

CASE STUDY

seventeen **Technical Analysis**

case study **OVERVIEW**

CS17.1	Application Overview and Model Development
CS17.2	Worksheets
CS17.3	User Interface
CS17.4	Procedures
CS17.5	Re-solve Options
CS17.6	Summary
CS17.7	Extensions

CS17.1 *Application Overview and Model Development*

We assume that a person buys shares of a stock when the average market value for a past time horizon falls below a certain value X; likewise, we assume that a person sells shares of a stock when the average market value for a past time horizon rises above a certain value Y. The “Technical Analysis of Market Timing” problem seeks to determine values for X and Y and a maximum profit for buying and selling shares for a certain time horizon. This application analyzes “Technical Analysis of Market Timing” for the two following options:

Option A: The time horizon and a risk rate are set, and we then use various values of X and Y (within a given bounds) to calculate the overall profit. A maximum profit is found among these values.

Option B: The time horizon shifts (within a given bound) and the best values of X and Y are determined for the maximum profit possible for scenarios of buy and sell, buy only, and sell only.

CS19.1.1 **Model Definition and Assumptions**

Let’s now discuss how to make the calculations for determining the overall profit for a given X and Y value. We perform these calculations in a calculation sheet with several columns. Note that the formulas for these calculations are only in the rows corresponding to the time horizon. In a later section, we will further describe the calculation sheet as well as the procedure that generates the calculations, *UpdateCalc*.

The first two columns contain historical data from the S&P records, which we use to calculate the average market price depending on X and Y. The next two columns check if the investor can buy or sell shares based on the current trial value for X and Y, respectively. We simply calculate the average market price based on the value of X for buying and X for selling. We then enter the formulas and copy them for the time horizon period.

```
Range("BuyCheck").Offset(StartCount, 0).FormulaR1C1 = "=AVERAGE(R[-" &
BuyX & "]C[-1]: R[-1]C[-1])"
Range("SellCheck").Offset(StartCount, 0).FormulaR1C1 = "=AVERAGE(R[-" &
SellY & "]C[-2]: R[-1]C[-2])"
Range(Range("BuyCheck").Offset(StartCount, 0),
Range("SellCheck").Offset(StartCount, 0)).Copy
Range(Range("BuyCheck").Offset(StartCount + 1, 0),
Range("SellCheck").Offset(EndCount, 0)).PasteSpecial
```

The next column checks whether or not the user currently owns shares. This is a logical check to ensure that the user does not sell shares that he or she does not own. For the first period in the time horizon, we perform this check using the initial number of shares provided by the user in an input form. For the remaining periods, we check the number of shares owned in the previous period.

```
If Range("StartShares").Value <> "" Then
    Range("Own").Offset(StartCount, 0).Value = "Yes"
```

```

Else
    Range("Own").Offset(StartCount, 0).Value = "No"
End If
Range("Own").Offset(StartCount + 1, 0).FormulaR1C1 = "=IF(R[-1]C[2] = ""Yes"",
""Yes"", IF(R[-1]C[3]=""Yes"", ""No"", R[-1]C))"
Range("Own").Offset(StartCount + 1, 0).Copy
Range(Range("Own").Offset(StartCount + 2, 0), Range("Own").Offset(EndCount,
0)).PasteSpecial

```

The next two columns confirm whether or not the user is going to buy or sell shares for the current period. We check whether or not we are in a scenario in which we are buying or selling. We then check whether or not the buy or sell check was met and whether or not we have shares to sell.

```

Range("Buy").Offset(StartCount, 0).FormulaR1C1 =
"=IF(BuyAllowed=""Yes"",IF(AND(RC[-2] = ""No"", RC[-6]>RC[-5]),
""Yes"", ""No""),""No"")"
Range("Sell").Offset(StartCount, 0).FormulaR1C1 =
"=IF(SellAllowed=""Yes"",IF(AND(RC[-3] = ""Yes"", RC[-7]<RC[-5]),
""Yes"", ""No""),""No"")"
Range(Range("Buy").Offset(StartCount, 0), Range("Sell").Offset(StartCount,
0)).Copy
Range(Range("Buy").Offset(StartCount + 1, 0), Range("Sell").Offset(EndCount,
0)).PasteSpecial

```

For either action, we must determine the number of shares by assuming that we will either buy all the shares for which we have funds or sell all the shares that we currently own. When buying shares, we divide our current capital by the sum of the market price and the transaction cost for buying.

```

Range("NumShares").Offset(StartCount, 0).FormulaR1C1 = "=IF(RC[-2]=""Yes"",
StartCap/(RC[-8]+CostBuy), " & "IF(RC[-1] = ""Yes"", 0, StartShares))"
Range("NumShares").Offset(StartCount + 1, 0).FormulaR1C1 = "=IF(RC[-2] =
""Yes"", R[-1]C[2]/(RC[-8]+CostBuy), " & "IF(RC[-1] = ""Yes"", 0, R[-1]C)"
Range("NumShares").Offset(StartCount + 1, 0).Copy
Range(Range("NumShares").Offset(StartCount+2,0),
Range("NumShares").Offset(EndCount, 0)).PasteSpecial

```

The next column calculates the cash flow. We begin with a starting capital specified by the user in an input form. Any time we buy, we spend all of our money. This cost comes from the market price of the stock and the transaction cost for buying shares. Any time we sell, we sell all of our shares and therefore gain their market price in value. However, we also incur a transaction cost for selling.

```

Range("CashFlow").Offset(StartCount, 0).FormulaR1C1 = "=IF(RC[-4] = ""Yes"",
StartCap-(RC[-10]+CostBuy)*RC[-2], " & "IF(RC[-3]=""Yes"",
StartCap+(RC[-10]-CostSell)*StartShares, StartCap+StartShares*Rate))"
Range("CashFlow").Offset(StartCount + 1, 0).FormulaR1C1 = "=IF(RC[-4] =
""Yes"", R[-1]C-(RC[-10]+CostBuy)*RC[-2], " & "IF(RC[-3]=""Yes"", R[-
1]C+(RC[-10]-CostSell)*R[-1]C[-2], R[-1]C+RC[-2]*Rate)"
Range("CashFlow").Offset(StartCount + 1, 0).Copy
Range(Range("CashFlow").Offset(StartCount+2,0),

```

```
Range("CashFlow").Offset(EndCount, 0).PasteSpecial
```

The overall profit is stored at the top of the sheet. It is equal to our cash flow at the end of the time horizon.

```
Range("Profit").Formula = "=" & Range("CashFlow").Offset(EndCount, 0).Address
```

The formulas provided here are in terms of the cells and variables used in the sheet and the procedure descriptions below.

CS19.1.2 **Input**

For both options, we require the following general input:

- Initial capital
- Initial number of shares
- Risk free rate
- Transaction costs for buying and selling shares
- Upper and lower bounds on X and Y

For Option A, we require the following additional input:

- Time horizon

For Option B, we require the following additional input:

- Shift in time horizon:
 - Starting time
 - Ending time
 - Step size for start time
 - Interval size of time horizon trial values

CS19.1.3 **Output**

The output for Option A is:

- Table with various X and Y values and profit values for each combination
- Maximum profit and X and Y for maximum profit
- Number of buys and sells at maximum profit

The output for Option B is:

- X and Y that yield maximum profit for each scenario:
 - Buying and selling
 - Buying only
 - Selling only

Values for each scenario for various time horizons

CS17.2 *Worksheets*

We will use four worksheets in this application: the welcome sheet, a calculation sheet, an output sheet for Option A, and an output sheet for Option B. The welcome sheet contains the title and the description of the application as well as some images (see Figure CS17.1). The "Start" button takes the user to an input form and then an options form. If the

user selects Option A, he or she will see the output sheet for Option A; otherwise, the output sheet for Option B appears.

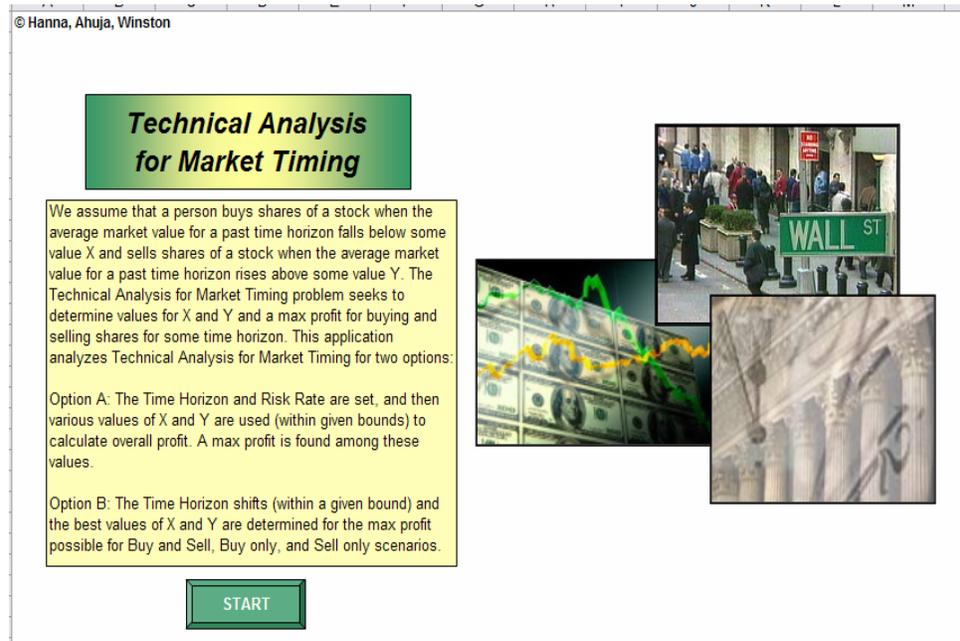


Figure CS17.1 The Welcome Sheet.

We determine the output for Option A and Option B by performing calculations on the “Calculations” sheet (see Figure CS17.2), which the user can view by clicking a “View Calc” button on either of the output sheets. The user can return to the output sheet by clicking the “Back” button. An “End” button allows the user to exit the application.

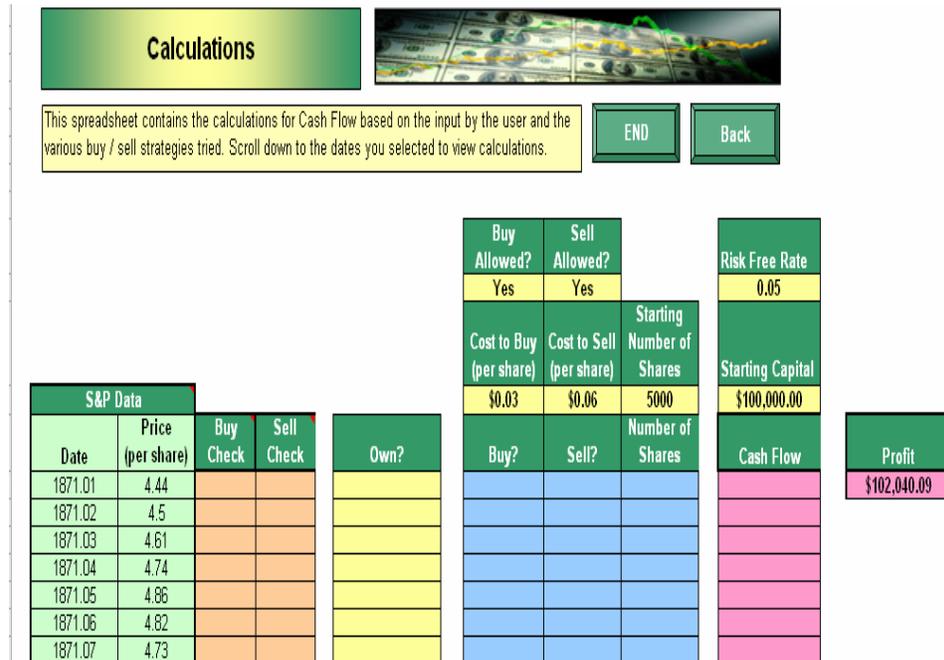


Figure CS17.2(a)

1920.04	8.6	8.863	8.883	No	No	No	0	143850
1920.05	8.06	8.718	8.849	No	No	No	0	143850
1920.06	7.92	8.530	8.730	No	No	No	0	143850
1920.07	7.91	8.363	8.536	No	No	No	0	143850
1920.08	7.6	8.210	8.376	No	No	No	0	143850
1920.09	7.87	8.127	8.211	No	No	No	0	143850
1920.1	7.88	7.993	8.091	No	No	No	0	143850
1920.11	7.48	7.873	8.064	No	No	No	0	143850
1920.12	6.81	7.777	7.915	No	No	No	0	143850
1921.01	7.11	7.592	7.691	No	No	No	0	143850
1921.02	7.06	7.458	7.573	No	No	No	0	143850
1921.03	6.88	7.368	7.465	No	No	No	0	143850
1921.04	6.91	7.203	7.336	No	No	No	0	143850
1921.05	7.12	7.042	7.250	No	Yes	No	20118.88112	0
1921.06	6.55	6.982	7.166	Yes	No	Yes	0	130571.5385
1921.07	6.53	6.938	6.990	No	No	No	0	130571.5385
1921.08	6.45	6.842	6.871	No	No	No	0	130571.5385
1921.09	6.61	6.740	6.826	No	No	No	0	130571.5385
1921.1	6.7	6.695	6.764	No	Yes	No	19401.4173	0
1921.11	7.06	6.660	6.719	Yes	No	No	19401.4173	970.0708652
1921.12	7.31	6.650	6.741	Yes	No	No	19401.4173	1940.14173
1922.01	7.3	6.777	6.791	Yes	No	No	19401.4173	2910.212596
1922.02	7.46	6.905	6.814	Yes	No	No	19401.4173	3880.283461
1922.03	7.74	7.073	6.928	Yes	No	No	19401.4173	4850.354326
1922.04	8.21	7.262	7.079	Yes	No	No	19401.4173	5820.425191

Figure CS17.2(b)

Figure CS17.2 The Calculation sheet.

The first two columns contain historical data from the S&P records. The dates are recorded by year and months, with months represented as decimals. The price per share is included for each date. The next two columns check if the investor can buy or sell shares based on the current trial value for X and Y, respectively. The next column checks whether or not the user currently owns shares. As previously mentioned, this is a logical check to ensure that the user does not sell shares he or she does not own.

Above the next set of columns are several input cells, which the user provided in the initial input form and in the re-solve options on the output sheets. The “Buy Allowed” and “Sell Allowed” cells permit the user to vary the scenarios for Option B. The “Buy” and “Sell” columns confirm whether or not the user wants to buy or sell shares for the current period. For either action, the number of shares is also determined. Next, we calculate the cash flow in the next column and store the overall profit at the top of the sheet. Note that the formulas for these calculations are only in the rows corresponding to the time horizon.

The output sheet for Option A has a worksheet user interface that allows the user to specify the risk rate and the time horizon (see Figure CS17.3). Bounds for the X and Y values have been set in the input form already. Based on these bounds, a table is created to try different combinations of X and Y in the calculations. We record the overall profit for each trial, the maximum profit amongst these is highlighted, and the number of buys and sells at this maximum profit value.

If the user changes the risk rate or time horizon, he or she must then click the “Re-solve” button to re-run the calculations. The “View Calc” button allows the user to view the calculation sheet and the “End” button to exit the application.

The output sheet for Option B displays the X and Y values that yield a maximum profit for a shifting time horizon (see Figure CS17.4). These X, Y, and profit values are calculated for three different scenarios: buying and selling, buying only, and selling only. The time horizon shifts according to the values provided by the user in a user form for Option B. If the user wishes to changes these values, he or she can press the “Re-solve” button. The “View Calc” button allows the user to view the calculation sheet and the “End” button to exit the application.

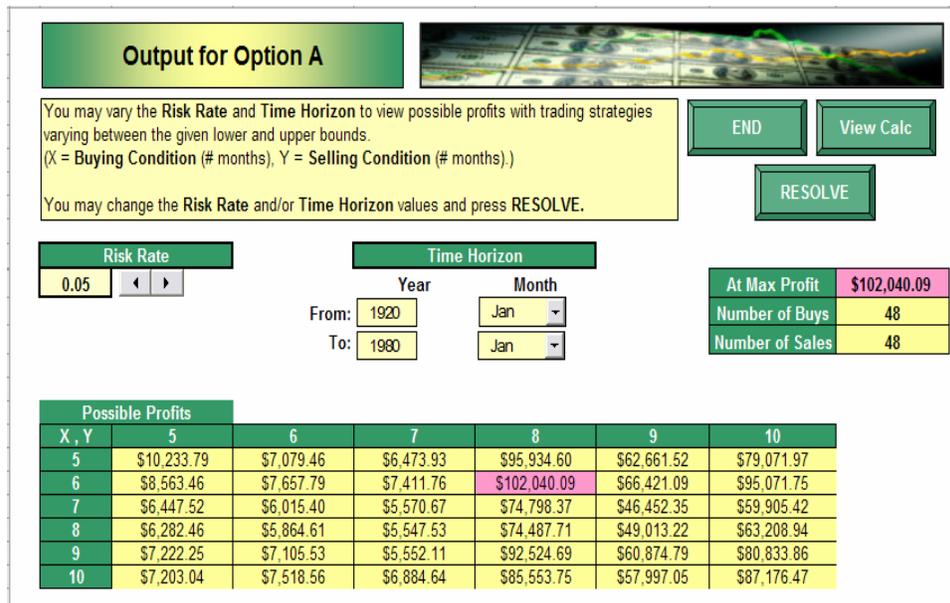


Figure CS17.3 The output sheet for Option A.

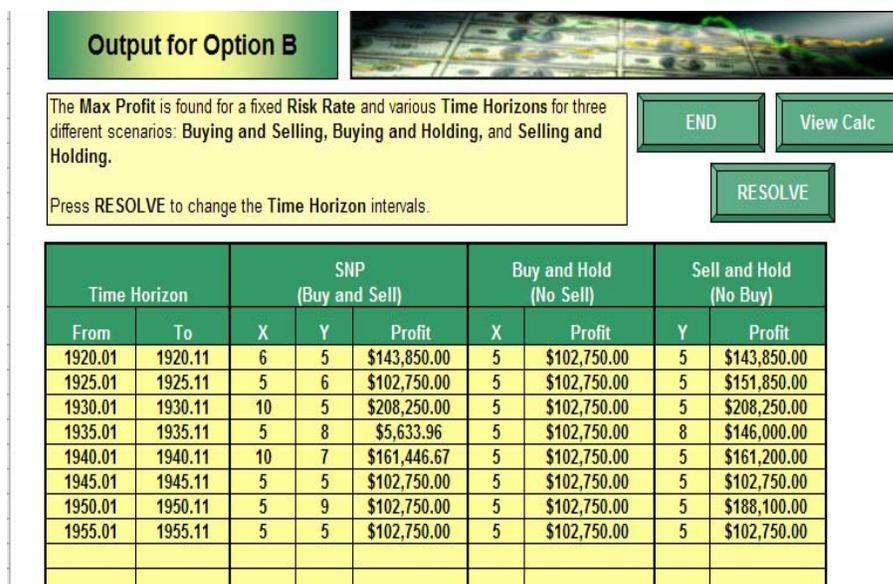


Figure CS17.4 The output sheet for Option B.

<p>Summary</p>	Welcome sheet	Application description and "Start" button.
	Calculation sheet	Input recorded and calculations performed.
	Output sheet for Option A	Output for Option A recorded and re-solve options given.
	Output sheet for Option B	Output for Option B recorded and re-solve options given.

CS17.3 *User Interface*

This application consists of four user forms, some controls on the output for Option A worksheet, and several navigational and functional buttons. The first form that the user sees is the input form (see Figure CS17.5). This form records the initial capital, the initial number of shares, the risk free rate, the transaction costs for buying and selling shares, and the bounds for X and Y. If the user is solving Option A, he or she will have a chance to modify the risk free rate and then re-solve. We use frames and text boxes to receive the input, as well as some extra labels to clarify formatting to the user.

Input Form

Please enter the following input values.

Initial Capital \$ 100000

Starting Number of Shares 5000

Risk Free Rate 5 %

Transaction Costs

Cost of Buying (per share) \$.03

Cost of Selling (per share) \$.06

Buying and Selling Condition Bounds (months)

	Lower Bound	Upper Bound
Buying Condition	5	10
Selling Condition	5	10

Cancel OK

Figure CS17.5 The input form.

The next form that the user sees is the options form (see Figure CS17.6), which allows the user to choose between Option A and Option B. We employ a frame with two mutually exclusive option buttons.

Options

Please select one of the following options.

See possible profits with varying risk rate and fixed time horizon.

Find max profit for a fixed risk rate and varying time horizon.

Cancel OK

Figure CS17.6 The options form.

If the user selects Option A, then the Option A form appears (see Figure CS17.7). This form receives the initial values for the time horizon, which (along with the risk rate provided in the input form) will be used to perform the initial calculations for Option A.

Figure CS17.7 The Option A initial time horizon form.

To re-solve Option A, the user does not return to any of these forms, but rather manipulates the controls on the worksheet to change the risk rate and/or the time horizon (see Figure CS17.8). We employ a spin button linked to a cell to allow the user to change the risk rate. We include two text boxes and two combo boxes to allow the user to select new years and/or months for the time horizon.

Possible Profits		5	6	7	8	9	10
5	X, Y	\$10,233.79	\$7,079.46	\$6,473.93	\$95,934.60	\$62,661.52	\$79,071.97
6		\$8,563.46	\$7,657.79	\$7,411.76	\$102,040.09	\$66,421.09	\$95,071.75
7		\$6,447.52	\$6,015.40	\$5,570.67	\$74,798.37	\$46,452.35	\$59,905.42
8		\$6,282.46	\$5,864.61	\$5,547.53	\$74,487.71	\$49,013.22	\$63,208.94
9		\$7,222.25	\$7,105.53	\$5,552.11	\$92,524.69	\$60,874.79	\$80,833.86
10		\$7,203.04	\$7,518.56	\$6,884.64	\$85,553.75	\$57,997.05	\$87,176.47

Figure CS17.8 The user interface controls on the output sheet for Option A.

If the user selects Option B from the option form, the Option B form appears next (see Figure CS17.9). This form receives input for the shifting time horizon. The length of each

trial horizon is set in the “Period of Time per Interval” text box. The step size for the shift in the starting date is also an input. The starting and ending dates are considered to be bounds for the trial time horizons. We include combo boxes for the months of the time horizon bounds. This form appears to the user again when he or she re-solves Option B.

Figure CS17.9 The Option B time horizon shift form.

 Summary	Input Form	Receives the initial input from the user.
	Options Form	Allows the user to select Option A or Option B.
	Option A Form	Receives the initial time horizon for Option A.
	Controls on output sheet for Option A	Allows the user to change the risk rate and the time horizon for resolving Option A.
	Option B Form	Receives input for the shifting time horizon for Option B. Also used in re-solve options for Option B.

CS17.4 *Procedures*

We will now outline the procedures for this application. We begin with the *Main* sub procedure and variable definitions (see Figure CS17.10). The *Main* sub procedure simply calls a *ClearPrev* procedure and displays the input form and options form. The *ClearPrev* procedure clears the output sheets' values and formatting for Option A and Option B. It also initializes some variables.

The code to receive the input on the input form appears in Figure CS17.11. In this code, the upper and lower bounds for X and Y are assigned to their corresponding variables. The other input values are recorded on the calculations sheet. The application makes note of the initial risk rate on the corresponding user interface of Option A's output sheet.

```

Option Explicit

Public i As Integer, j As Integer, DoA As Boolean, DoB As Boolean, _
BuyLB As Integer, BuyUB As Integer, SellLB As Integer, SellUB As Integer, _
BuyK As Integer, SellY As Integer, SizeK As Integer, SizeY As Integer, _
DateFrom As Double, DateTo As Double, StartTime As Double, EndTime As Double, _
StepSize As Integer, IntervalSize As Integer, MaxProfit As Double, MaxK As Integer, MaxY As Integer, _
StartCount As Integer, EndCount As Integer, iter As Integer, ws As Worksheet

Sub Main()
    Call ClearPrev

    frmInput.Show
    frmOptions.Show
End Sub

Sub ClearPrev()
    'OutputA
    With Range(Range("ProfitA"), Range("ProfitA").End(xlDown).End(xlToRight))
        .Interior.ColorIndex = xlNone
        .Borders(xlInsideHorizontal).LineStyle = xlNone
        .Borders(xlInsideVertical).LineStyle = xlNone
        .ClearContents
    End With
    Range("ProfitA").Interior.ColorIndex = 50
    Range("ProfitA").Value = "X , Y"

    'OutputB
    Range(Range("BFrom").Offset(1, 0), Range("NoBuyProfit").Offset(1, 0).End(xlDown)).ClearContents

    'DoA = False
    DoB = False
End Sub

```

Figure CS17.10 Variable declarations, Main procedure, and ClearPrev procedure.

```

Private Sub cmdOK_Click()
    'record values
    BuyLB = txtBuyLB.Value
    BuyUB = txtBuyUB.Value
    SellLB = txtSellLB.Value
    SellUB = txtSellUB.Value

    With Worksheets("StockData")
        .Range("StartCap").Value = txtCap.Value
        .Range("StartShares").Value = txtShares.Value
        .Range("Rate").Value = txtRate.Value / 100
        .Range("CostBuy").Value = txtCostBuy.Value
        .Range("CostSell").Value = txtCostSell.Value
    End With

    Worksheets("OutputA").spnRate.Value = txtRate.Value

    Unload Me
End Sub

Private Sub cmdCancel_Click()
    Unload Me
End Sub

```

Figure CS17.11 The code for the input form.

The code to receive the user's selection on the options form appears in Figure CS17.12. If the user selects Option A, then the output sheet for Option A is formatted, a corresponding Boolean variable is set, the user is taken to the output sheet for Option A, and the Option

A form comes into view. Once the user has entered the initial time horizon in the Option A form, the application calls the *SolveA* procedure. If the user selects Option B, a corresponding Boolean variable is set, the user is taken to the output sheet for Option B, and the application calls the *SolveB* procedure. In response to the *SolveB* procedure, the Option B form appears to receive the time horizon shift input.

```

Private Sub cmdOK_Click()
    If optA Then
        Unload Me

        'format Buy/Sell trial area
        i = 0
        Do
            With Range("BuyLB").Offset(i, 0)
                .Value = BuyLB + i
                .Interior.ColorIndex = 50
                .BorderAround Weight:=xlThin
            End With
            i = i + 1
        Loop While BuyLB + i <= BuyUB
        SizeX = i - 1

        j = 0
        Do
            With Range("SellLB").Offset(0, j)
                .Value = SellLB + j
                .Interior.ColorIndex = 50
                .BorderAround Weight:=xlThin
            End With
            j = j + 1
        Loop While SellLB + j <= SellUB
        SizeY = j - 1

        With Range(Range("ProfitA").Offset(1, 1), Range("ProfitA").Offset(SizeX + 1, SizeY + 1))
            .Interior.ColorIndex = 36
            .Borders(xlInsideHorizontal).Weight = xlThin
            .Borders(xlInsideVertical).Weight = xlThin
        End With

        DoB = False
        Worksheets("OutputA").Visible = True
        Worksheets("Welcome").Visible = False
        frmSolveA.Show
        Call SolveA

    ElseIf optB Then
        Unload Me

        DoB = True
        Worksheets("OutputB").Visible = True
        Worksheets("Welcome").Visible = False
        Call SolveB

    End If
End Sub

```

Figure CS17.12 The code for the options form.

The code for the Option A form records the initial time horizon on the user interface of Option A's output sheet (see Figure CS17.13). This form is also initialized to reveal the first month to be selected in the "Month" combo boxes.

The *SolveA* procedure performs the main calculations for Option A (see Figure CS17.14). It begins by determining whether or not we are solving Option A. If we are, then we record any changes made to the user interface controls on the output sheet for Option A. We use the *SolveA* procedure to solve Option B and should not record this information in that case.

We then loop through all of the X,Y combinations to calculate their profit values and determine the maximum profit scenario. To perform the calculations, we use an *UpdateCalc* procedure (see Figure CS17.15), which loops through the rows in the calculation sheet that correspond to the time horizon specified by the user. In these rows, we enter all of the necessary formulas for each column in the calculation sheet.

```
Private Sub cmdCancel_Click()  
    Unload Me  
  
    Worksheets("Welcome").Visible = True  
    Worksheets("OutputA").Visible = False  
  
End  
End Sub  
  
Private Sub cmdOK_Click()  
    'record  
    With Worksheets("OutputA")  
        .txtYearFrom.Value = txtYearFrom.Value  
        .txtYearTo.Value = txtYearTo.Value  
        .cmbMonthFrom.Value = cmbMonthFrom.Value  
        .cmbMonthTo.Value = cmbMonthTo.Value  
    End With  
  
    Unload Me  
End Sub  
  
Private Sub UserForm_Initialize()  
    cmbMonthFrom.RowSource = "Months"  
    cmbMonthFrom.ListIndex = 0  
    cmbMonthTo.RowSource = "Months"  
    cmbMonthTo.ListIndex = 0  
End Sub
```

Figure CS17.13 The code for the Option A form.

For Option B, we use the *SolveB* procedure shown in Figure CS17.16. This procedure begins by displaying the Option B form to the user. Since we will display this form every time Option B is re-solved, we call from the *SolveB* procedure instead of the options form.

Next, we loop through the various time horizon values, solving for the best X and Y values for maximum profit each time. For each time horizon value, we examine three scenarios: buying and selling, only buying, and only selling. To enforce these scenarios, we modify the "Buy Allowed" and "Sell Allowed" cells on the calculation sheet. We call the *SolveA* procedure to find the optimal X, Y, and profit values for each scenario. Again, the *SolveA* procedure calls the *UpdateCalc* procedure to modify the calculation sheet formulas as needed for the varying X, Y, and time horizon values.

The Option B form code simply records the time horizon bounds, step size, and interval length (see Figure CS17.17). The form is also initialized to select the first month in the "Months" combo boxes.

```

Sub SolveA()
    Application.ScreenUpdating = False
    If DoB = False Then
        'record and updates in time horizon and rate
        With Worksheets("OutputA")
            DateFrom = Cdbl(.txtYearFrom.Value & "." & .cmbMonthFrom.Value)
            DateTo = Cdbl(.txtYearTo.Value & "." & .cmbMonthTo.Value)
        End With
        Worksheets("StockData").Range("Rate").Value = Worksheets("OutputA").Range("RateSpn").Value
        Range("BuyAllowed").Value = "Yes"
        Range("SellAllowed").Value = "Yes"
    End If

    'find profit values for each BuyX, SellY combination
    MaxProfit = 0
    For i = 0 To SizeX
        For j = 0 To SizeY
            'update average calculations
            BuyX = Range("BuyLB").Offset(i, 0).Value
            SellY = Range("SellLB").Offset(0, j).Value
            Call UpdateCalc

            'record profit
            Range("ProfitA").Offset(i + 1, j + 1).Value = Range("Profit").Value

            'check if MaxProfit
            If Range("ProfitA").Offset(i + 1, j + 1).Value > MaxProfit Then
                MaxProfit = Range("ProfitA").Offset(i + 1, j + 1).Value
                'record MaxX, MaxY
                MaxX = i
                MaxY = j
            End If
        Next j
    Next i

    'highlight Range("ProfitA").Offset(MaxX+1, MaxY+1)
    Range("ProfitA").Offset(MaxX + 1, MaxY + 1).Interior.ColorIndex = 38
    Application.ScreenUpdating = True

    'extension? count # buys/sales at max profit
    Application.ScreenUpdating = False
    BuyX = Range("BuyLB").Offset(MaxX, 0).Value
    SellY = Range("SellLB").Offset(0, MaxY).Value
    Call UpdateCalc
    Range("MaxProfit").Value = Range("Profit").Value
    Range("NumBuys").Value = Application.WorksheetFunction.CountIf(Range(Range("Buy").Offset(StartCount, 0), _
        Range("Buy").Offset(EndCount, 0)), "Yes")
    Range("NumSales").Value = Application.WorksheetFunction.CountIf(Range(Range("Sell").Offset(StartCount, 0), _
        Range("Sell").Offset(EndCount, 0)), "Yes")
    Application.ScreenUpdating = True
End Sub

```

Figure CS17.14 The SolveA procedure.

```

Sub UpdateCalc()
'check that there is enough hist data for first calculations
Range("CalcData").ClearContents
StartCount = Application.WorksheetFunction.Match(DateFrom, Range("Dates"), 0)
EndCount = Application.WorksheetFunction.Match(DateTo, Range("Dates"), 0)
If StartCount < Application.WorksheetFunction.Max(BuyX, SellY) Then
    StartCount = Application.WorksheetFunction.Max(BuyX, SellY)
    Range(Range("BuyCheck").Offset(1, 0), Range("CashFlow").Offset(StartCount - 1, 0)).ClearContents
    MsgBox "Your starting date had to be shifted in order to apply your buying/selling conditions."
End If

'update formulas on Stock Data sheet
With Worksheets("StockData")
    Range("BuyCheck").Offset(StartCount, 0).FormulaR1C1 = "=AVERAGE(R[-" & BuyX & "]C[-1]: R[-1]C[-1])"
    Range("SellCheck").Offset(StartCount, 0).FormulaR1C1 = "=AVERAGE(R[-" & SellY & "]C[-2]: R[-1]C[-2])"
    Range(Range("BuyCheck").Offset(StartCount, 0), Range("SellCheck").Offset(StartCount, 0)).Copy
    Range(Range("BuyCheck").Offset(StartCount + 1, 0), Range("SellCheck").Offset(EndCount, 0)).PasteSpecial

    If Range("StartShares").Value <> "" Then
        Range("Own").Offset(StartCount, 0).Value = "Yes"
    Else
        Range("Own").Offset(StartCount, 0).Value = "No"
    End If
    Range("Own").Offset(StartCount + 1, 0).FormulaR1C1 = "=IF(R[-1]C[2] = ""Yes"", ""Yes"", IF(R[-1]C[3]="Yes"", ""No"", R[-1]C))"
    Range("Own").Offset(StartCount + 1, 0).Copy
    Range(Range("Own").Offset(StartCount + 2, 0), Range("Own").Offset(EndCount, 0)).PasteSpecial

    Range("Buy").Offset(StartCount, 0).FormulaR1C1 = "=IF(BuyAllowed=""Yes"",IF(AND(RC[-2] = ""No"", RC[-6]>RC[-5]), ""Yes"", ""No""),""No"")"
    Range("Sell").Offset(StartCount, 0).FormulaR1C1 = "=IF(SellAllowed=""Yes"",IF(AND(RC[-3] = ""Yes"", RC[-7]<RC[-6]), ""Yes"", ""No""),""No"")"
    Range(Range("Buy").Offset(StartCount, 0), Range("Sell").Offset(StartCount, 0)).Copy
    Range(Range("Buy").Offset(StartCount + 1, 0), Range("Sell").Offset(EndCount, 0)).PasteSpecial

    Range("NumShares").Offset(StartCount, 0).FormulaR1C1 = "=IF(RC[-2]="Yes"", StartCap/(RC[-8]+CostBuy), " & _
        "IF(RC[-1] = ""Yes"", 0, StartShares))"
    Range("NumShares").Offset(StartCount + 1, 0).FormulaR1C1 = "=IF(RC[-2] = ""Yes"", R[-1]C[2]/(RC[-8]+CostBuy), " & _
        "IF(RC[-1] = ""Yes"", 0, R[-1]C))"
    Range("NumShares").Offset(StartCount + 1, 0).Copy
    Range(Range("NumShares").Offset(StartCount + 2, 0), Range("NumShares").Offset(EndCount, 0)).PasteSpecial

    Range("CashFlow").Offset(StartCount, 0).FormulaR1C1 = "=IF(RC[-4] = ""Yes"", StartCap-(RC[-10]+CostBuy)*RC[-2], " & _
        "IF(RC[-3]="Yes"", StartCap+(RC[-10]-CostSell)*StartShares, StartCap+StartShares*Rate))"
    Range("CashFlow").Offset(StartCount + 1, 0).FormulaR1C1 = "=IF(RC[-4] = ""Yes"", R[-1]C-(RC[-10]+CostBuy)*RC[-2], " & _
        "IF(RC[-3]="Yes"", R[-1]C+(RC[-10]-CostSell)*R[-1]C[-2], R[-1]C+RC[-2]*Rate))"
    Range("CashFlow").Offset(StartCount + 1, 0).Copy
    Range(Range("CashFlow").Offset(StartCount + 2, 0), Range("CashFlow").Offset(EndCount, 0)).PasteSpecial

    Range("Profit").Formula = "=" & Range("CashFlow").Offset(EndCount, 0).Address
End With
End Sub

```

Figure CS17.15 The UpdateCalc procedure.

```

Sub SolveB()
    frmSolveB.Show
    Range(Range("BFrom").Offset(1, 0), Range("NoBuyProfit").Offset(1, 0).End(xlDown)).ClearContents

    Application.ScreenUpdating = False
    iter = 1
    DateFrom = StartTime
    DateTo = DateFrom + (Int(IntervalSize / 12) + (IntervalSize Mod 12) / 100)

    Do While DateTo <= EndTime
        Application.ScreenUpdating = False
        Range("BFrom").Offset(iter, 0).Value = DateFrom
        Range("BTo").Offset(iter, 0).Value = DateTo

        'buy/sell
        Range("BuyAllowed").Value = "Yes"
        Range("SellAllowed").Value = "Yes"
        Call SolveA
        Application.ScreenUpdating = False
        Range("SNPX").Offset(iter, 0).Value = Range("BuyLB").Offset(MaxX, 0).Value
        Range("SNPY").Offset(iter, 0).Value = Range("SellLB").Offset(0, MaxY).Value
        Range("SNPProfit").Offset(iter, 0).Value = Range("MaxProfit").Value

        'buy / no sell
        Range("BuyAllowed").Value = "Yes"
        Range("SellAllowed").Value = "No"
        Call SolveA
        Application.ScreenUpdating = False
        Range("NoSellX").Offset(iter, 0).Value = Range("BuyLB").Offset(MaxX, 0).Value
        Range("NoSellProfit").Offset(iter, 0).Value = Range("MaxProfit").Value

        'sell / no buy
        Range("BuyAllowed").Value = "No"
        Range("SellAllowed").Value = "Yes"
        Call SolveA
        Application.ScreenUpdating = False
        Range("NoBuyY").Offset(iter, 0).Value = Range("SellLB").Offset(0, MaxY).Value
        Range("NoBuyProfit").Offset(iter, 0).Value = Range("MaxProfit").Value

        DateFrom = DateFrom + (Int(StepSize / 12) + (StepSize Mod 12) / 100)
        DateTo = DateFrom + (Int(IntervalSize / 12) + (IntervalSize Mod 12) / 100)
        iter = iter + 1
    Loop

    Application.ScreenUpdating = True
End Sub

```

Figure CS17.16 The *SolveB* procedure.

```

Private Sub cmdCancel_Click()
    Unload Me

    Worksheets("Welcome").Visible = True
    Worksheets("OutputB").Visible = False
End Sub

Private Sub cmdOK_Click()
    'record
    StartTime = CDb1(txtStartYear.Value & "." & cmbStartMonth.Value)
    EndTime = CDb1(txtEndYear.Value & "." & cmbEndMonth.Value)
    StepSize = txtStepStart.Value
    IntervalSize = txtIntervalSize.Value

    Unload Me
End Sub

Private Sub UserForm_Initialize()
    cmbStartMonth.RowSource = "Months"
    cmbStartMonth.ListIndex = 0
    cmbEndMonth.RowSource = "Months"
    cmbEndMonth.ListIndex = 0
End Sub

```

Figure CS17.17 The code for the Option B form.

We use the navigational procedures for the “End,” “View Calc,” and “Back” buttons (see Figure CS17.18).

```

Sub EndProg()
    Worksheets("Welcome").Visible = True
    ActiveSheet.Visible = False
End Sub

Sub ViewCalc()
    Worksheets("StockData").Visible = True
    Set ws = ActiveSheet
    ws.Visible = False
End Sub

Sub GoBack()
    ws.Visible = True
    Worksheets("StockData").Visible = False
End Sub

```

Figure CS17.18 Navigational procedures.

 <p>Summary</p>	Main	Initializes the application and retrieves the initial input from user.
	ClearPrev	Initializes the variables, clears previous values, and clears animation layout formatting.
	Input Form code	Records the initial input.
	Options Form code	Determines whether to solve Option A or Option B.
	Option A Form code	Records initial time horizon for Option A.
	SolveA	Loops through combinations of X and Y to determine maximum profit.
	UpdateCalc	Updates formulas in calculation sheet depending on X, Y, and time horizon.
	SolveB	Loops through time horizon values for three scenarios.
	Option B Form code	Records time horizon shift values.
	Navigational	Used for navigational buttons.

CS17.5 *Re-solve Options*

There are two re-solve options for this application: Option A and Option B. For Option A, the user can change the risk rate and/or time horizon and click the “Re-solve” button to recall the *SolveA* procedure. For Option B, the user can click the “Re-solve” button to recall the *SolveB* procedure. This procedure again displays the Option B form, which allows the user to change the shift in time horizon values.

 <p>Summary</p>	Option A Re-solve Option	The user can change the risk rate and/or time horizon.
	Option B Re-solve Option	Allows the user to change the shift in time horizon values.

CS17.6 *Summary*

- The “Technical Analysis of Market Timing” problem seeks to determine values for X and Y and a max profit for buying and selling shares for a certain time horizon.
- There are four worksheets in this application: the welcome sheet, a calculation sheet, an output sheet for Option A, and an output sheet for Option B.
- There are four user forms, some controls on the output for the Option A worksheet, and a few navigational and functional buttons.
- There are several procedures for this application that allow the user to enter input, perform calculations, view results, and re-solve.
- There is a re-solve option for Option A in which the and Option B.

CS17.7 *Extensions*

- What other analysis could you add to the output sheets? (Hint: What about charts and/or graphs?)
- What other scenarios could be considered for Option B?
- Add error checking procedures to all of the form codes. For example, ensure that only integer numbers are entered for the step size in the Option B form.
- Expand this application to consider more than one stock at a time.