Info. Security and Privacy in Dist. System

Spring 2019

Project Report

On

Profiling Internet Users

By
Shivani Singh
U 61828927

Objective:

The objective of this project was to demonstrate if the internet usage of each subject is statistically indistinguishable when compared to the internet usage of the same subject over time, while simultaneously being distinguishable when compared to Internet usage of other subjects. Also, we need to study how the time window that is chosen for profiling affects the answer to the problem described above.

In this project we used octects/duration as a criterion to implement a profile for each user. Also, we had to use three time intervals 10sec, 227 sec and 300 sec to figure out which time window contains the minimum number of users which are statistically indistinguishable when compared to the internet usage of the same subject over time, while simultaneously being distinguishable when compared to Internet usage of other subjects.

Background Information:

➢ *Tools:*

This project is done using C# language in Visual Studio 2019 Integrated Development Environment (IDE). The reason for choosing C# is because it is pure object-oriented language. Also, working on C# also provide access to all the .NET Framework class libraries which makes writing a code a lot easier.

Visual Studio is a IDE use to build, edit and debug code. It contains powerful debugging tools which was very helpful while working on this project. This project was build using Console Application (.Net Framework).

➢ *Overview:*

In this project we have to analyse the user data. There are 54 files where each file corresponds to one user. “Real first packet” is the date and time which is in epoch format. We have to pick for first two weeks of Feb 2013, only those values where duration is not zero and time comes between 8 am to 5 pm.

For this project we have to spilt the time into time-window of 10 sec, 227 sec and 300 secs and calculate the P value. From the value of P we can observe that:

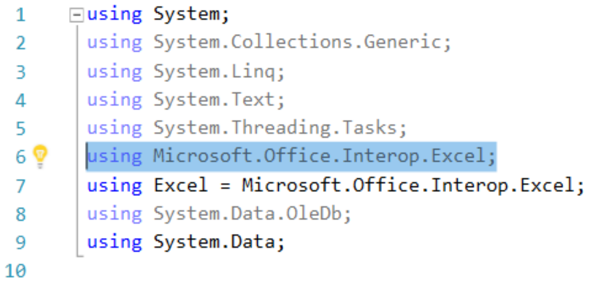
1. If $P \leq 0.5$:
   The correlation coefficient calculated for internet usage pattern for a User 2 is notably smaller than User 1. If so User 2 will be recognised as distinguishable from User 1.

2. If $P \geq 0.5$
   The correlation coefficient calculated for internet usage pattern for a User 2 is not notably smaller than User 1. If so User 2 will be recognised as indistinguishable from User 1.

Implementation:

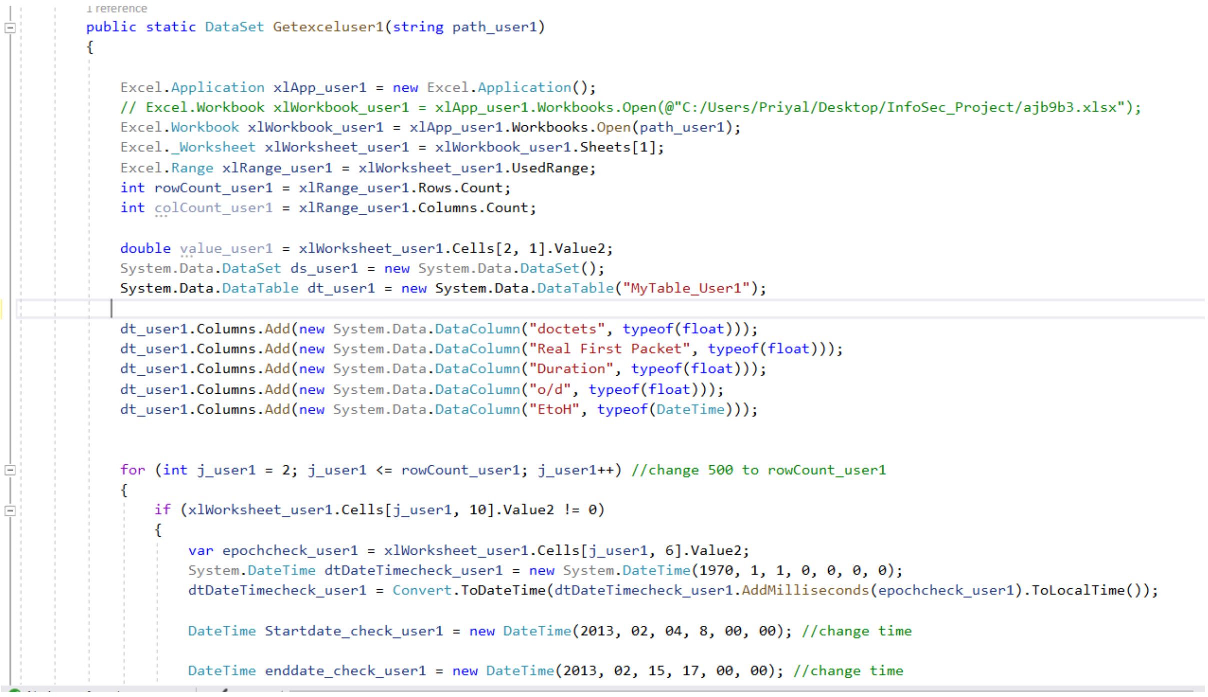
To develop this project first we need to download all the 54 files that was provided to us in a folder. Each file acts as a single user in this project. We have to open each file and compare it with itself and all the other file. The data in these files should be divided into time window of 10 sec, 227 sec and 300 sec. We have to calculate P value in all three-time frame.

- ➢ Start with by adding Microsoft Excel 16.0 object library in your Solution through Reference manager. This library helps in reading all the excel files that was provided.



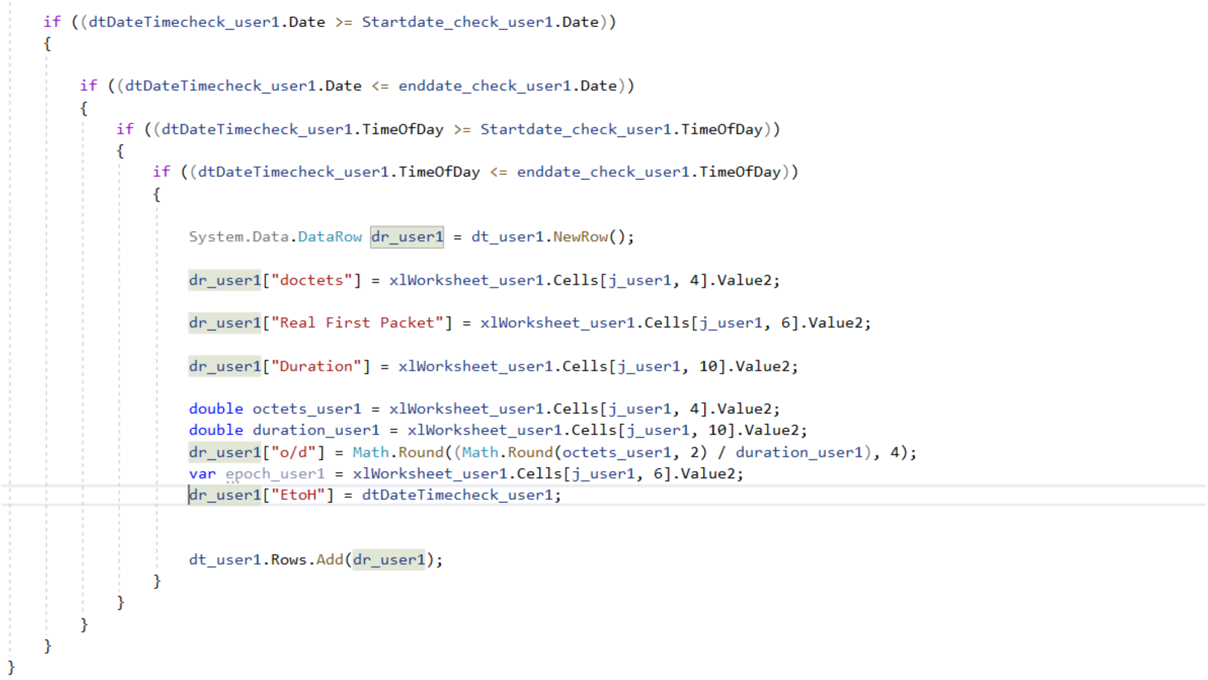
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using Microsoft.Office.Interop.Excel;
using Excel = Microsoft.Office.Interop.Excel;
using System.Data.OleDb;
using System.Data;
```

- ➢ The two excel files are opened and stored in Dataset. Before storing this data in Dataset it is filtered on the basis of duration, Date and time. The rows whose duration is 0 are removed. Also, as it is month long data, so we have to pick only first two week data between 8 am to 5 pm.



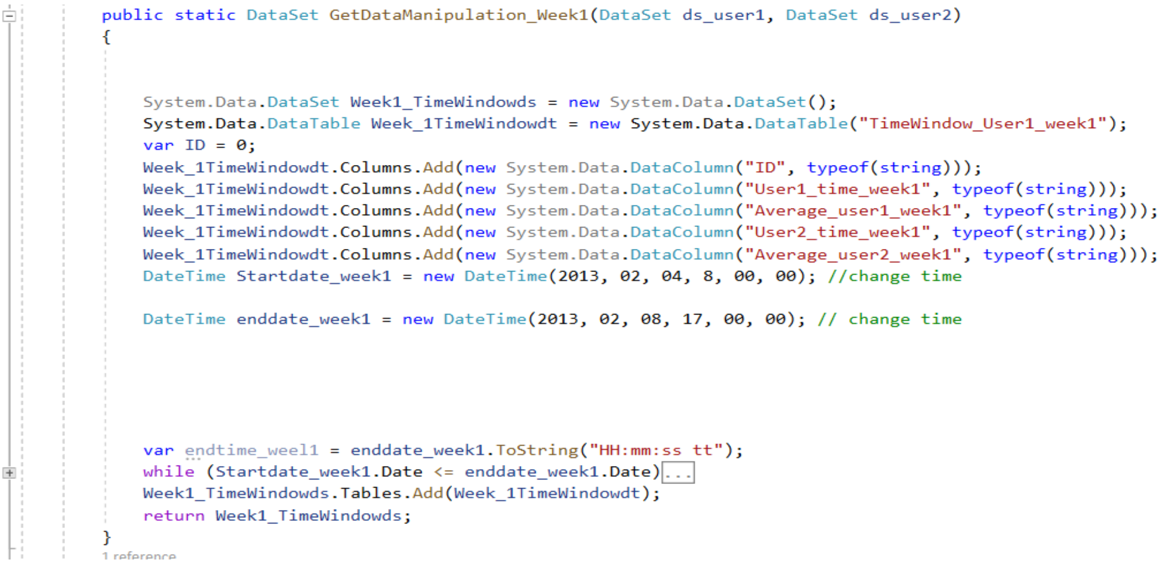
```
1 reference
public static DataSet Getexceluser1(string path_user1)
{

    Excel.Application xlApp_user1 = new Excel.Application();
    // Excel.Workbook xlWorkbook_user1 = xlApp_user1.Workbooks.Open(@"C:/Users/Priyal/Desktop/InfoSec_Project/ajb9b3.xlsx");
    Excel.Workbook xlWorkbook_user1 = xlApp_user1.Workbooks.Open(path_user1);
    Excel._Worksheet xlWorksheet_user1 = xlWorkbook_user1.Sheets[1];
    Excel.Range xlRange_user1 = xlWorksheet_user1.UsedRange;
    int rowCount_user1 = xlRange_user1.Rows.Count;
    int colCount_user1 = xlRange_user1.Columns.Count;

    double value_user1 = xlWorksheet_user1.Cells[2, 1].Value2;
    System.Data.DataSet ds_user1 = new System.Data.DataSet();
    System.Data.DataTable dt_user1 = new System.Data.DataTable("MyTable_User1");

    dt_user1.Columns.Add(new System.Data.DataColumn("doctets", typeof(float)));
    dt_user1.Columns.Add(new System.Data.DataColumn("Real First Packet", typeof(float)));
    dt_user1.Columns.Add(new System.Data.DataColumn("Duration", typeof(float)));
    dt_user1.Columns.Add(new System.Data.DataColumn("o/d", typeof(float)));
    dt_user1.Columns.Add(new System.Data.DataColumn("EtoH", typeof(DateTime)));


    for (int j_user1 = 2; j_user1 <= rowCount_user1; j_user1++) //change 500 to rowCount_user1
    {
        if (xlWorksheet_user1.Cells[j_user1, 10].Value2 != 0)
        {
            var epochcheck_user1 = xlWorksheet_user1.Cells[j_user1, 6].Value2;
            System.DateTime dtDateTimecheck_user1 = new System.DateTime(1970, 1, 1, 0, 0, 0, 0);
            dtDateTimecheck_user1 = Convert.ToDateTime(dtDateTimecheck_user1.AddMilliseconds(epochcheck_user1).ToLocalTime());

            DateTime Startdate_check_user1 = new DateTime(2013, 02, 04, 8, 00, 00); //change time

            DateTime enddate_check_user1 = new DateTime(2013, 02, 15, 17, 00, 00); //change time
```

Calculate octets/duration and convert epoch into human readable time format and store it along with filtered data from the excel file to make the calculation easier.



```
if ((dtDateTimecheck_user1.Date >= Startdate_check_user1.Date))
{

    if ((dtDateTimecheck_user1.Date <= enddate_check_user1.Date))
    {
        if ((dtDateTimecheck_user1.TimeOfDay >= Startdate_check_user1.TimeOfDay))
        {
            if ((dtDateTimecheck_user1.TimeOfDay <= enddate_check_user1.TimeOfDay))
            {

                System.Data.DataRow dr_user1 = dt_user1.NewRow();

                dr_user1["doctets"] = xlWorksheet_user1.Cells[j_user1, 4].Value2;

                dr_user1["Real First Packet"] = xlWorksheet_user1.Cells[j_user1, 6].Value2;

                dr_user1["Duration"] = xlWorksheet_user1.Cells[j_user1, 10].Value2;

                double octets_user1 = xlWorksheet_user1.Cells[j_user1, 4].Value2;
                double duration_user1 = xlWorksheet_user1.Cells[j_user1, 10].Value2;
                dr_user1["o/d"] = Math.Round((Math.Round(octets_user1, 2) / duration_user1), 4);
                var epoch_user1 = xlWorksheet_user1.Cells[j_user1, 6].Value2;
                dr_user1["EtoH"] = dtDateTimecheck_user1;


                dt_user1.Rows.Add(dr_user1);
            }
        }
    }
}
}
```

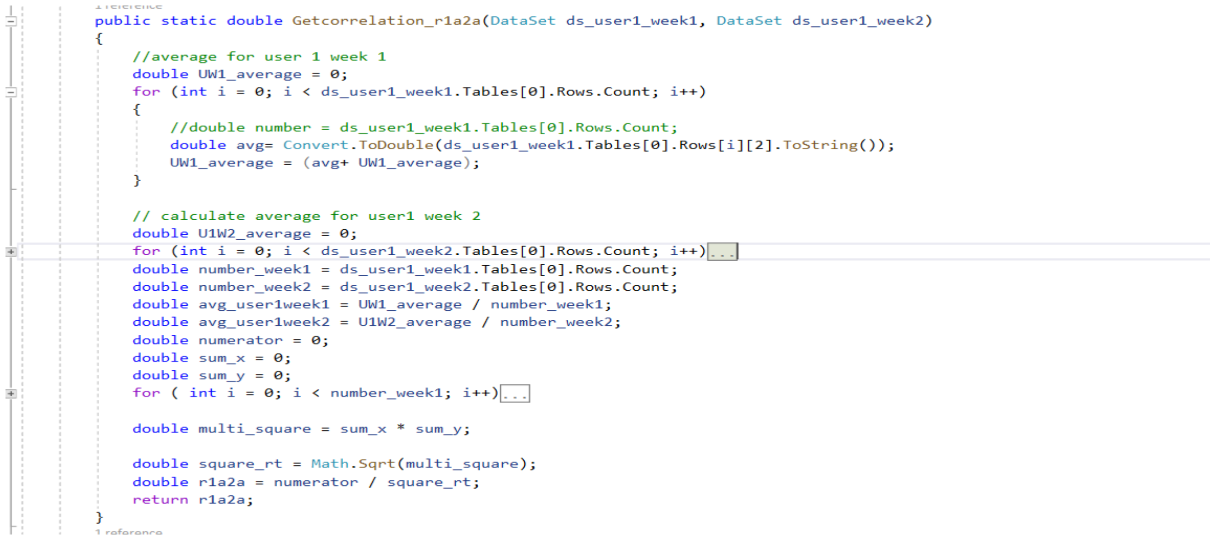
- ➢ Next 10 secs, 227 secs and 300secs time windows are created for the first 2 weeks of Feb 2013 between 8am to 5 pm. Calculate the Octets/duration using dockets and duration from the excel file also, convert the epoch format into human readable time format and if this time format comes under any of the time-window that was created then add it against that time-window. In a case where multiple octets/duration comes under same time-window calculate the average and then store it in the Dataset. This data manipulation has to be done separately for week 1 and week 2 for both User 1 and User 2.



```
public static DataSet GetDataManipulation_Week1(DataSet ds_user1, DataSet ds_user2)
{


    System.Data.DataSet Week1_TimeWindowds = new System.Data.DataSet();
    System.Data.DataTable Week_1TimeWindowdt = new System.Data.DataTable("TimeWindow_User1_week1");
    var ID = 0;
    Week_1TimeWindowdt.Columns.Add(new System.Data.DataColumn("ID", typeof(string)));
    Week_1TimeWindowdt.Columns.Add(new System.Data.DataColumn("User1_time_week1", typeof(string)));
    Week_1TimeWindowdt.Columns.Add(new System.Data.DataColumn("Average_user1_week1", typeof(string)));
    Week_1TimeWindowdt.Columns.Add(new System.Data.DataColumn("User2_time_week1", typeof(string)));
    Week_1TimeWindowdt.Columns.Add(new System.Data.DataColumn("Average_user2_week1", typeof(string)));
    DateTime Startdate_week1 = new DateTime(2013, 02, 04, 8, 00, 00); //change time

    DateTime enddate_week1 = new DateTime(2013, 02, 08, 17, 00, 00); // change time





    var endtime_weel1 = enddate_week1.ToString("HH:mm:ss tt");
    while (Startdate_week1.Date <= enddate_week1.Date)...
    Week1_TimeWindowds.Tables.Add(Week_1TimeWindowdt);
    return Week1_TimeWindowds;
}
1 reference
```

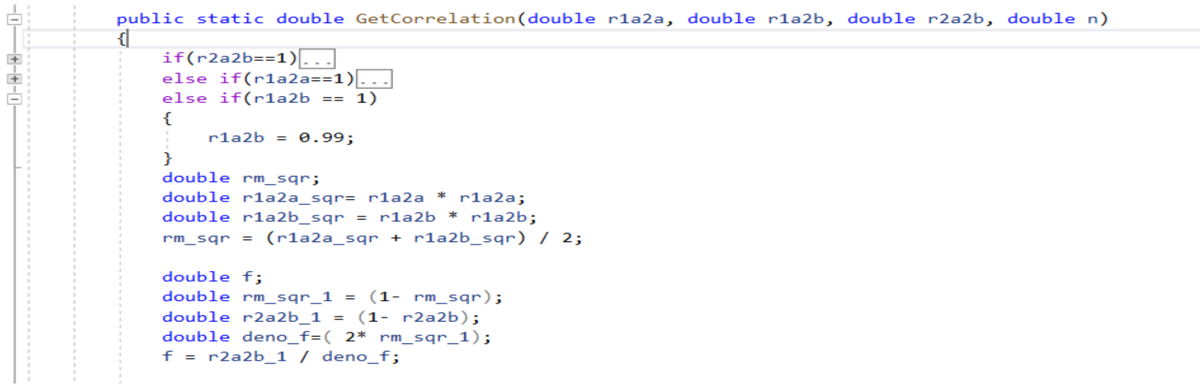
- ➢ After this data is generated, calculate r1a2a, r1a2b, r2a2b by using the formula provided to us.

$$\rho = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2}}$$

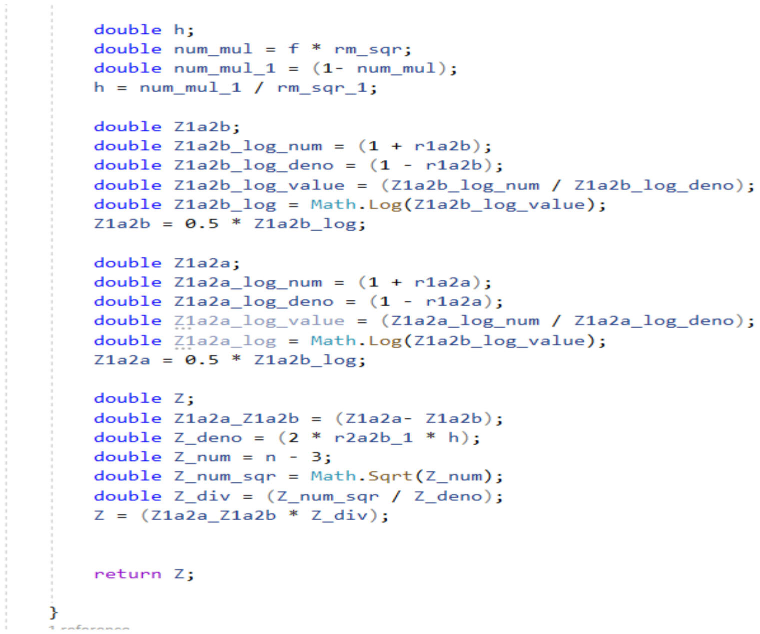


```
public static double Getcorrelation_r1a2a(DataSet ds_user1_week1, DataSet ds_user1_week2)
{
    //average for user 1 week 1
    double UW1_average = 0;
    for (int i = 0; i < ds_user1_week1.Tables[0].Rows.Count; i++)
    {
        //double number = ds_user1_week1.Tables[0].Rows.Count;
        double avg= Convert.ToDouble(ds_user1_week1.Tables[0].Rows[i][2].ToString());
        UW1_average = (avg+ UW1_average);
    }

    // calculate average for user1 week 2
    double U1W2_average = 0;
    for (int i = 0; i < ds_user1_week2.Tables[0].Rows.Count; i++)...
    double number_week1 = ds_user1_week1.Tables[0].Rows.Count;
    double number_week2 = ds_user1_week2.Tables[0].Rows.Count;
    double avg_user1week1 = UW1_average / number_week1;
    double avg_user1week2 = U1W2_average / number_week2;
    double numerator = 0;
    double sum_x = 0;
    double sum_y = 0;
    for ( int i = 0; i < number_week1; i++)...

    double multi_square = sum_x * sum_y;

    double square_rt = Math.Sqrt(multi_square);
    double r1a2a = numerator / square_rt;
    return r1a2a;
}
1 reference
```

In the same way r1a2b and r2a2b is calculated. After getting all the three values we can calculate Z and P using the information provided to us in the file.

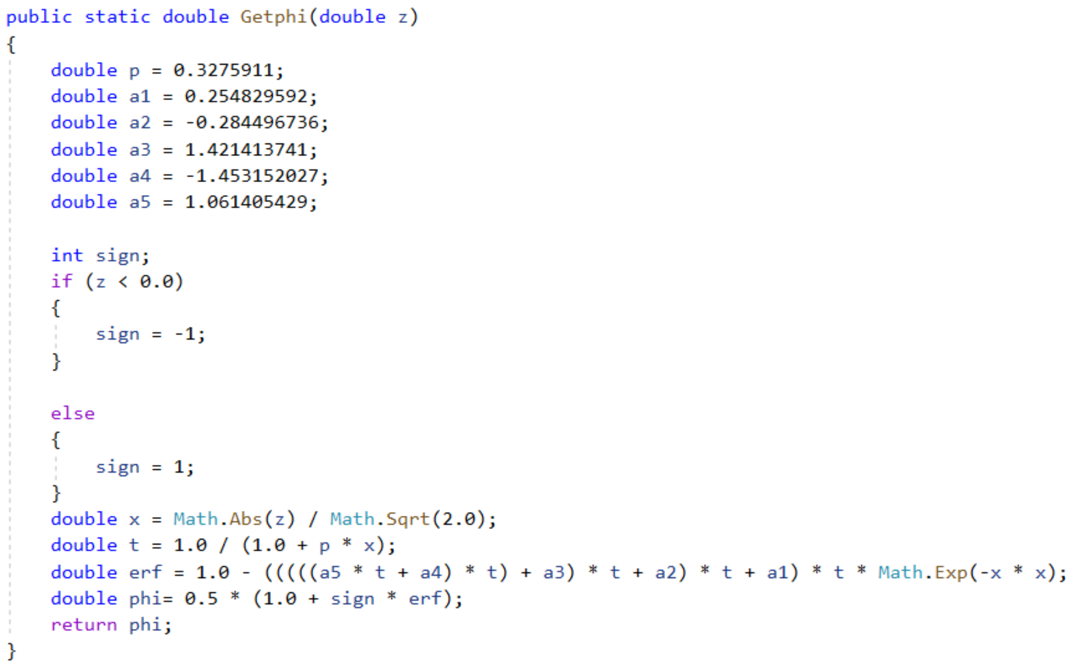


```
public static double GetCorrelation(double r1a2a, double r1a2b, double r2a2b, double n)
{
    if(r2a2b==1)...
    else if(r1a2a==1)...
    else if(r1a2b == 1)
    {
        r1a2b = 0.99;
    }
    double rm_sqr;
    double r1a2a_sqr= r1a2a * r1a2a;
    double r1a2b_sqr = r1a2b * r1a2b;
    rm_sqr = (r1a2a_sqr + r1a2b_sqr) / 2;

    double f;
    double rm_sqr_1 = (1- rm_sqr);
    double r2a2b_1 = (1- r2a2b);
    double deno_f=( 2* rm_sqr_1);
    f = r2a2b_1 / deno_f;
```



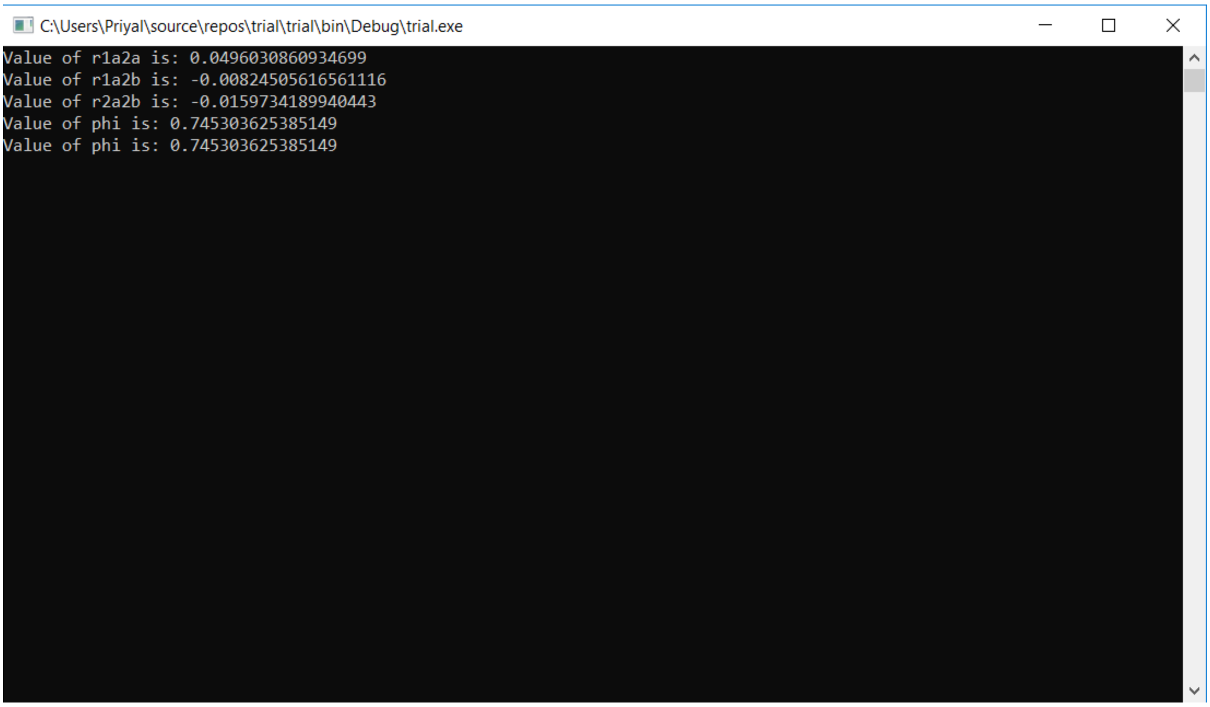
```
    double h;
    double num_mul = f * rm_sqr;
    double num_mul_1 = (1- num_mul);
    h = num_mul_1 / rm_sqr_1;

    double Z1a2b;
    double Z1a2b_log_num = (1 + r1a2b);
    double Z1a2b_log_deno = (1 - r1a2b);
    double Z1a2b_log_value = (Z1a2b_log_num / Z1a2b_log_deno);
    double Z1a2b_log = Math.Log(Z1a2b_log_value);
    Z1a2b = 0.5 * Z1a2b_log;

    double Z1a2a;
    double Z1a2a_log_num = (1 + r1a2a);
    double Z1a2a_log_deno = (1 - r1a2a);
    double Z1a2a_log_value = (Z1a2a_log_num / Z1a2a_log_deno);
    double Z1a2a_log = Math.Log(Z1a2b_log_value);
    Z1a2a = 0.5 * Z1a2b_log;

    double Z;
    double Z1a2a_Z1a2b = (Z1a2a- Z1a2b);
    double Z_deno = (2 * r2a2b_1 * h);
    double Z_num = n - 3;
    double Z_num_sqr = Math.Sqrt(Z_num);
    double Z_div = (Z_num_sqr / Z_deno);
    Z = (Z1a2a_Z1a2b * Z_div);


    return Z;

}
```

Function for calculating P was already provided to us.



```
public static double Getphi(double z)
{
    double p = 0.3275911;
    double a1 = 0.254829592;
    double a2 = -0.284496736;
    double a3 = 1.421413741;
    double a4 = -1.453152027;
    double a5 = 1.061405429;

    int sign;
    if (z < 0.0)
    {
        sign = -1;
    }

    else
    {
        sign = 1;
    }
    double x = Math.Abs(z) / Math.Sqrt(2.0);
    double t = 1.0 / (1.0 + p * x);
    double erf = 1.0 - (((((a5 * t + a4) * t) + a3) * t + a2) * t + a1) * t * Math.Exp(-x * x);
    double phi= 0.5 * (1.0 + sign * erf);
    return phi;
}
```

Result:


C:\Users\Priyal\source\repos\trial\trial\bin\Debug\trial.exe

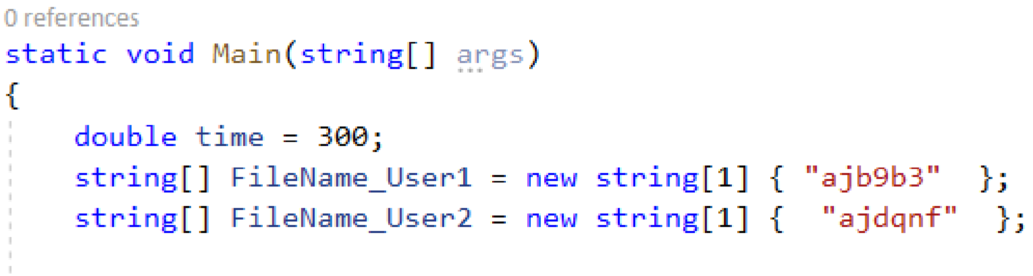
```
Value of r1a2a is: 0.0496030860934699
Value of r1a2b is: -0.00824505616561116
Value of r2a2b is: -0.0159734189940443
Value of phi is: 0.745303625385149
Value of phi is: 0.745303625385149
```

Conclusion:

By looking at the value of P we can conclude that for all the users which are paired with themselves are getting P value as 0.5 which makes them indistinguishable and makes good correlation. Whereas when the users are paired with another user then P value is between 0 to 1 from which we can come to the conclusion that the these two users are indistinguishable.

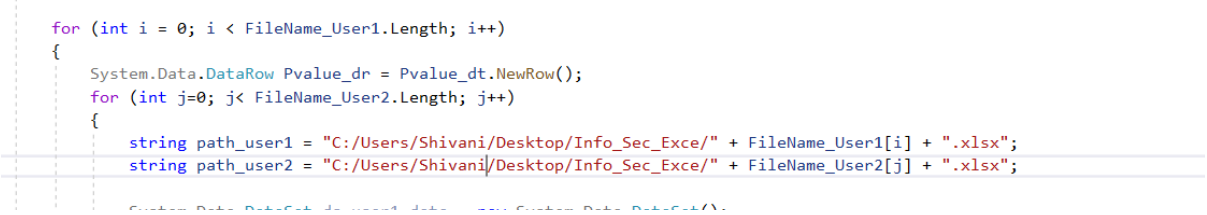
Steps to run this Project:

- Open the Solution in Visual studio and in Main method change time. Here, instead of 300 you can put any value from 10secs, 227 secs and 300 secs.



```
0 references
static void Main(string[] args)
{
    double time = 300;
    string[] FileName_User1 = new string[1] { "ajb9b3"  };
    string[] FileName_User2 = new string[1] {  "ajdqnf"  };
```

- Also, In *line 21* and *line 22* add the file name of whose P value you want to calculate.



```
for (int i = 0; i < FileName_User1.Length; i++)
{
    System.Data.DataRow Pvalue_dr = Pvalue_dt.NewRow();
    for (int j=0; j< FileName_User2.Length; j++)
    {
        string path_user1 = "C:/Users/Shivani/Desktop/Info_Sec_Exce/" + FileName_User1[i] + ".xlsx";
        string path_user2 = "C:/Users/Shivani/Desktop/Info_Sec_Exce/" + FileName_User2[j] + ".xlsx";

        System.Data.DataSet ds_user1_data = new System.Data.DataSet();
```

- In *line 32 and line 33* add the path of your file.
- Click on start.