```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Drawing;
using RDotNet;
using System.IO;
using System.Runtime.InteropServices;

namespace CropShearSetPoint
{
    class Program : System.Windows.Forms.Form
    {
//------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------
        // This code is meant for hiding the console window.
        [DllImport("kernel32.dll")]
        static extern IntPtr GetConsoleWindow();

        [DllImport("user32.dll")]
        static extern bool ShowWindow(IntPtr hWnd, int nCmdShow);

        const int SW_HIDE = 0;
        const int SW_SHOW = 5;

//------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------
        static int Main(string[] args)
        {

            var handle = GetConsoleWindow();

            //// Hide the Console
            //ShowWindow(handle, SW_HIDE);

            // Show the Console
            ShowWindow(handle, SW_SHOW);

            Console.WriteLine("Input the grade of the coil you want to use");
```

```
            // Coiling mode must be HIGH for this program to run.
            // The console application is called from the windows form application.
            string grade = args[0];

            // The console application working as a stand alone.
            // string grade = "0";
            // string grade = Console.ReadLine();

            // Read the input value of new L2 set points from text file corresponding to the input grade.
            var inputFilePath = "C:/Users/Dewal Agarwal/Desktop/Intern
Project/CropShearSetPoint/Resources/Grade_ip.txt";
            // Write the output of the final setpoints to the text file for the crop shear.
            var outputFilePath = "C:/Users/Dewal Agarwal/Desktop/Intern
Project/CropShearSetPoint/Resources/Grade_op.txt";

            // This is the value of the latest set points of the Head and Tail.
            string[] L2New = System.IO.File.ReadAllLines(@inputFilePath);
            double L2H = Convert.ToDouble(L2New[0]);
            double L2T = Convert.ToDouble(L2New[1]);

            // While naming the token/identifier for the grade input prefix the index with the Grade Class.

//------------------------------------------------------------------------------------------------------------
-----------------------------------------------------
            // SERIES 200
//------------------------------------------------------------------------------------------------------------
-----------------------------------------------------
            // FOR GRADE JSLUSD
            if (grade == "21")
            {
                // These are the errors based upon the speed analysis and physical measurements of the crops.
                // This error needs to be altered in future if other discrepency are detected.
                const int FixedErrorH = 150;
                const int FixedErrorT = 100;

                var scriptFilePath = "C:/Users/Dewal Agarwal/Desktop/Intern
Project/CropShearSetPoint/R_Script/R_Script.R";      // Path of the R script
                var headFilePath = "C:/Users/Dewal Agarwal/Desktop/Intern
Project/CropShearSetPoint/Resources/200_Grade_JSLUSD/Grade_JSLUSDH.txt";    // Path of the H data
```

```
                var tailFilePath = "C:/Users/Dewal Agarwal/Desktop/Intern
Project/CropShearSetPoint/Resources/200_Grade_JSLUSD/Grade_JSLUSDT.txt";    // Path of the T data
                var foreFilePath = "C:/Users/Dewal Agarwal/Desktop/Intern
Project/CropShearSetPoint/Resources/200_Grade_JSLUSD/Grade_JSLUSDfop.txt";  // Path of the forecast

                ExecuteScriptFile(scriptFilePath, headFilePath, tailFilePath, foreFilePath);

                // Read each line of the file into a string array. Each element of the array is one line of the
file.
                string[] lines = System.IO.File.ReadAllLines(@foreFilePath);

                string HeadForecast = lines[0];
                HeadForecast = HeadForecast.TrimStart('[');
                HeadForecast = HeadForecast.TrimStart('1');
                HeadForecast = HeadForecast.TrimStart(']');
                HeadForecast = HeadForecast.Trim();
                double Head = Convert.ToDouble(HeadForecast);

                string TailForecast = lines[1];
                TailForecast = TailForecast.TrimStart('[');
                TailForecast = TailForecast.TrimStart('1');
                TailForecast = TailForecast.TrimStart(']');
                TailForecast = TailForecast.Trim();
                double Tail = Convert.ToDouble(TailForecast);

                // This is the final setpoint that needs to be sent to the HMI
                double L2Hop = (L2H + Head) / 2 + FixedErrorH;
                double L2Top = (L2T + Tail) / 2 + FixedErrorT;

                // This is for the precaution
                if (L2Hop >= 600 || L2Top >= 350)
                {
                    L2Hop = 600;
                    L2Top = 350;
                }

                // Write the output values as calculated above into the text file.
                string[] output = { Convert.ToString(L2Hop), Convert.ToString(L2Top) };
                System.IO.File.WriteAllLines(@outputFilePath, output);

                // Display the file contents by using a foreach loop.
```

```
                System.Console.WriteLine("The forecasted values of Head and Tail setpoints are-");
                foreach (string line in lines)
                {
                    Console.WriteLine("\t" + line); // Use a tab to indent each line of the file.
                }

                // Display the final setpoints for the Head and Tail Cut.
                System.Console.WriteLine("The final setpoints of Head and Tail are-");
                foreach (string line in output)
                {
                    Console.WriteLine("\t" + line); // Use a tab to indent each line of the file.
                }

                // Writing the new value to the file and removing the old one from it.
                var file = new List<string>(System.IO.File.ReadAllLines(@headFilePath));
                file.RemoveAt(1);
                File.WriteAllLines(@headFilePath, file.ToArray());

                string newContent = L2New[0];
                File.AppendAllText(@headFilePath, newContent);

                // Removing the oldest observation of setpoint and adding the latest one (FIFO).
                file = new List<string>(System.IO.File.ReadAllLines(@tailFilePath));
                file.RemoveAt(1);
                File.WriteAllLines(@tailFilePath, file.ToArray());

                newContent = L2New[1];
                File.AppendAllText(@tailFilePath, newContent);

                //System.Threading.Thread.Sleep(3000);
            }

//-------------------------------------------------------------------------------------------------------
-------------------------------------------------------
            // FOR GRADE JT
            else if (grade == "22")
            {
                // These are the errors based upon the speed analysis and physical measurements of the crops.
                // This error needs to be altered in future if other discrepency are detected.
                const int FixedErrorH = 150;
                const int FixedErrorT = 100;
```

```
                var scriptFilePath = "C:/Users/Dewal Agarwal/Desktop/Intern
Project/CropShearSetPoint/R_Script/R_Script.R";      // Path of the R script
                var headFilePath = "C:/Users/Dewal Agarwal/Desktop/Intern
Project/CropShearSetPoint/Resources/200_Grade_JT/Grade_JTH.txt";    // Path of the H data
                var tailFilePath = "C:/Users/Dewal Agarwal/Desktop/Intern
Project/CropShearSetPoint/Resources/200_Grade_JT/Grade_JTT.txt";    // Path of the T data
                var foreFilePath = "C:/Users/Dewal Agarwal/Desktop/Intern
Project/CropShearSetPoint/Resources/200_Grade_JT/Grade_JTfop.txt";  // Path of the forecast

                ExecuteScriptFile(scriptFilePath, headFilePath, tailFilePath, foreFilePath);

                // Read each line of the file into a string array. Each element of the array is one line of the
file.
                string[] lines = System.IO.File.ReadAllLines(@foreFilePath);

                string HeadForecast = lines[0];
                HeadForecast = HeadForecast.TrimStart('[');
                HeadForecast = HeadForecast.TrimStart('1');
                HeadForecast = HeadForecast.TrimStart(']');
                HeadForecast = HeadForecast.Trim();
                double Head = Convert.ToDouble(HeadForecast);

                string TailForecast = lines[1];
                TailForecast = TailForecast.TrimStart('[');
                TailForecast = TailForecast.TrimStart('1');
                TailForecast = TailForecast.TrimStart(']');
                TailForecast = TailForecast.Trim();
                double Tail = Convert.ToDouble(TailForecast);

                // This is the final setpoint that needs to be sent to the HMI
                double L2Hop = (L2H + Head) / 2 + FixedErrorH;
                double L2Top = (L2T + Tail) / 2 + FixedErrorT;

                // This is for the precaution
                if (L2Hop >= 600 || L2Top >= 350)
                {
                    L2Hop = 600;
                    L2Top = 350;
                }
```

```
                // Write the output values as calculated above into the text file.
                string[] output = { Convert.ToString(L2Hop), Convert.ToString(L2Top) };
                System.IO.File.WriteAllLines(@outputFilePath, output);

                // Display the file contents by using a foreach loop.
                System.Console.WriteLine("The forecasted values of Head and Tail setpoints are-");
                foreach (string line in lines)
                {
                    Console.WriteLine("\t" + line); // Use a tab to indent each line of the file.
                }

                // Display the final setpoints for the Head and Tail Cut.
                System.Console.WriteLine("The final setpoints of Head and Tail are-");
                foreach (string line in output)
                {
                    Console.WriteLine("\t" + line); // Use a tab to indent each line of the file.
                }

                // Writing the new value to the file and removing the old one from it.
                var file = new List<string>(System.IO.File.ReadAllLines(@headFilePath));
                file.RemoveAt(1);
                File.WriteAllLines(@headFilePath, file.ToArray());

                string newContent = L2New[0];
                File.AppendAllText(@headFilePath, newContent);

                // Removing the oldest observation of setpoint and adding the latest one (FIFO).
                file = new List<string>(System.IO.File.ReadAllLines(@tailFilePath));
                file.RemoveAt(1);
                File.WriteAllLines(@tailFilePath, file.ToArray());

                newContent = L2New[1];
                File.AppendAllText(@tailFilePath, newContent);

                //System.Threading.Thread.Sleep(3000);
            }

//-------------------------------------------------------------------------------------------------------------
------------------------------------------------------
            // SERIES 300
//-------------------------------------------------------------------------------------------------------------
```

```
------------------------------------------------------------
            // FOR GRADE 304
            else if (grade == "31")
            {
                // These are the errors based upon the speed analysis and physical measurements of the crops.
                // This error needs to be altered in future if other discrepency are detected.
                const int FixedErrorH = 150;
                const int FixedErrorT = 100;

                var scriptFilePath = "C:/Users/Dewal Agarwal/Desktop/Intern
Project/CropShearSetPoint/R_Script/R_Script.R";   // Path of the R script
                var headFilePath = "C:/Users/Dewal Agarwal/Desktop/Intern
Project/CropShearSetPoint/Resources/300_Grade_304/Grade_304H.txt";    // Path of the H data
                var tailFilePath = "C:/Users/Dewal Agarwal/Desktop/Intern
Project/CropShearSetPoint/Resources/300_Grade_304/Grade_304T.txt";    // Path of the T data
                var foreFilePath = "C:/Users/Dewal Agarwal/Desktop/Intern
Project/CropShearSetPoint/Resources/300_Grade_304/Grade_304fop.txt";  // Path of the forecast

                ExecuteScriptFile(scriptFilePath, headFilePath, tailFilePath, foreFilePath);
                // Read each line of the file into a string array. Each element of the array is one line of the
file.
                string[] lines = System.IO.File.ReadAllLines(@foreFilePath);

                string HeadForecast = lines[0];
                HeadForecast = HeadForecast.TrimStart('[');
                HeadForecast = HeadForecast.TrimStart('1');
                HeadForecast = HeadForecast.TrimStart(']');
                HeadForecast = HeadForecast.Trim();
                double Head = Convert.ToDouble(HeadForecast);

                string TailForecast = lines[1];
                TailForecast = TailForecast.TrimStart('[');
                TailForecast = TailForecast.TrimStart('1');
                TailForecast = TailForecast.TrimStart(']');
                TailForecast = TailForecast.Trim();
                double Tail = Convert.ToDouble(TailForecast);

                // This is the final setpoint that needs to be sent to the HMI
                double L2Hop = (L2H + Head) / 2 + FixedErrorH;
                double L2Top = (L2T + Tail) / 2 + FixedErrorT;
```

```
// This is for the precaution
if (L2Hop >= 480 || L2Top >= 400)
{
    L2Hop = 480;
    L2Top = 400;
}

// Write the output values as calculated above into the text file.
string[] output = { Convert.ToString(L2Hop), Convert.ToString(L2Top) };
System.IO.File.WriteAllLines(@outputFilePath, output);

// Display the file contents by using a foreach loop.
System.Console.WriteLine("The forecasted values of Head and Tail setpoints are-");
foreach (string line in lines)
{
    Console.WriteLine("\t" + line); // Use a tab to indent each line of the file.
}

// Display the final setpoints for the Head and Tail Cut.
System.Console.WriteLine("The final setpoints of Head and Tail are-");
foreach (string line in output)
{
    Console.WriteLine("\t" + line); // Use a tab to indent each line of the file.
}

// Removing the oldest observation of setpoint and adding the latest one (FIFO).
var file = new List<string>(System.IO.File.ReadAllLines(@headFilePath));
file.RemoveAt(1);
File.WriteAllLines(@headFilePath, file.ToArray());

string newContent = L2New[0];
File.AppendAllText(@headFilePath, newContent);

// Writing the new value to the file and removing the old one from it.
file = new List<string>(System.IO.File.ReadAllLines(@tailFilePath));
file.RemoveAt(1);
File.WriteAllLines(@tailFilePath, file.ToArray());

newContent = L2New[1];
File.AppendAllText(@tailFilePath, newContent);
```

```
                //System.Threading.Thread.Sleep(3000);
            }

//-------------------------------------------------------------------------------------------------------------
------------------------------------------------------
            // FOR GRADE 301
            else if (grade == "32")
            {
                // These are the errors based upon the speed analysis and physical measurements of the crops.
                // This error needs to be altered in future if other discrepency are detected.
                const int FixedErrorH = 150;
                const int FixedErrorT = 100;

                var scriptFilePath = "C:/Users/Dewal Agarwal/Desktop/Intern
Project/CropShearSetPoint/R_Script/R_Script.R";   // Path of the R script
                var headFilePath = "C:/Users/Dewal Agarwal/Desktop/Intern
Project/CropShearSetPoint/Resources/300_Grade_301/Grade_301H.txt";    // Path of the H data
                var tailFilePath = "C:/Users/Dewal Agarwal/Desktop/Intern
Project/CropShearSetPoint/Resources/300_Grade_301/Grade_301T.txt";    // Path of the T data
                var foreFilePath = "C:/Users/Dewal Agarwal/Desktop/Intern
Project/CropShearSetPoint/Resources/300_Grade_301/Grade_301fop.txt";  // Path of the forecast

                ExecuteScriptFile(scriptFilePath, headFilePath, tailFilePath, foreFilePath);
                // Read each line of the file into a string array. Each element of the array is one line of the
file.
                string[] lines = System.IO.File.ReadAllLines(@foreFilePath);

                string HeadForecast = lines[0];
                HeadForecast = HeadForecast.TrimStart('[');
                HeadForecast = HeadForecast.TrimStart('1');
                HeadForecast = HeadForecast.TrimStart(']');
                HeadForecast = HeadForecast.Trim();
                double Head = Convert.ToDouble(HeadForecast);

                string TailForecast = lines[1];
                TailForecast = TailForecast.TrimStart('[');
                TailForecast = TailForecast.TrimStart('1');
                TailForecast = TailForecast.TrimStart(']');
                TailForecast = TailForecast.Trim();
                double Tail = Convert.ToDouble(TailForecast);
```

```
// This is the final setpoint that needs to be sent to the HMI
double L2Hop = (L2H + Head) / 2 + FixedErrorH;
double L2Top = (L2T + Tail) / 2 + FixedErrorT;

// This is for the precaution
if (L2Hop >= 480 || L2Top >= 400)
{
    L2Hop = 480;
    L2Top = 400;
}

// Write the output values as calculated above into the text file.
string[] output = { Convert.ToString(L2Hop), Convert.ToString(L2Top) };
System.IO.File.WriteAllLines(@outputFilePath, output);

// Display the file contents by using a foreach loop.
System.Console.WriteLine("The forecasted values of Head and Tail setpoints are-");
foreach (string line in lines)
{
    Console.WriteLine("\t" + line); // Use a tab to indent each line of the file.
}

// Display the final setpoints for the Head and Tail Cut.
System.Console.WriteLine("The final setpoints of Head and Tail are-");
foreach (string line in output)
{
    Console.WriteLine("\t" + line); // Use a tab to indent each line of the file.
}

// Removing the oldest observation of setpoint and adding the latest one (FIFO).
var file = new List<string>(System.IO.File.ReadAllLines(@headFilePath));
file.RemoveAt(1);
File.WriteAllLines(@headFilePath, file.ToArray());

string newContent = L2New[0];
File.AppendAllText(@headFilePath, newContent);

// Writing the new value to the file and removing the old one from it.
file = new List<string>(System.IO.File.ReadAllLines(@tailFilePath));
file.RemoveAt(1);
File.WriteAllLines(@tailFilePath, file.ToArray());
```

```
                newContent = L2New[1];
                File.AppendAllText(@tailFilePath, newContent);

                //System.Threading.Thread.Sleep(3000);
            }

//-------------------------------------------------------------------------------------------------------------
-----------------------------------------------------
            // SERIES 400
//-------------------------------------------------------------------------------------------------------------
-----------------------------------------------------
            // FOR GRADE 409L
            else if (grade == "41")
            {
                // These are the errors based upon the speed analysis and physical measurements of the crops.
                // This error needs to be altered in future if other discrepency are detected.
                const int FixedErrorH = 180;
                const int FixedErrorT = 120;

                var scriptFilePath = "C:/Users/Dewal Agarwal/Desktop/Intern
Project/CropShearSetPoint/R_Script/R_Script.R";   // Path of the R script
                var headFilePath = "C:/Users/Dewal Agarwal/Desktop/Intern
Project/CropShearSetPoint/Resources/400_Grade_409L/Grade_409LH.txt";    // Path of the H data
                var tailFilePath = "C:/Users/Dewal Agarwal/Desktop/Intern
Project/CropShearSetPoint/Resources/400_Grade_409L/Grade_409LT.txt";    // Path of the T data
                var foreFilePath = "C:/Users/Dewal Agarwal/Desktop/Intern
Project/CropShearSetPoint/Resources/400_Grade_409L/Grade_409fop.txt";   // Path of the Forecast

                // Calls the function for executing the R Script.
                ExecuteScriptFile(scriptFilePath, headFilePath, tailFilePath, foreFilePath);

                // Read each line of the file into a string array. Each element of the array is one line of the
file.
                string[] lines = System.IO.File.ReadAllLines(@foreFilePath);

                string HeadForecast = lines[0];
                HeadForecast = HeadForecast.TrimStart('[');
                HeadForecast = HeadForecast.TrimStart('1');
                HeadForecast = HeadForecast.TrimStart(']');
                HeadForecast = HeadForecast.Trim();
```

```
double Head = Convert.ToDouble(HeadForecast);

string TailForecast = lines[1];
TailForecast = TailForecast.TrimStart('[');
TailForecast = TailForecast.TrimStart('1');
TailForecast = TailForecast.TrimStart(']');
TailForecast = TailForecast.Trim();
double Tail = Convert.ToDouble(TailForecast);

// This is the final setpoint that needs to be sent to the HMI
double L2Hop = (L2H + Head) / 2 + FixedErrorH;
double L2Top = (L2T + Tail) / 2 + FixedErrorT;

// This is for the precaution, based on the manual setpoints used.
if (L2Hop >= 550 || L2Top >= 450)
{
    L2Hop = 550;
    L2Top = 450;
}

// Write the output values as calculated above into the text file.
string[] output = { Convert.ToString(L2Hop), Convert.ToString(L2Top) };
System.IO.File.WriteAllLines(@outputFilePath, output);

// Display the file contents by using a foreach loop.
System.Console.WriteLine("The forecasted values of Head and Tail setpoints are-");
foreach (string line in lines)
{
    Console.WriteLine("\t" + line); // Use a tab to indent each line of the file.
}

// Display the final setpoints for the Head and Tail Cut.
System.Console.WriteLine("The final setpoints of Head and Tail are-");
foreach (string line in output)
{
    Console.WriteLine("\t" + line); // Use a tab to indent each line of the file.
}

// Writing the new value to the file and removing the old one from it.
var file = new List<string>(System.IO.File.ReadAllLines(@headFilePath));
file.RemoveAt(1);
```

```
                File.WriteAllLines(@headFilePath, file.ToArray());

                string newContent = L2New[0];
                File.AppendAllText(@headFilePath, newContent);

                // Removing the oldest observation of setpoint and adding the latest one (FIFO).
                file = new List<string>(System.IO.File.ReadAllLines(@tailFilePath));
                file.RemoveAt(1);
                File.WriteAllLines(@tailFilePath, file.ToArray());

                newContent = L2New[1];
                File.AppendAllText(@tailFilePath, newContent);

                // Delay of 3 sec before console window closes.
                //System.Threading.Thread.Sleep(3000);
            }

//-------------------------------------------------------------------------------------------------------------
------------------------------------------------------
            // FOR GRADE 430
            else if (grade == "42")
            {
                // These are the errors based upon the speed analysis and physical measurements of the crops.
                // This error needs to be altered in future if other discrepency are detected.
                const int FixedErrorH = 180;
                const int FixedErrorT = 120;

                var scriptFilePath = "C:/Users/Dewal Agarwal/Desktop/Intern
Project/CropShearSetPoint/R_Script/R_Script.R";   // Path of the R script
                var headFilePath = "C:/Users/Dewal Agarwal/Desktop/Intern
Project/CropShearSetPoint/Resources/400_Grade_430/Grade_430H.txt";    // Path of the H data
                var tailFilePath = "C:/Users/Dewal Agarwal/Desktop/Intern
Project/CropShearSetPoint/Resources/400_Grade_430/Grade_430T.txt";    // Path of the T data
                var foreFilePath = "C:/Users/Dewal Agarwal/Desktop/Intern
Project/CropShearSetPoint/Resources/400_Grade_430/Grade_430fop.txt";  // Path of the forecast

                ExecuteScriptFile(scriptFilePath, headFilePath, tailFilePath, foreFilePath);

                // Read each line of the file into a string array. Each element of the array is one line of the
file.
                string[] lines = System.IO.File.ReadAllLines(@foreFilePath);
```

```
string HeadForecast = lines[0];
HeadForecast = HeadForecast.TrimStart('[');
HeadForecast = HeadForecast.TrimStart('1');
HeadForecast = HeadForecast.TrimStart(']');
HeadForecast = HeadForecast.Trim();
double Head = Convert.ToDouble(HeadForecast);

string TailForecast = lines[1];
TailForecast = TailForecast.TrimStart('[');
TailForecast = TailForecast.TrimStart('1');
TailForecast = TailForecast.TrimStart(']');
TailForecast = TailForecast.Trim();
double Tail = Convert.ToDouble(TailForecast);

// This is the final setpoint that needs to be sent to the HMI
double L2Hop = (L2H + Head) / 2 + FixedErrorH;
double L2Top = (L2T + Tail) / 2 + FixedErrorT;

// This is for the precaution
if (L2Hop >= 550 || L2Top >= 400)
{
    L2Hop = 550;
    L2Top = 400;
}

// Write the output values as calculated above into the text file.
string[] output = { Convert.ToString(L2Hop), Convert.ToString(L2Top) };
System.IO.File.WriteAllLines(@outputFilePath, output);

// Display the file contents by using a foreach loop.
System.Console.WriteLine("The forecasted values of Head and Tail setpoints are-");
foreach (string line in lines)
{
    Console.WriteLine("\t" + line); // Use a tab to indent each line of the file.
}

// Display the final setpoints for the Head and Tail Cut.
System.Console.WriteLine("The final setpoints of Head and Tail are-");
foreach (string line in output)
{
```

```
                    Console.WriteLine("\t" + line); // Use a tab to indent each line of the file.
                }

                // Writing the new value to the file and removing the old one from it.
                var file = new List<string>(System.IO.File.ReadAllLines(@headFilePath));
                file.RemoveAt(1);
                File.WriteAllLines(@headFilePath, file.ToArray());

                string newContent = L2New[0];
                File.AppendAllText(@headFilePath, newContent);

                // Removing the oldest observation of setpoint and adding the latest one (FIFO).
                file = new List<string>(System.IO.File.ReadAllLines(@tailFilePath));
                file.RemoveAt(1);
                File.WriteAllLines(@tailFilePath, file.ToArray());

                newContent = L2New[1];
                File.AppendAllText(@tailFilePath, newContent);

                //System.Threading.Thread.Sleep(3000);
            }
//-------------------------------------------------------------------------------------------------------------
------------------------------------------------------
            // Smilarly we can declare more grade and there corresponding data base file paths.
            // For any other input, that isnt declared initially.
            else
            {
                Console.WriteLine("Unspecified grade.");
            }

//-------------------------------------------------------------------------------------------------------------
------------------------------------------------------

            // Function for starting the new form that displays output.
            // If not required just comment the code written in the next two lines.
            Application.EnableVisualStyles();
            Application.Run(new Form1());

            return 0;

        }
```

```
//-------------------------------------------------------------------------------------------------------------
-------------------------------------------------------
        // For initialising the R Engine in, to to process the setpoints and produce the forecasted value.
        // R Version 3.4.0 needs to be pre-installed before using this part of the code, it is used to run the
R Script.
        public static void ExecuteScriptFile(string scriptFilePath, string paramForScript1, string
paramForScript2, string paramForScript3)
        {
            using (var en = REngine.GetInstance())
            {
                var args_r = new string[3] { paramForScript1, paramForScript2, paramForScript3 };
                var execution = "source('" + scriptFilePath + "')";
                en.SetCommandLineArguments(args_r);
                en.Evaluate(execution);
            }
        }

    }
}
```