Information Security Privacy Distributed Systems

Project Report

Profiling Internet Users

By

Vineeth Reddy Cherukupally

U24689350

Objective:

The aim of the Project is 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 statistically distinguishable when compared to Internet usage of other subjects. We also show that how the time window frames (10secs, 227secs, 5 mins) chosen for profiling affects the results of the correlation.

Technology Used:

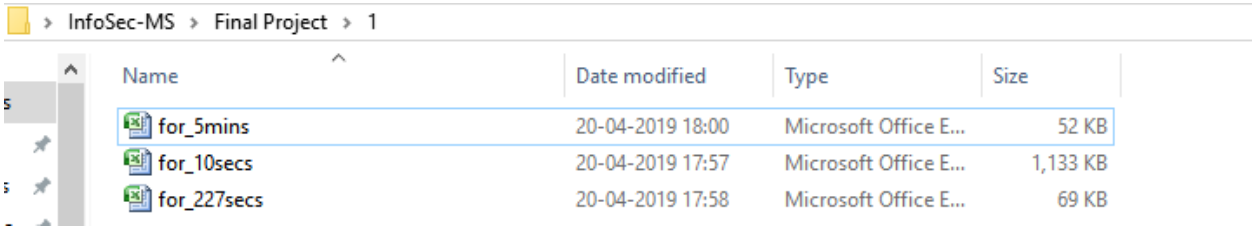
The technology used for this project is **Python 3.6** in "**Spyder editor**" from the **Anaconda** User Interface.

Approach:

In this project we use a parameter for creating a profile for each user, called the **"avginternetusage"** which is the ratio of **"docktets/Duration"**.

We split our analysis into 3 different time windows of 10 seconds, 227 seconds and 5 minutes.

Hence, we generate 3 different excel files per user dynamically representing the 3 different time-frames. There are a total of 54 users, hence 162 excel files are generated in total, dynamically. Below shown are the files generated for user 1.



Code flow:

Initially, we extract the given excel data files representing 54 users and set them to a global path for accessing them in the code.

Then we create the empty data frames for week 1 and week 2 for each user for the 10secs, 5mins and 227 secs. These new dataframes have the fields **"week"**,

“starttime”, **“endtime”** and **“avginternetusage”** which is the **“docktets/Duration”** ratio.

```
# -*- coding: utf-8 -*-
"""
Created on Tue Apr 16 01:13:31 2019

@author: Vineeth Reddy C
"""
import os
import glob
import pandas as pd
from pandasql import sqldf
import math
from scipy import stats

datafiles = glob.glob('C:/Users/Vineeth reddy/Desktop/InfoSec-MS/Information Security _ Privacy Material/*.xlsx')
path = ('C:/Users/Vineeth reddy/Desktop/InfoSec-MS/Information Security _ Privacy Material/')
print(len(datafiles))

df10_S_W1 = pd.DataFrame(columns= ['week','starttime','endtime','avginternetusage'])
df10_S_W2 = pd.DataFrame(columns= ['week','starttime','endtime','avginternetusage'])

df5_M_W1 = pd.DataFrame(columns= ['week','starttime','endtime','avginternetusage'])
df5_M_W2 = pd.DataFrame(columns= ['week','starttime','endtime','avginternetusage'])

df227_S_W1 = pd.DataFrame(columns= ['week','starttime','endtime','avginternetusage'])
df227_S_W2 = pd.DataFrame(columns= ['week','starttime','endtime','avginternetusage'])
```

Next, we find the timeframes the 10 second windows, here the online epoch converter is used just to find timestamp limits. A hardcoded epoch time of 8am, 4th Feb, 2013 is fixed as the low time stamp and a highest timestamp of 5pm, 8th Feb, 2013 is used, for the week 1 (Monday –Friday). Similarly, A hardcoded epoch time of 8am, 11th Feb, 2013 is fixed as the low time stamp and a high timestamp of 5pm, 15th Feb, 2013 is used, for the week 2 (Monday –Friday).

For every 10 seconds interval, a 10,000 epochs are added since they are milliseconds and when end of day is encountered (5pm) a 54000000 epochs are added which corresponds to the time between 5pm and the 8am of next day.

Finally, the data frames which are generated for 2 weeks are merged into one data frame.

This approach is the same for 5mins and 227 seconds windows, except a 300000 and 227000 epoch values are used for jumps.

```
# For 10 Seconds window
# Week1
lowTimeStampWeek1 = 1359982800000 # 8am, 4th Feb, 2013
highTimeStampWeek1 = 1360360800000 # 5pm, 8th Feb, 2013
cnt1 = 0
while lowTimeStampWeek1 < highTimeStampWeek1:
     df10_S_W1.loc[-1] = [1, lowTimeStampWeek1, ((lowTimeStampWeek1 + 10000) - 1), 0]
     df10_S_W1.index = df10_S_W1.index + 1
     lowTimeStampWeek1 = (lowTimeStampWeek1 + 10000)  #10secs intervals
     cnt1 = cnt1 + 1;
     if cnt1 == 3241: # skipping epochs if it is 5pm each day for the 10 seconds window
         lowTimeStampWeek1 = lowTimeStampWeek1 + 54000000
         cnt1 = 0

lowTimeStampWeek2 = 1360587600000
highTimeStampWeek2 = 1360965600000
cnt2 = 0
while lowTimeStampWeek2 < highTimeStampWeek2:
     df10_S_W2.loc[-1] = [2, lowTimeStampWeek2, ((lowTimeStampWeek2 + 10000) - 1), 0]
     df10_S_W2.index = df10_S_W2.index + 1
     lowTimeStampWeek2 = (lowTimeStampWeek2 + 10000)
     cnt2 = (cnt2 + 1)
     if cnt2 == 3241:
         lowTimeStampWeek2 = lowTimeStampWeek2 + 54000000
         cnt2 = 0

df_10S = pd.concat([df10_S_W1,df10_S_W2])
df_10S.sort_values(by=['starttime'])
```

We iterate over each data frame created, based on the **"Real First Packet"** for each user and store the **"docktets/Duration"** in the **"Internet Usage"** column. This followed by using the package **"sqldf"** for writing database queries on the dataframes, for optimizing the code to run much faster than using iterations.

```
user_id = 1
os.chdir(path)
for filename in os.listdir():
    user_10S = df_10S
    user_227S = df_227S
    user_5M = df_5M

    user_data = pd.read_excel(filename, sheetname='Sheet1')
    user_data = pd.DataFrame(user_data)

    #Filtering the data from 4th Feb to 15th Feb and removing all durations=0
    user_data = user_data[(user_data['Real First Packet'] >= 1359982800000) & (user_data['Real First Packet'] <= 1360965600000) & (user_d

    user_data = user_data.sort_values(by=['Real First Packet'])

    frame_row_size = len(user_data.axes[0])
    if frame_row_size > 0:
        user_data.loc[:,'InternetUsage'] = 0
        for i in user_data.iterrows():
            user_data.loc[:,'InternetUsage'] = user_data.loc[:,'doctets']/user_data.loc[:,'Duration']
        print(user_data.loc[:,'InternetUsage'])

        user_10S = sqldf("""SELECT a.week as week, a.starttime as starttime, a.endtime as endtime, case when d.InternetUsage is not null
        user_227S = sqldf("""SELECT u.week as week, u.starttime as starttime, u.endtime as endtime, case when d.InternetUsage is not null
        user_5M = sqldf("""SELECT m.week as week, m.starttime as starttime, m.endtime as endtime, case when d.InternetUsage is not null t

    os.makedirs(os.path.join('C:/Users/Vineeth reddy/Desktop/InfoSec-MS/Final Project/',  str(user_id)))

    user_10S.to_excel("C:/Users/Vineeth reddy/Desktop/InfoSec-MS/Final Project/"+str(user_id)+"/for_10secs.xlsx")
    user_227S.to_excel("C:/Users/Vineeth reddy/Desktop/InfoSec-MS/Final Project/"+str(user_id)+"/for_227secs.xlsx")
    user_5M.to_excel("C:/Users/Vineeth reddy/Desktop/InfoSec-MS/Final Project/"+str(user_id)+"/for_5mins.xlsx")
    user_id += 1
```

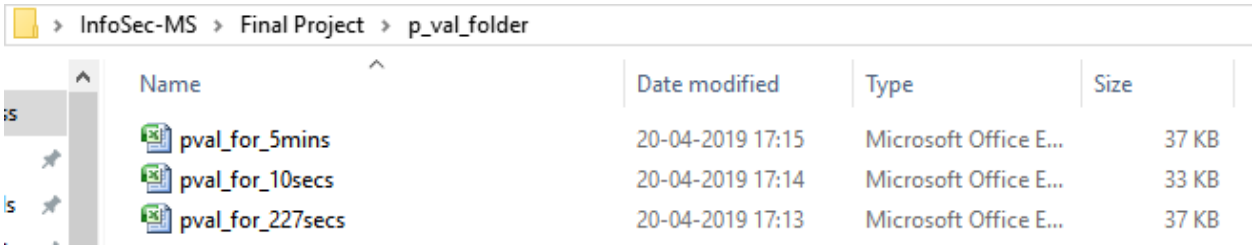
P-calculation:

We use functions **finding_z( )** and **finding_p( )** for calculating p and z values.

```
## p calculation
w, h = 54, 54;
p_compare_users = [[0 for x in range(w)] for y in range(h)]

def finding_z(r1a2a, r1a2b, r2a2b, N):
    rm2 = (math.pow(r1a2a,2) + math.pow(r1a2b,2)) / 2
    f = (1 - r2a2b) / (2 * (1 - rm2))
    h = (1 - (f * rm2)) / (1 - rm2)
    z1a2b = (0.5) * (math.log((1 + r1a2b) / (1 - r1a2b)))
    z1a2a = (0.5) * (math.log((1 + r1a2a) / (1 - r1a2a)))
    z = (z1a2a - z1a2b) * (math.sqrt(N - 3) / (2 * (1 - r2a2b) * h))
    return(z)

def finding_p(z):
    p = 0.3275911
    a1 = 0.254829592
    a2 = -0.284496736
    a3 = 1.421413741
    a4 = -1.453152027
    a5 = 1.061405429

    if z < 0.0:
        sign = -1
    else:
        sign = 1

    x = abs(z) / math.sqrt(2.0)

    t = 1.0 / (1.0 + p * x)

    erf = 1.0 - (((((a5 * t + a4) * t) + a3) * t + a2) * t + a1) * t * math.exp(-x * x)

    return(0.5 * (1.0 + sign * erf))
```

Another function **gen_user_data( )** is used for generating the correlations between all the 54 users. We capture the **"avginternetusage"** values for each user and create a list of data frames. We create a 54 * 54 matrix in the form of list of lists for storing the p-values.

Inside the double for-loop shown below we find the **Spearman's correlation** between the **User 'a' week 1** data and **User 'a' week 2** data which is stored in the **r1a2a** variable. Similarly, the **User 'a' week 1** data and **User 'b' week 2** data which is stored in the **r1a2b** variable and the **User 'a' week 2** data and **User 'b' week 2** data which is stored in the **r2a2b** variable.



Then these values are passed into the **finding_z( )** and **finding_p( )** functions and calculate the p-vales for the 3 different windows, 10secs, 227secs and 5mins as shown below.

```
def gen_user_data(file_name):
    print(file_name)
    userweek_1 = list()
    userweek_2 = list()
    for i in range(1,55):
        data_gen = pd.read_excel("C:/Users/Vineeth reddy/Desktop/InfoSec-MS/Final Project/"+str(i)+file_name)
        data_gen = pd.DataFrame(data_gen)
        userweek_1.append(data_gen[data_gen['week'] == 1].filter(['avginternetusage']))
        userweek_2.append(data_gen[data_gen['week'] == 2].filter(['avginternetusage']))
    for j in range(0,54):
        for k in range(0,54):
                r1a2a_res, p1a2a_res = stats.spearmanr(userweek_1[j],userweek_2[j])
                r1a2b_res, p1a2b_res = stats.spearmanr(userweek_1[j],userweek_2[k])
                r2a2b_res, p2a2b_res = stats.spearmanr(userweek_2[j],userweek_2[k])
                if r1a2a_res == 1:
                    r1a2a_res = 0.99
                if r1a2b_res == 1:
                    r1a2b_res = 0.99
                if r2a2b_res == 1:
                    r2a2b_res = 0.99
                z_val = finding_z(r1a2a_res, r1a2b_res, r2a2b_res, userweek_1[j].shape[0])
                #print(z_val)
                p_val = finding_p(z_val)
                #print(p_val)
                p_compare_users[j][k] = p_val
    final_data = pd.DataFrame(p_compare_users)
    final_data.to_excel("C:/Users/Vineeth reddy/Desktop/InfoSec-MS/Final Project/p_val_folder"+file_name)

os.makedirs('C:/Users/Vineeth reddy/Desktop/InfoSec-MS/Final Project/p_val_folder/')
gen_user_data("/for_10secs.xlsx")
gen_user_data("/for_227secs.xlsx")
gen_user_data("/for_5mins.xlsx")
```

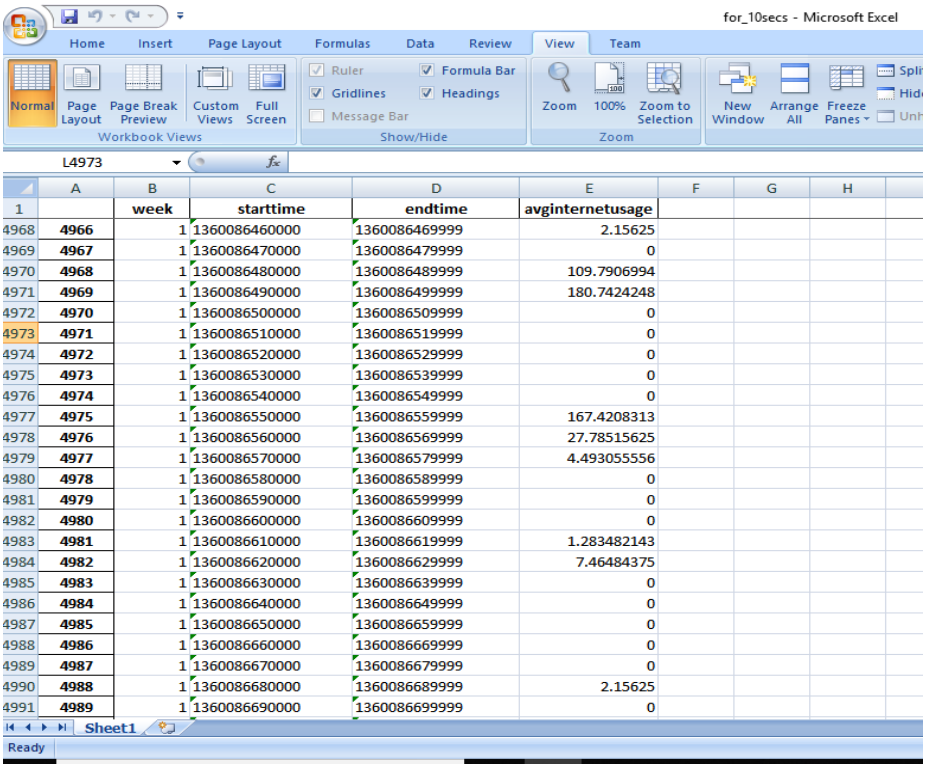
Results:

The code can be executed on any python editor, but you can use spyder in anaconda navigator and then click on run after changing the file path.

The results for the p-values and the user files generated can be viewed in the "Final Project" folder inside the zip file.

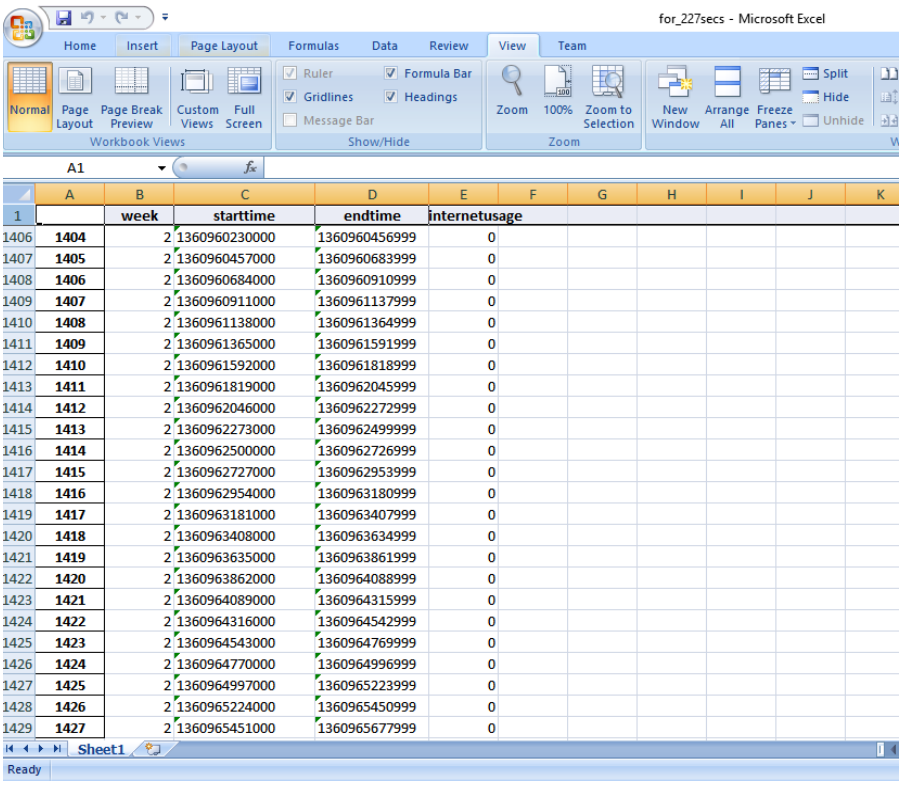
Shown below are files generated for the user 1, which store the dataframes for week 1 and 2 and contain **"avginternetusage"** which is **"docktet/Duration"**. 10 second window files have 32400 rows, 227 second windows have 1427 rows

and 5 minute windows have 1080 rows. Similarly all such files are generated for 54 users.



	A	B	C	D	E	F	G	H
1		week	starttime	endtime	avginternetusage			
4968	4966	1	1360086460000	1360086469999	2.15625			
4969	4967	1	1360086470000	1360086479999	0			
4970	4968	1	1360086480000	1360086489999	109.7906994			
4971	4969	1	1360086490000	1360086499999	180.7424248			
4972	4970	1	1360086500000	1360086509999	0			
4973	4971	1	1360086510000	1360086519999	0			
4974	4972	1	1360086520000	1360086529999	0			
4975	4973	1	1360086530000	1360086539999	0			
4976	4974	1	1360086540000	1360086549999	0			
4977	4975	1	1360086550000	1360086559999	167.4208313			
4978	4976	1	1360086560000	1360086569999	27.78515625			
4979	4977	1	1360086570000	1360086579999	4.493055556			
4980	4978	1	1360086580000	1360086589999	0			
4981	4979	1	1360086590000	1360086599999	0			
4982	4980	1	1360086600000	1360086609999	0			
4983	4981	1	1360086610000	1360086619999	1.283482143			
4984	4982	1	1360086620000	1360086629999	7.46484375			
4985	4983	1	1360086630000	1360086639999	0			
4986	4984	1	1360086640000	1360086649999	0			
4987	4985	1	1360086650000	1360086659999	0			
4988	4986	1	1360086660000	1360086669999	0			
4989	4987	1	1360086670000	1360086679999	0			
4990	4988	1	1360086680000	1360086689999	2.15625			
4991	4989	1	1360086690000	1360086699999	0			

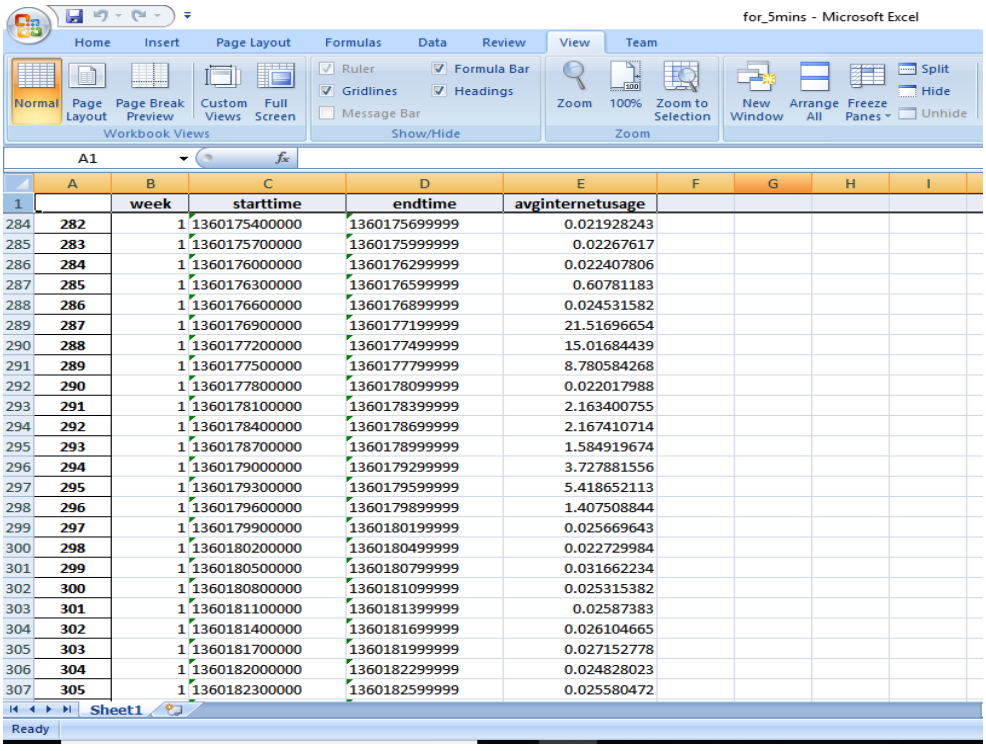
User 1’s 10secs-window



	A	week	starttime	endtime	internetusage
1406	1404	2	1360960230000	1360960456999	0
1407	1405	2	1360960457000	1360960683999	0
1408	1406	2	1360960684000	1360960910999	0
1409	1407	2	1360960911000	1360961137999	0
1410	1408	2	1360961138000	1360961364999	0
1411	1409	2	1360961365000	1360961591999	0
1412	1410	2	1360961592000	1360961818999	0
1413	1411	2	1360961819000	1360962045999	0
1414	1412	2	1360962046000	1360962272999	0
1415	1413	2	1360962273000	1360962499999	0
1416	1414	2	1360962500000	1360962726999	0
1417	1415	2	1360962727000	1360962953999	0
1418	1416	2	1360962954000	1360963180999	0
1419	1417	2	1360963181000	1360963407999	0
1420	1418	2	1360963408000	1360963634999	0
1421	1419	2	1360963635000	1360963861999	0
1422	1420	2	1360963862000	1360964088999	0
1423	1421	2	1360964089000	1360964315999	0
1424	1422	2	1360964316000	1360964542999	0
1425	1423	2	1360964543000	1360964769999	0
1426	1424	2	1360964770000	1360964996999	0
1427	1425	2	1360964997000	1360965223999	0
1428	1426	2	1360965224000	1360965450999	0
1429	1427	2	1360965451000	1360965677999	0

Sheet1
Ready

User 1’s 227secs-window



	A	week	starttime	endtime	avginternetusage
284	282	1	1360175400000	1360175699999	0.021928243
285	283	1	1360175700000	1360175999999	0.02267617
286	284	1	1360176000000	1360176299999	0.022407806
287	285	1	1360176300000	1360176599999	0.60781183
288	286	1	1360176600000	1360176899999	0.024531582
289	287	1	1360176900000	1360177199999	21.51696654
290	288	1	1360177200000	1360177499999	15.01684439
291	289	1	1360177500000	1360177799999	8.780584268
292	290	1	1360177800000	1360178099999	0.022017988
293	291	1	1360178100000	1360178399999	2.163400755
294	292	1	1360178400000	1360178699999	2.167410714
295	293	1	1360178700000	1360178999999	1.584919674
296	294	1	1360179000000	1360179299999	3.727881556
297	295	1	1360179300000	1360179599999	5.418652113
298	296	1	1360179600000	1360179899999	1.407508844
299	297	1	1360179900000	1360180199999	0.025669643
300	298	1	1360180200000	1360180499999	0.022729984
301	299	1	1360180500000	1360180799999	0.031662234
302	300	1	1360180800000	1360181099999	0.025315382
303	301	1	1360181100000	1360181399999	0.02587383
304	302	1	1360181400000	1360181699999	0.026104665
305	303	1	1360181700000	1360181999999	0.027152778
306	304	1	1360182000000	1360182299999	0.024828023
307	305	1	1360182300000	1360182599999	0.025580472

Sheet1
Ready

User 1’s 5mins-window

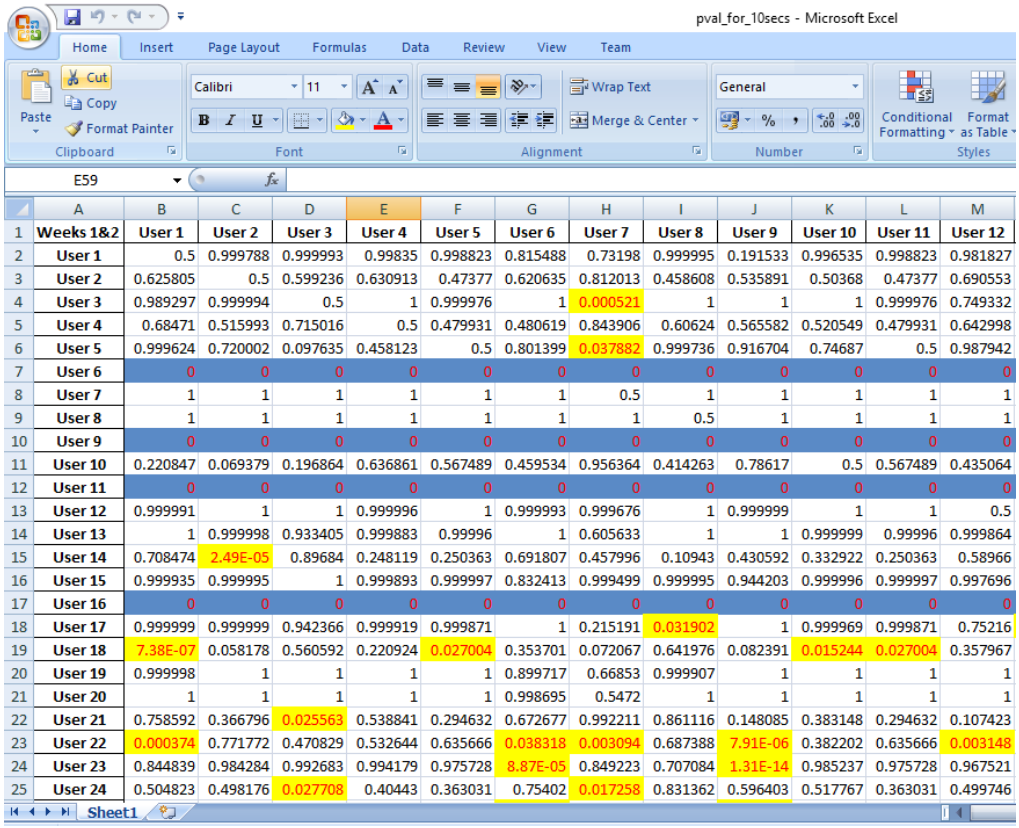
P-values:

Weeks 1 &2 are taken for the p-value generation. The P-values for 10secs, 227secs and 5mins are generated into 3 excel files.

The empty cells denote that there is no correlation between those users (NaN).

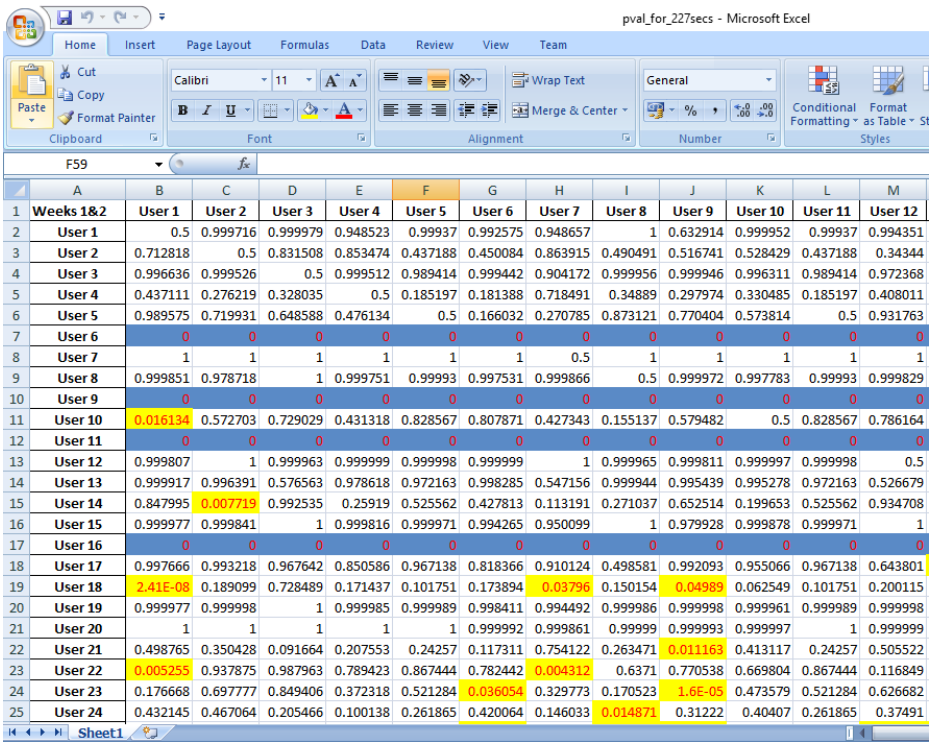
"Empty cells correspond to NaN"

The results for the p-values and the user files generated can be viewed in the "Final Project" folder inside the zip file.



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Weeks 1&2	User 1	User 2	User 3	User 4	User 5	User 6	User 7	User 8	User 9	User 10	User 11	User 12
2	User 1	0.5	0.999788	0.999993	0.99835	0.998823	0.815488	0.73198	0.999995	0.191533	0.996535	0.998823	0.981827
3	User 2	0.625805	0.5	0.599236	0.630913	0.47377	0.620635	0.812013	0.458608	0.535891	0.50368	0.47377	0.690553
4	User 3	0.989297	0.999994	0.5	1	0.999976	1	0.000521	1	1	1	0.999976	0.749332
5	User 4	0.68471	0.515993	0.715016	0.5	0.479931	0.480619	0.843906	0.60624	0.565582	0.520549	0.479931	0.642998
6	User 5	0.999624	0.720002	0.097635	0.458123	0.5	0.801399	0.037882	0.999736	0.916704	0.74687	0.5	0.987942
7	User 6	0	0	0	0	0	0	0	0	0	0	0	0
8	User 7	1	1	1	1	1	1	0.5	1	1	1	1	1
9	User 8	1	1	1	1	1	1	1	0.5	1	1	1	1
10	User 9	0	0	0	0	0	0	0	0	0	0	0	0
11	User 10	0.220847	0.069379	0.196864	0.636861	0.567489	0.459534	0.956364	0.414263	0.78617	0.5	0.567489	0.435064
12	User 11	0	0	0	0	0	0	0	0	0	0	0	0
13	User 12	0.999991	1	1	0.999996	1	0.999993	0.999676	1	0.999999	1	1	0.5
14	User 13	1	0.999998	0.933405	0.999883	0.99996	1	0.605633	1	1	0.999999	0.99996	0.999864
15	User 14	0.708474	2.49E-05	0.89684	0.248119	0.250363	0.691807	0.457996	0.10943	0.430592	0.332922	0.250363	0.58966
16	User 15	0.999935	0.999995	1	0.999893	0.999997	0.832413	0.999499	0.999995	0.944203	0.999996	0.999997	0.997696
17	User 16	0	0	0	0	0	0	0	0	0	0	0	0
18	User 17	0.999999	0.999999	0.942366	0.999919	0.999871	1	0.215191	0.031902	1	0.999969	0.999871	0.75216
19	User 18	7.38E-07	0.058178	0.560592	0.220924	0.027004	0.353701	0.072067	0.641976	0.082391	0.015244	0.027004	0.357967
20	User 19	0.999998	1	1	1	1	0.899717	0.66853	0.999907	1	1	1	1
21	User 20	1	1	1	1	1	0.998695	0.5472	1	1	1	1	1
22	User 21	0.758592	0.366796	0.025563	0.538841	0.294632	0.672677	0.992211	0.861116	0.148085	0.383148	0.294632	0.107423
23	User 22	0.000374	0.771772	0.470829	0.532644	0.635666	0.038318	0.003094	0.687388	7.91E-06	0.382202	0.635666	0.003148
24	User 23	0.844839	0.984284	0.992683	0.994179	0.975728	8.87E-05	0.849223	0.707084	1.31E-14	0.985237	0.975728	0.967521
25	User 24	0.504823	0.498176	0.027708	0.40443	0.363031	0.75402	0.017258	0.831362	0.596403	0.517767	0.363031	0.499746

P-Values for 10 secs window



pval_for_227secs - Microsoft Excel

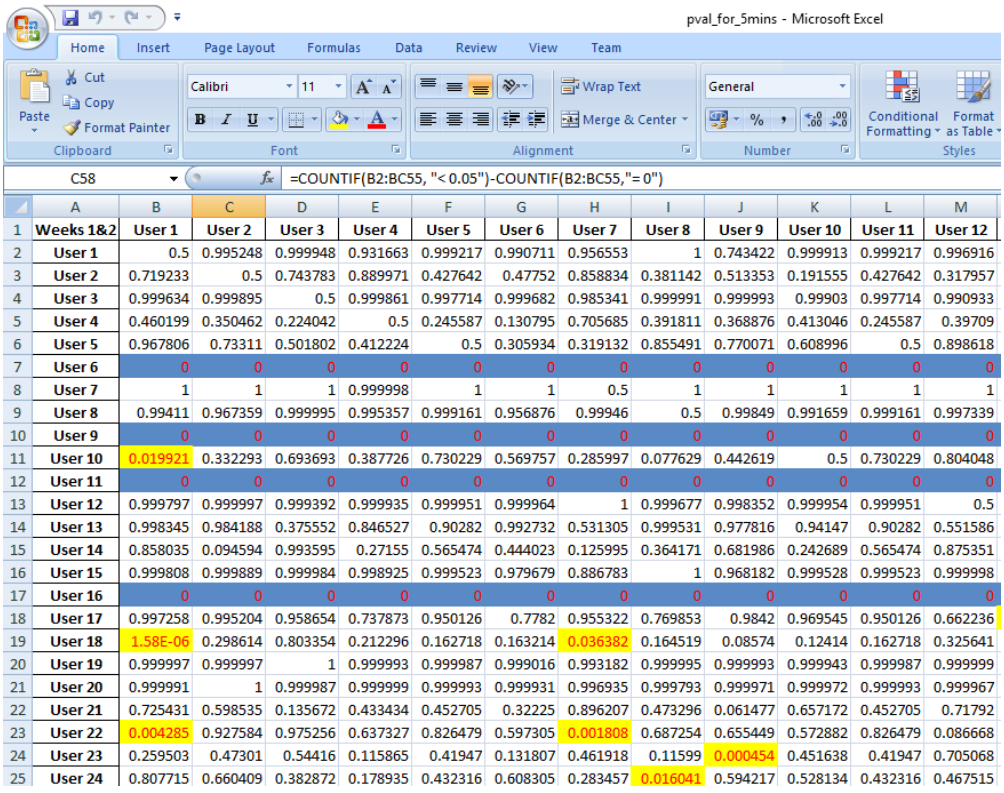
Home | Insert | Page Layout | Formulas | Data | Review | View | Team

F59

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Weeks 1&2	User 1	User 2	User 3	User 4	User 5	User 6	User 7	User 8	User 9	User 10	User 11	User 12
2	User 1	0.5	0.999716	0.999979	0.948523	0.99937	0.992575	0.948657	1	0.632914	0.999952	0.99937	0.994351
3	User 2	0.712818	0.5	0.831508	0.853474	0.437188	0.450084	0.863915	0.490491	0.516741	0.528429	0.437188	0.34344
4	User 3	0.996636	0.999526	0.5	0.999512	0.989414	0.999442	0.904172	0.999956	0.999946	0.996311	0.989414	0.972368
5	User 4	0.437111	0.276219	0.328035	0.5	0.185197	0.181388	0.718491	0.34889	0.297974	0.330485	0.185197	0.408011
6	User 5	0.989575	0.719931	0.648588	0.476134	0.5	0.166032	0.270785	0.873121	0.770404	0.573814	0.5	0.931763
7	User 6	0	0	0	0	0	0	0	0	0	0	0	0
8	User 7	1	1	1	1	1	1	0.5	1	1	1	1	1
9	User 8	0.999851	0.978718	1	0.999751	0.99993	0.997531	0.999866	0.5	0.999972	0.997783	0.99993	0.999829
10	User 9	0	0	0	0	0	0	0	0	0	0	0	0
11	User 10	0.016134	0.572703	0.729029	0.431318	0.828567	0.807871	0.427343	0.155137	0.579482	0.5	0.828567	0.786164
12	User 11	0	0	0	0	0	0	0	0	0	0	0	0
13	User 12	0.999807	1	0.999963	0.999999	0.999998	0.999999	1	0.999965	0.999811	0.999997	0.999998	0.5
14	User 13	0.999917	0.996391	0.576563	0.978618	0.972163	0.998285	0.547156	0.999944	0.995439	0.995278	0.972163	0.526679
15	User 14	0.847995	0.007719	0.992535	0.25919	0.525562	0.427813	0.113191	0.271037	0.652514	0.199653	0.525562	0.934708
16	User 15	0.999977	0.999841	1	0.999816	0.999971	0.994265	0.950099	1	0.979928	0.999878	0.999971	1
17	User 16	0	0	0	0	0	0	0	0	0	0	0	0
18	User 17	0.997666	0.993218	0.967642	0.850586	0.967138	0.818366	0.910124	0.498581	0.992093	0.955066	0.967138	0.643801
19	User 18	2.41E-08	0.189099	0.728489	0.171437	0.101751	0.173894	0.03796	0.150154	0.04989	0.062549	0.101751	0.200115
20	User 19	0.999977	0.999998	1	0.999985	0.999989	0.998411	0.994492	0.999986	0.999998	0.999961	0.999989	0.999998
21	User 20	1	1	1	1	1	0.999992	0.999861	0.99999	0.999993	0.999997	1	0.999999
22	User 21	0.498765	0.350428	0.091664	0.207553	0.24257	0.117311	0.754122	0.263471	0.011163	0.413117	0.24257	0.505522
23	User 22	0.005255	0.937875	0.987963	0.789423	0.867444	0.782442	0.004312	0.6371	0.770538	0.669804	0.867444	0.116849
24	User 23	0.176668	0.697777	0.849406	0.372318	0.521284	0.036054	0.329773	0.170523	1.6E-05	0.473579	0.521284	0.626682
25	User 24	0.432145	0.467064	0.205466	0.100138	0.261865	0.420064	0.146033	0.014871	0.31222	0.40407	0.261865	0.37491

Sheet1

P-Values for 227 secs window



pval_for_5mins - Microsoft Excel

Home | Insert | Page Layout | Formulas | Data | Review | View | Team

C58 =COUNTIF(B2:BC55, "< 0.05")-COUNTIF(B2:BC55,"= 0")

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Weeks 1&2	User 1	User 2	User 3	User 4	User 5	User 6	User 7	User 8	User 9	User 10	User 11	User 12
2	User 1	0.5	0.995248	0.999948	0.931663	0.999217	0.990711	0.956553	1	0.743422	0.999913	0.999217	0.996916
3	User 2	0.719233	0.5	0.743783	0.889971	0.427642	0.47752	0.858834	0.381142	0.513353	0.191555	0.427642	0.317957
4	User 3	0.999634	0.999895	0.5	0.999861	0.997714	0.999682	0.985341	0.999991	0.999993	0.99903	0.997714	0.990933
5	User 4	0.460199	0.350462	0.224042	0.5	0.245587	0.130795	0.705685	0.391811	0.368876	0.413046	0.245587	0.39709
6	User 5	0.967806	0.73311	0.501802	0.412224	0.5	0.305934	0.319132	0.855491	0.770071	0.608996	0.5	0.898618
7	User 6	0	0	0	0	0	0	0	0	0	0	0	0
8	User 7	1	1	1	0.999998	1	1	0.5	1	1	1	1	1
9	User 8	0.99411	0.967359	0.999995	0.995357	0.999161	0.956876	0.99946	0.5	0.99849	0.991659	0.999161	0.997339
10	User 9	0	0	0	0	0	0	0	0	0	0	0	0
11	User 10	0.019921	0.332293	0.693693	0.387726	0.730229	0.569757	0.285997	0.077629	0.442619	0.5	0.730229	0.804048
12	User 11	0	0	0	0	0	0	0	0	0	0	0	0
13	User 12	0.999797	0.999997	0.999392	0.999935	0.999951	0.999964	1	0.999677	0.998352	0.999954	0.999951	0.5
14	User 13	0.998345	0.984188	0.375552	0.846527	0.90282	0.992732	0.531305	0.999531	0.977816	0.94147	0.90282	0.551586
15	User 14	0.858035	0.094594	0.993595	0.27155	0.565474	0.444023	0.125995	0.364171	0.681986	0.242689	0.565474	0.875351
16	User 15	0.999808	0.999889	0.999984	0.998925	0.999523	0.979679	0.886783	1	0.968182	0.999528	0.999523	0.999998
17	User 16	0	0	0	0	0	0	0	0	0	0	0	0
18	User 17	0.997258	0.995204	0.958654	0.737873	0.950126	0.7782	0.955322	0.769853	0.9842	0.969545	0.950126	0.662236
19	User 18	1.58E-06	0.298614	0.803354	0.212296	0.162718	0.163214	0.036382	0.164519	0.08574	0.12414	0.162718	0.325641
20	User 19	0.999997	0.999997	1	0.999993	0.999987	0.999016	0.993182	0.999995	0.999993	0.999943	0.999987	0.999999
21	User 20	0.999991	1	0.999987	0.999999	0.999993	0.999931	0.996935	0.999793	0.999971	0.999972	0.999993	0.999967
22	User 21	0.725431	0.598535	0.135672	0.433434	0.452705	0.32225	0.896207	0.473296	0.061477	0.657172	0.452705	0.71792
23	User 22	0.004285	0.927584	0.975256	0.637327	0.826479	0.597305	0.001808	0.687254	0.655449	0.572882	0.826479	0.086668
24	User 23	0.259503	0.47301	0.54416	0.115865	0.41947	0.131807	0.461918	0.11599	0.000454	0.451638	0.41947	0.705068
25	User 24	0.807715	0.660409	0.382872	0.178935	0.432316	0.608305	0.283457	0.016041	0.594217	0.528134	0.432316	0.467515

P-Values for 5min window

From the 3 files there are trends where are no correlations between some users which are indicated by missing p-values. The larger the p-value the more is the chance that the user1 is indistinguishable from user 2. The smaller the p-value the more is the chance that the user1 is distinguishable from user 2. There are 199 p-values less than 0.05 in 10secs window, 129 p-values less than 0.05 in 227 secs window and 95 p-values less than 0.05 in 5mins window. Since there are maximum p-values less than 0.05 in 10-sec window, the **"10-sec window"** is the better choice for profiling the users.