.3646151 0.14748022 -0.4298435
[11,] -1.0401079 0.9690868 1.004870067 0.5071709 0.02532739 -0.4663473
[12,] -1.0175005 1.0014159 0.925966399 0.5082165 -0.01847951 -0.3996188
[13,] -0.9937726 1.0118085 0.873512185 0.5035721 -0.03022440 -0.3648958
[14,] -0.9847423 1.0201693 0.851751026 0.4992058 -0.02581443 -0.3605693
[15,] -1.0190162 0.8228218 1.060997733 0.4981515 -0.04839613 -0.3145587
[16,] -1.0294004 0.6854065 1.175282890 0.5717331 -0.07239792 -0.3306241
[17,] -1.0504914 0.4296337 1.367000028 0.6737616 0.01127328 -0.4311772
[18,] -1.3253928 0.4973327 1.567019426 0.9714272 -0.60206688 -0.1083197
[19,] -1.4012056 0.6521158 1.490019610 1.0597091 -0.69647999 -0.1041588
[20,] -1.3902296 0.6426671 1.447763477 1.0653974 -0.63239921 -0.1331992
[21,] -1.3459035 0.5207800 1.516078736 1.0630612 -0.59190061 -0.1621158
[22,] -1.3183336 0.5308501 1.467414179 1.0713319 -0.62274219 -0.1285204
[23,] -1.3298078 0.5440668 1.467267275 1.1638432 -0.66661922 -0.1787503
[24,] -1.4262067 0.5038296 1.418845804 1.2049648 -0.43848969 -0.2629438
[25,] -1.4849004 0.6941399 1.196519612 1.2677775 -0.33026949 -0.3432671
[26,] -1.4363129 0.7179354 1.135961811 1.1154920 -0.20358232 -0.3294940
[27,] -1.4483694 0.6291043 1.299861488 1.0923254 -0.20881848 -0.3641033
[28,] -1.4356893 0.6302454 1.278327963 1.0796101 -0.21408898 -0.3384053
[29,] -1.4896425 0.6611086 1.357771247 1.1054550 -0.22134307 -0.4133493
```

Part D

Aim: Compute the turnover rate and compare it Markowitzian optimization

Code: Note the code is in continuation of the code above. The comments in the code below explain the procedure step by step.

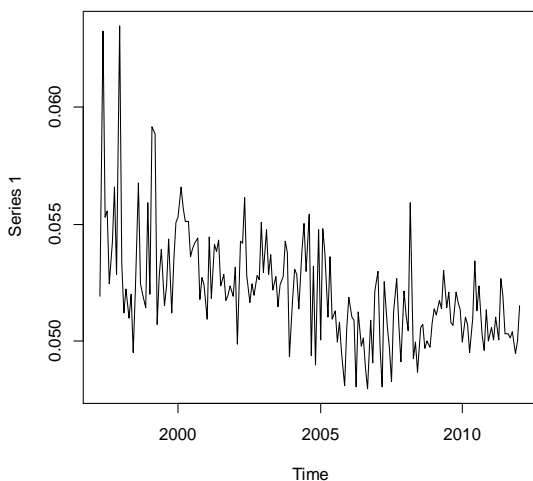
```
## Part D - Strategy Turnover ##

start_amt<-matrix(1,1,6)
start_m<-matrix(0,178,1)
turn_over_BL<-matrix(0,178,1)

for (y in 1:178){
dollars_t = start_amt * w_opt[y,]
returns_t=dollars_t*orig_data[60+i, 2:7]
dollars_tp1=dollars_t+returns_t
turn_over_tp1= sum(abs(dollars_tp1 - dollars_t))/sum(abs(dollars_tp1))
turn_over_BL[y]=turn_over_tp1
start_amt=start_amt*(1+ret_real[y,])
}

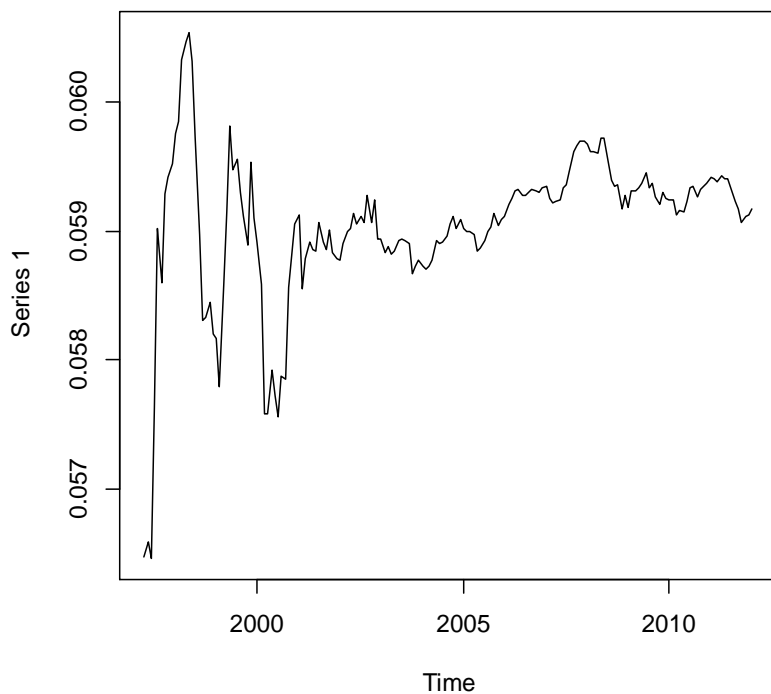
turn_over_BL.ts = ts (turn_over_BL, start=c(1997, 4), frequency=12)
plot(turn_over_BL.ts)
```

The turnover rate against time is shown below for the BL asset allocation method:



In comparison, the turnover of the Markowitz mean-variance optimization was as below. The reason the plain Markowitz optimization results in a lower change in turnover rate is because the MVO method produces corner solutions and does not change allocations over time from these corner solutions. The MVO method always shorts 1st, 5th and 6th portfolio by huge amounts and goes long the other portfolios.

[1,]	-1.1649676	0.9374615	1.199678943	0.4781608	0.42149583	-0.8718295
[2,]	-1.0790842	0.8997424	1.141743754	0.4044966	0.40865975	-0.7755582
[3,]	-1.0698081	0.9540912	1.112375598	0.3792171	0.42786901	-0.8037448
[4,]	-1.0139947	0.9926896	0.915171368	0.3935455	0.33067202	-0.6180838
[5,]	-0.9938194	1.0878185	0.783887637	0.3965308	0.21502652	-0.4894439
[6,]	-0.9758717	1.1371054	0.824041540	0.2919846	0.25818954	-0.5354494
[7,]	-0.9503966	1.1442489	0.742120555	0.3117732	0.20779701	-0.4555431
[8,]	-0.9837378	1.1423194	0.794066482	0.3381829	0.17742465	-0.4682556
[9,]	-0.9908857	1.1158223	0.805885285	0.3676416	0.16008881	-0.4585523
[10,]	-0.9816328	1.1155372	0.783843775	0.3646151	0.14748022	-0.4298435
[11,]	-1.0401079	0.9690868	1.004870067	0.5071709	0.02532739	-0.4663473
[12,]	-1.0175005	1.0014159	0.925966399	0.5082165	-0.01847951	-0.3996188
[13,]	-0.9937726	1.0118085	0.873512185	0.5035721	-0.03022440	-0.3648958
[14,]	-0.9847423	1.0201693	0.851751026	0.4992058	-0.02581443	-0.3605693
[15,]	-1.0190162	0.8228218	1.060997733	0.4981515	-0.04839613	-0.3145587
[16,]	-1.0294004	0.6854065	1.175282890	0.5717331	-0.07239792	-0.3306241
[17,]	-1.0504914	0.4296337	1.367000028	0.6737616	0.01127328	-0.4311772
[18,]	-1.3253928	0.4973327	1.567019426	0.9714272	-0.60206688	-0.1083197
[19,]	-1.4012056	0.6521158	1.490019610	1.0597091	-0.69647999	-0.1041588
[20,]	-1.3902296	0.6426671	1.447763477	1.0653974	-0.63239921	-0.1331992
[21,]	-1.3459035	0.5207800	1.516078736	1.0630612	-0.59190061	-0.1621158
[22,]	-1.3183336	0.5308501	1.467414179	1.0713319	-0.62274219	-0.1285204
[23,]	-1.3298078	0.5440668	1.467267275	1.1638432	-0.66661922	-0.1787503
[24,]	-1.4262067	0.5038296	1.418845804	1.2049648	-0.43848969	-0.2629438
[25,]	-1.4849004	0.6941399	1.196519612	1.2677775	-0.33026949	-0.3432671
[26,]	-1.4363129	0.7179354	1.135961811	1.1154920	-0.20358232	-0.3294940
[27,]	-1.4483694	0.6291043	1.299861488	1.0923254	-0.20881848	-0.3641033
[28,]	-1.4356893	0.6302454	1.278327963	1.0796101	-0.21408898	-0.3384053
[29,]	-1.4896425	0.6611086	1.357771247	1.1054550	-0.22134307	-0.4133493
[30,]	-1.5256899	0.6930569	1.356163920	1.1750940	-0.29258675	-0.4060382
[31,]	-1.6249608	1.0652597	1.183794473	1.2975202	-0.49936645	-0.4222472
[32,]	-1.5357263	1.0484281	0.929257946	1.2908651	-0.41088049	-0.3219443
[33,]	-1.5104148	1.3225281	0.731554094	1.3692723	-0.57862483	-0.3343149
[34,]	-1.4059179	1.4051354	0.576217340	1.3690791	-0.66884658	-0.2756673
[35,]	-1.4459404	1.4520716	0.676233643	1.3731568	-0.67075873	-0.3847630
[36,]	-1.4086739	1.5302053	0.816781372	1.3978520	-0.68155814	-0.6546066
[37,]	-1.4060485	1.5293671	0.813852172	1.3947370	-0.68192038	-0.6499875
[38,]	-1.4351481	1.5952417	0.759405559	1.3582340	-0.80741682	-0.4703164



Appendix – Complete Code for BL asset allocation method

```
# Final Project STAT 240P Karan Malhotra

# Read original data along with predictive variables
orig_data<-read.csv("C:/Users/RS/Desktop/final_project_raw_data_v2.csv", header=TRUE, stringsAsFactors=FALSE)

#declare libraries
library(MASS)

#declare matrix that will contain the predicted returns
predicted_mat=matrix(0,178,6)
#declare matrix that will contain the equilibrium risk premia
eq_riskprem_mat=matrix(0,178,6)
# declare matrix that will contain optimum portfolio weights, after maximizing for sharpe ratio across the efficient frontier
w_opt<-matrix(0,178,6)
# matrix containing the dates on which the predictions are made
predicted_vec_date=matrix(0,178,1)
# declare risk aversion factor
risk_av<-2.5
weff<-matrix(0,178,6)

for (i in 1:178){

# Specify start and end row of the first 5 year window
start_row=i
end_row=i+59

# y_var is a matrix containing excess returns of the six portfolios in a rolling 5 year window
y_var=as.matrix(orig_data[start_row:end_row,2:7])
# x_var is a matrix containing the factors on which the predictive regression is going to be run in a rolling 5 year window
x_var=as.matrix(orig_data[start_row:end_row,8:27])
# W is a matrix containing the market weights of the six portfolios at time t
W<-as.matrix(orig_data[end_row,28:33])
#risk free rate of return
rf<-orig_data[i,34]

# degrees of freedom used in calculation of the error term of the predicted y- variable (used in the matrix omega in BL)
dof<-nrow(x_var)-ncol(x_var)-1

# To compute the covariance matrix an expanding window of data is used (instead of a 5 year rolling window)
y_var_exp<-as.matrix(orig_data[1:end_row, 2:7])
cov_y_var_exp<-cov(y_var_exp)
# Calculate the equilibrium risk premium for the excess returns
eq_riskprem<-t(risk_av*(cov_y_var_exp%*%t(W)))
# Add back the risk free rate of return to get the equilibrium risk premiums
eq_riskprem_mat[i,]<-eq_riskprem+rf
```

```

#Omega to store the variances of the error terms of the predicted return
omega<-diag(6)
# P is the projection matrix
P<-diag(6)
# Tau is the calculated as below, decreasing in value as time window expands
tau<-1/(nrow(y_var_exp))

for (j in 1:6){

# Perform linear regression on all factors
initial_model<-lm(y_var[,j]~x_var[,1:20])

initial_coefs<-t(as.matrix(coef(initial_model)))
# Get data from the explanatory variables that will be multiplied with the coefficients of the regression fitted
x_input<-rbind(1,t(as.matrix(orig_data[end_row+1,8:27])))
# This is the predicted return for 1 asset in 1 time period
predicted_ret<-(initial_coefs%%x_input)
date_predicted_ret<-orig_data[end_row+1,1]
y<-y_var[,j]
x<-cbind(1,x_var[,1:20])
eye<-diag(nrow(x))
# Compute the out of sample fitted value variance (error)
err_pred_ret<-((t(y)%%(eye-
x%%(solve(t(x)%%x))%%t(x))%%y)/dof)*(t(x_input)%%solve((t(x)%%x))%%x_input)
# Store the error in the omega matrix to be used later in the BL process
omega[j,j]<-err_pred_ret

predicted_mat[i,j]<-predicted_ret
predicted_vec_date[i,1]<-date_predicted_ret

#end j
}

# Compute the estimate of the mean returns according to the master Black Litterman formula
UBL<-
(solve(solve(tau*cov_y_var_exp)+t(P)%%(solve(omega))%%P))%%(((solve(tau*cov_y_var_exp))%%t(eq_riskprem))+
t(P)%%(solve(omega))%%predicted_mat[i,])
# Compute the estimate of the covariance matrix according to the master Black Litterman formula
CBL<-cov_y_var_exp+(solve(solve(tau*cov_y_var_exp)+t(P)%%(solve(omega))%%P))

ones<-as.matrix(rep(1,length(UBL)))

tUBL<-t(UBL)
invCBL<-solve(CBL)

A<-as.numeric(tUBL%%invCBL%%ones)
B<-as.numeric(tUBL%%invCBL%%UBL)
C<-as.numeric((t(ones))%%invCBL%%ones)
D<-B*C-A^2

# Define a function that inputs the target return and outputs portfolio return, st-dev of the return, sharpe ratio, and the
port. weights
funcports <- function (target){

```

```

weff<-(B*(invCBL%%ones)-A*(invCBL%%UBL)+target*(C*(invCBL%%UBL)-A*(invCBL%%ones)))/D
portret<-tUBL%%weff
vareff<-(B-2*target*A+(target^2)*C)/D
stdeff<-vareff^0.5
sharpe<-(portret/stdeff)*(sqrt(12))
output<-c(portret,stdeff, sharpe, t(weff))
return (output)
}

rets <- seq(0,0.02,by=0.0001)
final<-matrix(0,length(rets),9)

# Run a loop that generates the efficient frontier and stores all the output in a matrix
for (g in 1:length(rets)){
final[g,1]<-funcports(rets[g])[1]
final[g,2]<-funcports(rets[g])[2]
final[g,3]<-funcports(rets[g])[3]
final[g,4:9]<-funcports(rets[g])[4:9]
end
}

# Get the portfolio weights for which the sharp ratio is maximized
opt_index<-which.max(final[,3])
w_opt[i,<-final[opt_index, 4:9]

# end i
}

# act_ret is the matrix of the actual returns on the six portfolios
act_ret<-orig_data[61:238,2:7]

# Declare a matrix ret_real that will contain the realized returns of the BL asset allocation process by multiplying port.
weights with the actual returns
ret_real<-matrix(0,178,1)
for (h in 1:178){
ret_real[h,1]<-w_opt[h,]%%t(act_ret[h,])
}

# Cumulative Returns #

# Convert the returns to log return to support additivity
ret_real_ln<-log(1+ret_real)

# Do a cumulative sum on the log returns to get the total return over time
cum_ret_real<-cumsum(ret_real_ln)

#Gives you the cumulative return on the strategy from April 1997 to Jan 2012
str_percent_ret<-exp(cum_ret_real[length(cum_ret_real)])-1

cum_ret_real.ts<-ts(cum_ret_real, start=c(1997,4), frequency=12)
plot(cum_ret_real.ts)

### Part D - Strategy Turnover ###

```

```
start_amt<-matrix(1,1,6)
start_m<-matrix(0,178,1)
turn_over_BL<-matrix(0,178,1)

for (y in 1:178){
dollars_t = start_amt * w_opt[y,]
returns_t=dollars_t*orig_data[60+i, 2:7]
dollars_tp1=dollars_t+returns_t
turn_over_tp1= sum(abs(dollars_tp1 - dollars_t))/sum(abs(dollars_tp1))
turn_over_BL[y]=turn_over_tp1
start_amt=start_amt*(1+ret_real[y,])
}

turn_over_BL.ts = ts (turn_over_BL, start=c(1997, 4), frequency=12)
plot(turn_over_BL.ts)
```