**Code Hierarchy**

**\*\***The code can be downloaded from https://github.com/seyonv/MIE479

Instructions on how to run(to see graphs):

1. Navigate to Capstone-MATLAB

2. Add to Path(Selected Folders and Subfolders) MS\_REGRESS\_FEX\_1.08, Symbol\_Files, MainProgramFiles

3. In CSV\_Files, add nyse\_SP to path

4. Open “a.m” and modify parameters as desired. The comments above the file explain the parameters

and how to choose them

5. run “a.m”

Instructions on how to run if you want to see all MATLAB workspace variables(for detailed examination)

1. Complete instructions #1 to #3 in “Instructions on how to run(to see graphs)”

2. In **a.m**, comment out all the current code, and uncomment the line “run create\_inflation\_hedged\_portfolio.m”

3. In **create\_inflation-hedged\_portfolio.m**, delete the function header and its corresponding end(at the very end). Proceed to uncomment the section at the beginning that says to uncomment it and define variables as desired.

-----------------------------------------------------------------------------------------------------

**(CODE) Means insert the code**

(CODE) a.m – This is the file the user runs. The user defines the input parameters include time periods, desired return and which ticker symbols to use

(CODE) create\_inflation\_hedged\_portfolio.m – This is the main program file where all operations are executed. It calls the appropriate data fetching functions, conducts MLE to generate the RegimeSwitching model, constructs the markov tree for expected inflation and conditional variance of inflation.

It then fetches data again for out-of-sample data and test the model in comparison with standard MVO, S/&P 500 and two mutual funds. The cumulative return results are plotted(graphs for rebalanced portfolio and non-rebalanced portfolio

(NO CODE) NYSE\_symbols.csv: This is a ticker list of all the stocks on the NYSE(it was used to help fetch csv files of individual stocks

(NO CODE) riskfreerate2.csv: This has monthly data on the 3-month treasury bill rate(which is used as the risk free rate in sharpe ratio /& modified sharpe ratio computations.

(NO CODE) unemployment\_rate.csv: This has monthly data on the US unemployment rate. It was not used in the current iteration of this project but may be include as part of the regime switching model as a linear term.

(NO CODE) SP500tickernames.xlsx: Ticker list of all the current S/&P500 stocks.

-----------------------------------------------------------------------------------------------------

CSV\_Files: This folder contains all the CSV files used for obtaining asset data. Note that in the current iteration, only nyse\_SP is used

->inf\_funds: This has time series data for particular funds and ETF’s whose goal is to provide inflation-protection. Many of these funds invest up to 80% of holdings into TIPS

->inflation\_assets:These are assets that are known to have a high inflation beta and correlate well with inflation. They are generally part of industries such as oil, gold and real estate.

->nyse\_SP: This folder has csv files for NYSE stocks that date back to 1970 at the least as well as all S/&P500 stocks

->S/&P500 reformatted: This folder has csv files for S/&P500 stock

-----------------------------------------------------------------------------------------------------

Images\_For\_reference: This folder contains various images used for reference. It contains images for rebalanced/non-rebalanced portfolios and the regime switching image. It also saves the matlab workspace associated with generating each of the figures. The folder also has a picture demonstrating a sample

RS model with 3 regime and a figure showing historical inflation rates.

-----------------------------------------------------------------------------------------------------

MS\_Regress\_FEX\_1.08 – This is the package created by Marcelo Perlin. It contains example files on how to create different regime switching models. The

MS\_Regress\_MSVAR function is used where MSVAR stands for Markov Switching Vector AutoRegressive. The function returns transition probabilities and the mean, autoregressive term and variance associated with each regime

-----------------------------------------------------------------------------------------------------

Symbol\_Files: This folder contains files who act as a ticker list. The ticker list has data for every stock ticker in the file fetched from the CSV\_Files folder. There are two matrices present in the file. One contains the names of the stocks while the other contains the names of the corresponding stock CSV files(simply stockname.csv).

Note that; the last three assets in each file are LOMMX(CGM Mutual Fund) ,VWELX,(Vanguard Wellington Investor Shares), ^GPSC(S/&P500)

->infsymbols.m: has ticker list corresponding to asset that move the same way as inflation

->secondSymbols.m: a small subset, of 25 stocks, used for testing

->Symbols\_NYSE\_SP.m: has stocks on the NYSE and the S/&P 500. Duplicates are not present

->Symbols\_NYSE.m: has only stocks symbols on the NYSE

->Symbols\_SP.m: : has only stocks on the S/&P500

->Symbols.m: Identical Copy to Symbols\_SP

->(CODE)thirdSymbols.m: a very small subset, of 3 stocks, used for testing. Lower in the appendix, the code is shown for this

to display the format of the files, but to not take up so much space by displaying many ticker symbols.

-----------------------------------------------------------------------------------------------------

MainProgramFiles

-ManualComputation\_Inflationbeta.xlsx: This was just an excel file created where regression was applied to compute inflation betas. Because the MATLAB polyfit function was used in the main code, this was used a comparison to ensure that the results matched up and the Beta’s were compute correctly

-(CODE)RegimeSwitching\_MLE.m: This function is used to compute the parameters that define the regime switching. It takes in the number of regimes to define and the inflation data as input parameters. It defines the presence of an intercept, an autoregressive term and the noise term as being normally distributed with mean 0 and variance /sigma^2.It finds the sum of the log likelihood and using the procedure documented above, returns the values of the mean, autoregressive and variance term for each regime, along with the transition probabilities of going from one regime to another

-----------------------------------------------------------------------------------------------------

(In MainProgramFiles)

Fetch\_Functions: This folder contains all functions that fetch data from CSV files or from within existing MATLAB vectors. Note that all data used is organized from most recent to oldest so that the row number for a particular date is consistent among all CSV files.

-(CODE) all\_inflation\_data.m- This retrieves all the historical inflation data (used for computing inflation betas, not RS model). It’s returned in the form or

infmonth, infyear and infprice so that it’s easy to fetch inflation data between a particular time period by looking at infmonth and infyear.

-(CODE) all\_riskfree\_data.m- Similar to all\_inflation\_data but for the 3-month treasury bill prices

-(CODE) all\_stock\_data.m- Similar to all\_inflation\_data but also returns fails\_symbols and success\_symbols so stocks csv files that don’t

work or don’t exist can be examined.

-(CODE) fetch\_inflation\_data2.m- Takes in the variables return by all\_inflation\_data and a desired time period and returns the prices associated

with that time period as well as 2 index variables representing where in the infprice vector, data was fetched from.

-(CODE) fetch\_regime\_inflation\_data.m- This fetches inflation data from either the monthly, quarterly or annual spreadsheet. This data is used to compute the RS model

-(CODE) fetch\_riskfree\_data2.m- Similar to fetch\_inflation\_data2 but for 3-month treasury bill prices

-(CODE) SEC\_fetch\_stock\_data.m- Similar to fetch\_inflation\_data2 but for all the stock prices. It also returns the stock names that have data that go back as early as the passed start date. It returns the marketprice as a separate vector.

-----------------------------------------------------------------------------------------------------

(In MainProgramFiles)

Inflation\_data:

->inflation\_data\_annual: annual inflation data (used for RS model computation)

->inflation\_data\_quarterly: quarterly inflation data (used for RS model computation)

->inflation\_data\_monthly: monthly inflation data (used for RS model computation)

->inflation\_data\_1200: This is the file used to compute inflation betas. It is the same as inflation\_data\_monthly but it is organized differently.

-----------------------------------------------------------------------------------------------------

(In MainProgramFiles)

MarkovTreeFunctions

-(CODE) new\_exp\_inf2.m- This is the main function used to compute the expected inflation rate.

-(CODE) new\_exp\_infvar2.m- This is the main function used to compute the expected variance for the inflation rate.It initially goes through every terminal node and assigned either the variance of regime 1 or the variance of regime 2 to be its value. It then loops through the time periods(from n to 1) and computes the conditional variance for each node(representing each path). It calls the recursive function new\_infvar\_nodeval to do this and ultimately returns the expected inflation variance for the root node as well as tet conditional variance associated with each path.

-(CODE) new\_infvar\_nodeval.m- This is the recursive function that computes the expected inflation variance associated with each node. The base case is for period 1 and this is important because it differs depending on the starting regime. For every period, it applies the expectation equation for the conditional variance using the transition probabilities.

-(CODE) terminal\_inflation.m- This function takes in the parameters of the RS equation(for each regime) as well as the number of periods and the binary paths associated with each node. It then computes the terminal node values by using a for loop to recurse over every period for each node. It adds the mean for the associated regime at that time period and eventually

adds the initial inflation rate, which is multiplied by a product representing the multiplication of the corresponding autoregressive coefficient for each time period.

-(CODE) terminal\_nodes.m- This function goes through every terminal node(there are 2^n of them) and assigns a value to it that represents the unique path associated with it. For example for a node that represents the path:

Initial Regime \rightarrow Regime 2 \right arrow Regime 2 \rightarrow Regime 1. Would have the path 110 associated with it

-----------------------------------------------------------------------------------------------------

(In MainProgramFiles)

Miscellaneous\_functions

-(CODE) divide\_interval.m- This function is used to divide the desired dates(in months) into equally spaced periods. It is returned as a series of start and end index matrices. The first division always represents the initial in-sample data. The rest of the divisions represent the extra data used for each iteration or rebalancing.

-(CODE) regimecount.m- This function create goes through the smoothing probabilities associated with each time and returns the associated regime with each date(based on which probability is higher, the probability of being in regime 1 or of being in regime 2). This function is particularly important when used in context with the current date because knowledge of the current regime is necessary in order to generate the correct markov trees.

-(CODE) return\_inflation\_file.m- This function returns the appropriate inflation file depending if you want to use monthly, quarterly or annual data to generate the RS model. Monthly is good because it accounts for all fluctuations in regime price, but there are computational limitations on far you can recurse. 2^20( about a million) is as far as one can realistically recurse and this returns the expected inflation rate 1 year and 8 months from now. Quarterly is good because it can account for inflation fluctuations(not as well as monthly), but it can also allow prediction of expected inflation of up to 5 years from now so it has a computational benefit. .Annual does not really account for many fluctuations in the inflation rate so it is advised to not use it.

-----------------------------------------------------------------------------------------------------

(In MainProgramFiles)

MVO\_functions

-(CODE) benchmark\_MVO.m- This computes the standard MVO portfolio(adjusted for transaction costs). It returns the asset allocation and objective function that quadprog returns.

-(CODE) fetch\_stock\_data.m- REMOVE THIS FILE FROM LATEX

-(CODE) main\_MVO.m- This is the main file where our optimization model is applied and the asset weights are computed.

The function takes in historical inflation, asset returns, transaction costs, inflation betas, desired return, the previous portfolio allocation and the expected inflation rate and expected inflation variance generated from the Markov tree. The function uses the inflation betas for each asset and modifies the nominal return of each asset by adding (inflation Beta) \* (the expected inflation rate). This is the expected asset return(mu) used in the final model. It uses the expected inflation variance as the conventional variance of the market term and it used to compute the noise term as well.

The function returns the asset allocation, the objective function, an adj vector which is a combination of many variables that allows for easy comparison, nominal return vector and the covariance vector(temp\_Q). The temp\_Q vector is used to compute the Sharpe ratio associated with our custom model.

-(CODE) MVO\_comparison.m- This function returns the cumulative returns(essentially portfolio value over time) of

five portfolios: standard MVO, custom inflation-hedged portfolio, S/&P500 and two mutual funds. It takes in the asset allocations of the first two portfolios and computes the cumulative returns. It then plots cumulative returns on a graph as well as regular returns. These graphs are used to view a comparison of the portfolio values over time.

-(CODE) MVO\_params.m- This function simply returns the geometric mean returns, the expected return and covariance matrix for each asset.

-(CODE) nominal\_parameters.m- REMOVE THIS FILE FROM LATEX

-(CODE) solve\_beta3.m- This function takes in the asset prices and inflation rate values. It computes the corresponding

asset returns and then uses matlab’s polyfit function to compute the inflation Beta’s associated with each asset over the time period of data inserted. It also returns an Rsquared value which represents how “accurate” the value of the Beta for each asset is. This value is not really considered because although there is a connection between the inflation rate and the stock price of an asset, this link is not very strong so it is not expected to have a good Rsquared value.

-(CODE) solve\_mvo\_params.m- This function is identical to MVO\_params

-----------------------------------------------------------------------------------------------------

(In MainProgramFiles)

SharpeRatioCalculation

-(CODE) calculateSharpeRatio.m – This function takes in the expected return and covariance of the assets in addition to the asset allocation for our model and for standard MVO. it also takes in temp\_Q, the computed objective function of our model and the risk-free rate for the current period. It returns the sharpe ratio and modified sharpe ratio associated for each portfolio.

-(CODE) sharperatio2.m –This function reforms the actual Sharpe ratio equation and is called from calculateSharpeRatio.