Assignment 4

Due : 11/13/2022 11:59pm

Point : 175

1. (100 points) Using the provided file Assignment4_Q1.xlsm do the following:
    a. Write the implementation of the subroutine **CalcPresentValue(.),** whose template may be found in the BondAnalyticsLib module. Write a wrapper function **CalcPV()** in InterfaceModule that take FaceValue, CouponRate, CouponFreq, SettlementDate, ExpiryDate, MktRate as arguments and return Present Value of the bond . This function will create a variable of Bond type, fill it with the bond data from the arguments and call CalcPresentValue() to calculate present value. Test the function CalcPV() on the single bond data on the Bond worksheet using formula. *(30 points)*
    b. Write **FormatData()** and **FormatColumns()** subroutine in the In the InterfaceFunctions module to do the following formatting - *(30 points)*
        i. BondData worksheet, there is bond data. First, given the first row being comprised of column labels, create a range using VBA called ColNames by finding the last element of the row. Then, set the color to light gray, and set the column name labels in italics. Finally, format this row such that there is a rectangular border around each column name. Put this in a subroutine called **FormatColumns(),** in the InterfaceFunctions module.
        ii. By dropping down one row to cell A2 (you may hard code only A1), use VBA to assign the given data to a range called InputData. Set the color to light green, and again format this range to have rectangular borders around each cell. Write this VBA code in a subroutine called **FormatData()**, in the InterfaceFunctions module. For the remaining two columns (to be calculated), again starting from the 2nd row, set the color to light blue, and format this range to have rectangular borders around each cell. Name the first column range (not including the column name) as pv, and the 2nd column range (again, not including the column name) as ytm.
    c. Write the implementation of the **CalcYieldToMaturity(.)** subroutine, located in the InterfaceFunctions module. This subroutine will loop through each row in BondData worksheet, use Goal Seek to calculate yield to maturity (You can create extra column on the sheet if needed). Place a button on the BondData sheet in order to run this subroutine. *(40 points)*

    Note 1: Use semi-annual compounding. To make this question simple, we assumed flat yield curve i.e. same market rate for different maturities when calculate present values.

    Note 2: For this problem, write your code in a robust way such that it could handle an arbitrary number of rows (and column names); i.e., the only hard-coding there should be of a cell address in your code is A1.

2. (75 points) This exercise uses the Eureka North American monthly hedge fund return data (almost same what you used in previous assignment), contained in the Data worksheet in Assignment4_Q2.xlsx. There is a second worksheet in this file called Style. Solver has been configured to estimate a set of weights for the funds NAHFI through NALSH that will track the style of the fund NAMAC. This is done by bringing in 24 months of data, and then minimizing the variance of the tracking errors (last column at right). Note that the quantity being minimized is actually 10000 x Error Variance, so that solver can get more precise results.
   This is a common task that one needs to do in practice. The estimated weights are then used to predict the next month's target return (in this case, NAMAC).
    a. Your first job is to write VBA subroutine called **Tracker()** that will implement Solver as given; however, it should be able to take in an arbitrary number of style variables. Save the file to Surname_GivenName_Assgn4_Q2.xlsm *(30 points)*
    b. Next, insert a new worksheet called Style1, with range names the same as on the Style sheet, but with the range names appended with the number 1; e.g., the range MinWeights should be named MinWeights1, and so on. Then, write an interface subroutine called **Track1()** that will map these range names to the arguments in Tracker, and call the Tracker sub. Check that your results match those that you get in the

Style sheet. Also, check that the result for the predicted target fund for the following month matches. Then, change the MinWeights constraints to 10% and MaxWeights to 90% for each tracking fund, run Solver manually in the xlsx file, and then verify that the results you get using your VBA code in your xlsm file match. *(20 points)*

c. Finally, insert a second worksheet called Style2, and set it up the same as Style1 except that you will eliminate the NAARB fund from your set of tracking (predictor) funds, and you will use 36 months of data beginning with 2019.01.01. Rename the ranges with a suffix of 2 (e.g. MinWeights -> MinWeights2, etc.), and write another interface sub called Track2 that will take the ranges in the sheet, map them to the parameters in Tracker, and execute the Tracker function. Again, call this routine for both sets of min/max constraints as in part b). You can check your results by modifying the Solver setup in the xlsx file that is provided. *(20 points)*

Note for this assignment: Zip up your two files and the .bas files into a file of the form YourSurname_YourGivenName_Assgn4.zip, and submit it via Canvas.