Evaluation Tasks and Description

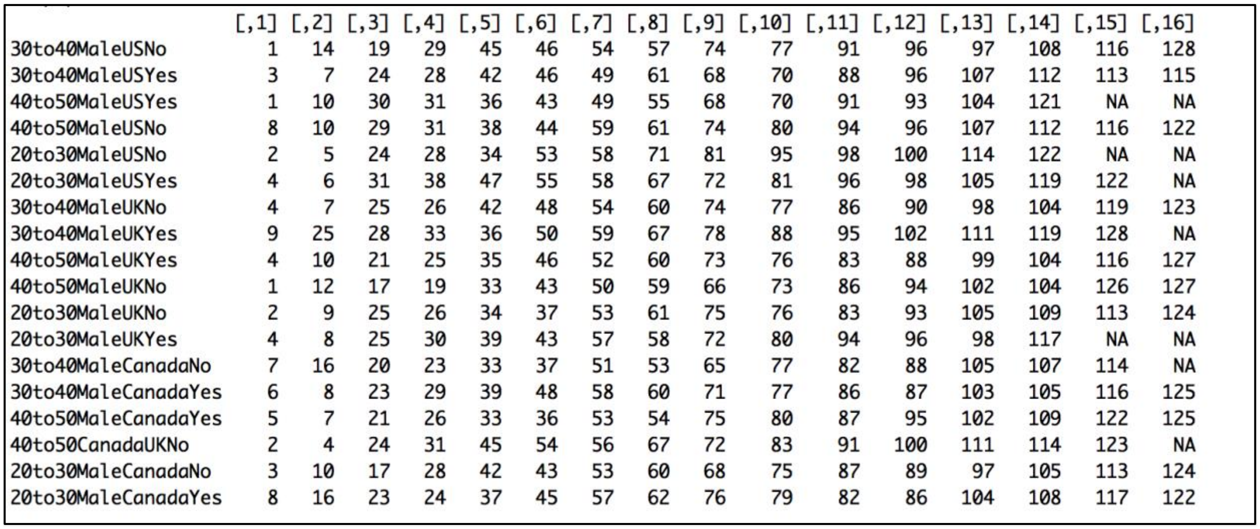
A description on what evaluation tasks you have done and the purpose of the tasks.

Data Analysis

Data used for analysis was https://www.kaggle.com/osmi/mental-health-in-tech-survey. Data analysis was done in R to identify the possible privacy attacks. Two different scenarios were introduced to better the consequences of attacks. Data description and analysis scenarios are described in datasource.pdf.

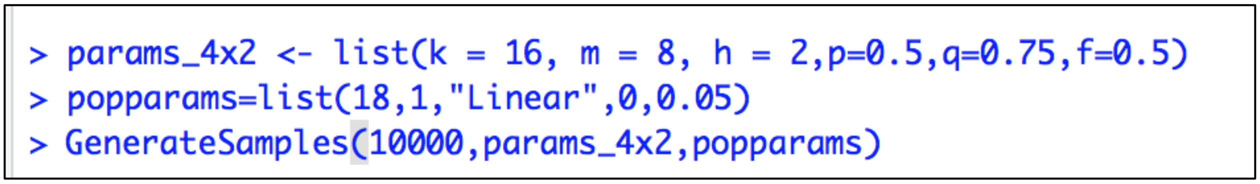
Implementation:

The implementation is done to demonstrate effectiveness of RAPPOR technique in preserving the privacy of individual data from the survey. The first step towards to implementation is to convert strings to bit values using hashing techniques. These bit values represent data from an individual. Bloom filters are applied to the user inputs which is followed RAPPOR encoding. This encoded data and parameters such as h, k m, p, q, f, used for encoding are sent to server.



```
                     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14] [,15] [,16]
30to40MaleUSNo          1   14   19   29   45   46   54   57   74    77    91    96    97   108   116   128
30to40MaleUSYes         3    7   24   28   42   46   49   61   68    70    88    96   107   112   113   115
40to50MaleUSYes         1   10   30   31   36   43   49   55   68    70    91    93   104   121    NA    NA
40to50MaleUSNo          8   10   29   31   38   44   59   61   74    80    94    96   107   112   116   122
20to30MaleUSNo          2    5   24   28   34   53   58   71   81    95    98   100   114   122    NA    NA
20to30MaleUSYes         4    6   31   38   47   55   58   67   72    81    96    98   105   119   122    NA
30to40MaleUKNo          4    7   25   26   42   48   54   60   74    77    86    90    98   104   119   123
30to40MaleUKYes         9   25   28   33   36   50   59   67   78    88    95   102   111   119   128    NA
40to50MaleUKYes         4   10   21   25   35   46   52   60   73    76    83    88    99   104   116   127
40to50MaleUKNo          1   12   17   19   33   43   50   59   66    73    86    94   102   104   126   127
20to30MaleUKNo          2    9   25   26   34   37   53   61   75    76    83    93   105   109   113   124
20to30MaleUKYes         4    8   25   30   39   43   57   58   72    80    94    96    98   117    NA    NA
30to40MaleCanadaNo      7   16   20   23   33   37   51   53   65    77    82    88   105   107   114    NA
30to40MaleCanadaYes     6    8   23   29   39   48   58   60   71    77    86    87   103   105   116   125
40to50MaleCanadaYes     5    7   21   26   33   36   53   54   75    80    87    95   102   109   122   125
40to50CanadaUKNo        2    4   24   31   45   54   56   67   72    83    91   100   111   114   123    NA
20to30MaleCanadaNo      3   10   17   28   42   43   53   60   68    75    87    89    97   105   113   124
20to30MaleCanadaYes     8   16   23   24   37   45   57   62   76    79    82    86   104   108   117   122
```

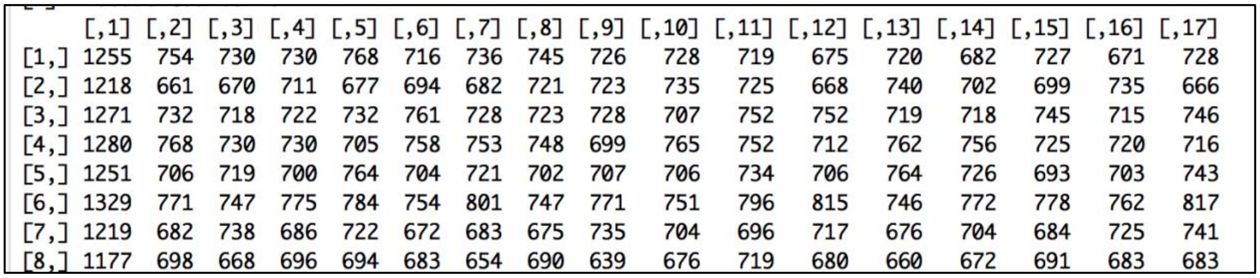
Fig. 1 Bloom filter bits vs. input – Bloom filter is represented by 16 bits (k). It means each row of the survey is represented as 16 bits.



```
> params_4x2 <- list(k = 16, m = 8, h = 2,p=0.5,q=0.75,f=0.5)
> popparams=list(18,1,"Linear",0,0.05)
> GenerateSamples(10000,params_4x2,popparams)
```

Fig. 2 Encoding input data

GenerateSamples method mentioned in the screenshot (Fig. 2) generates 10000 samples for user values from survey and perform encoding on the data. Results of RAPPOR encoding is given in below screenshot (Fig.3). It is a matrix of 17 candidate strings and 8 cohorts. Each value in the matrix represents rappor encoded values for each candidate string.

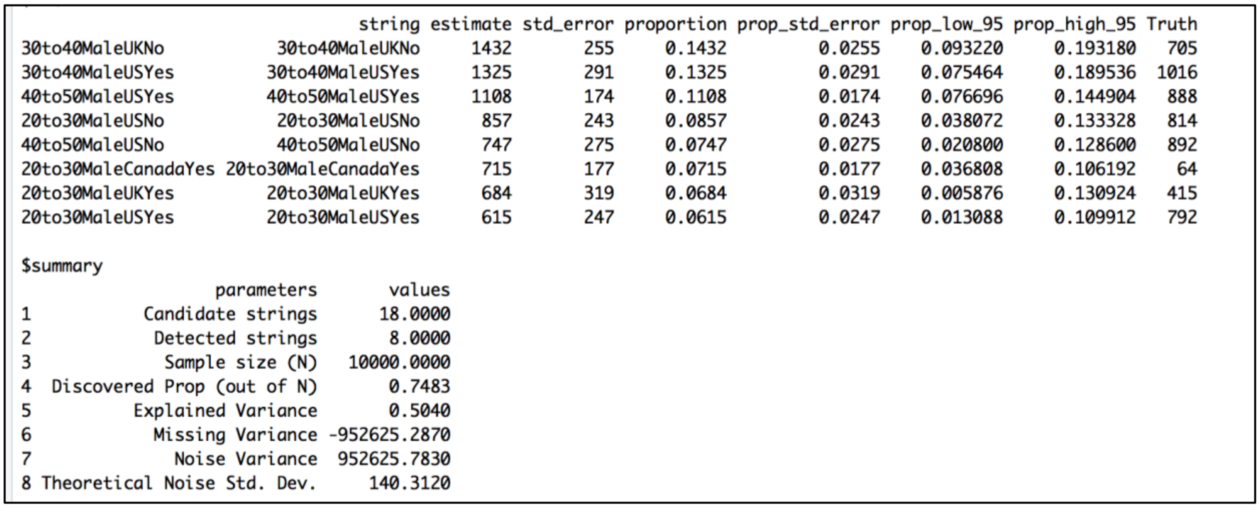


```
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14] [,15] [,16] [,17]
[1,] 1255  754  730  730  768  716  736  745  726   728   719   675   720   682   727   671   728
[2,] 1218  661  670  711  677  694  682  721  723   735   725   668   740   702   699   735   666
[3,] 1271  732  718  722  732  761  728  723  728   707   752   752   719   718   745   715   746
[4,] 1280  768  730  730  705  758  753  748  699   765   752   712   762   756   725   720   716
[5,] 1251  706  719  700  764  704  721  702  707   706   734   706   764   726   693   703   743
[6,] 1329  771  747  775  784  754  801  747  771   751   796   815   746   772   778   762   817
[7,] 1219  682  738  686  722  672  683  675  735   704   696   717   676   704   684   725   741
[8,] 1177  698  668  696  694  683  654  690  639   676   719   680   660   672   691   683   683
```

Fig. 3 RAPPOR Encoded data

This technique can preserve data privacy by sending the encoded string to server and not the actual data. Now that all the values are encoded, it is followed by decode function which could be used for publishing and for further statistical analysis.

RAPPOR encoded values, Bloom filter map (Fig. 1), parameters (Fig. 2) are sent to the decode function. The obtained results are given in the below screen shots

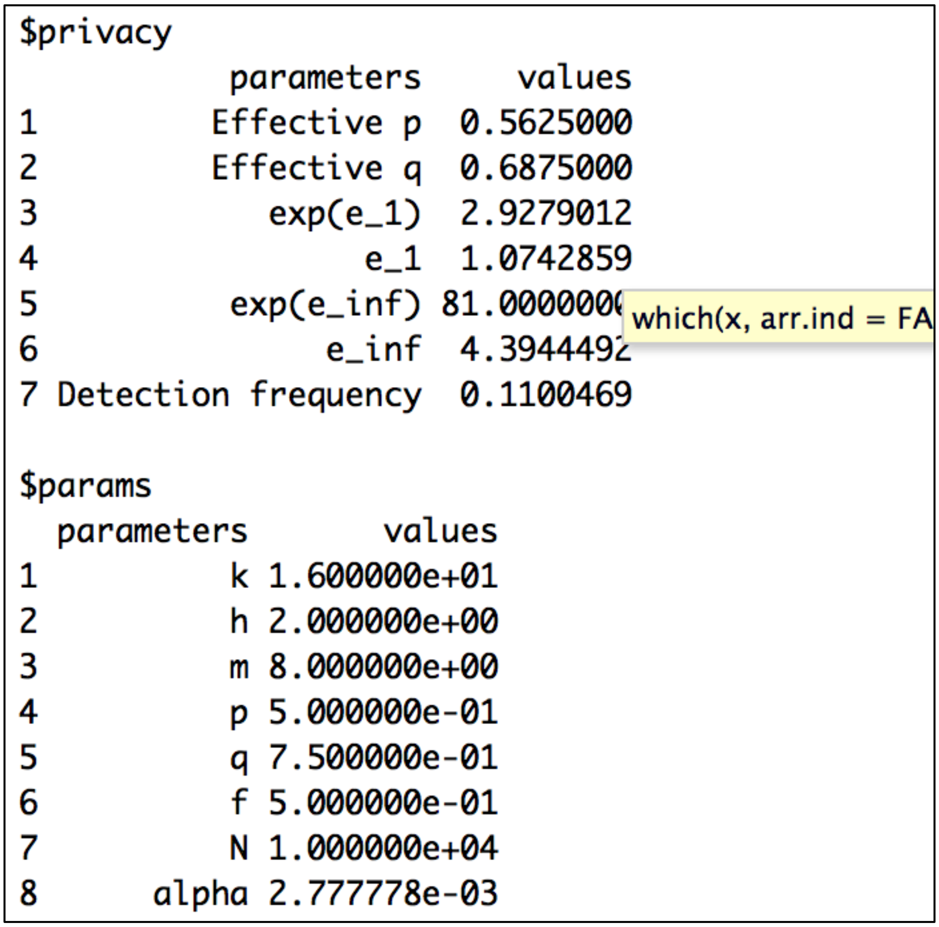


```
                                string estimate std_error proportion prop_std_error prop_low_95 prop_high_95 Truth
30to40MaleUKNo             30to40MaleUKNo     1432       255     0.1432         0.0255    0.093220     0.193180   705
30to40MaleUSYes           30to40MaleUSYes     1325       291     0.1325         0.0291    0.075464     0.189536  1016
40to50MaleUSYes           40to50MaleUSYes     1108       174     0.1108         0.0174    0.076696     0.144904   888
20to30MaleUSNo             20to30MaleUSNo      857       243     0.0857         0.0243    0.038072     0.133328   814
40to50MaleUSNo             40to50MaleUSNo      747       275     0.0747         0.0275    0.020800     0.128600   892
20to30MaleCanadaYes 20to30MaleCanadaYes      715       177     0.0715         0.0177    0.036808     0.106192    64
20to30MaleUKYes           20to30MaleUKYes      684       319     0.0684         0.0319    0.005876     0.130924   415
20to30MaleUSYes           20to30MaleUSYes      615       247     0.0615         0.0247    0.013088     0.109912   792

$summary
                  parameters        values
1          Candidate strings       18.0000
2           Detected strings        8.0000
3            Sample size (N)    10000.0000
4  Discovered Prop (out of N)       0.7483
5         Explained Variance        0.5040
6           Missing Variance  -952625.2870
7             Noise Variance   952625.7830
8 Theoretical Noise Std. Dev.     140.3120
```

Fig. 4 Truth vs. Estimate

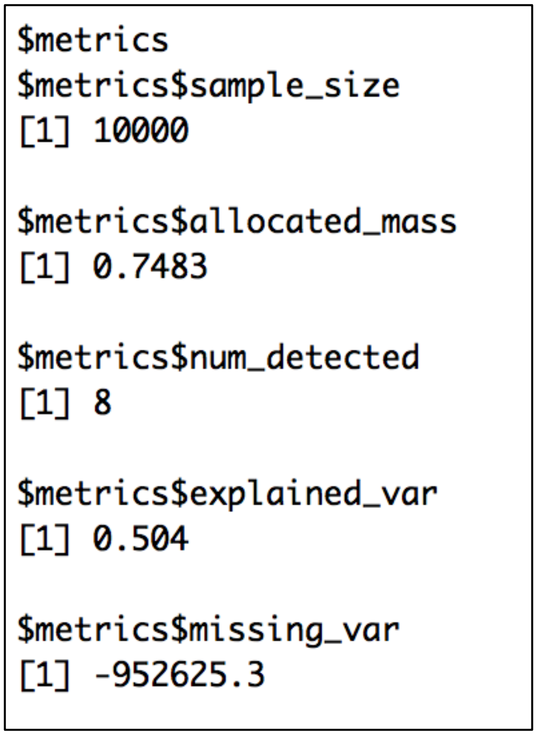
From Fig. 4, it is seen that the first input string (30to40MaleUKNo) is sent to the server 705 times. However, the estimated value by the decode function is 1432. Std error between these values is 255 and the corresponding proportion is 0.0255 which is well below the significance level (alpha = 0.05). It is inferred from the above results that the actual data is privacy preserved and could also effectively used for obtaining statistics.



```
$privacy
           parameters     values
1         Effective p  0.5625000
2         Effective q  0.6875000
3            exp(e_1)  2.9279012
4                 e_1  1.0742859
5          exp(e_inf) 81.0000000
6               e_inf  4.3944492
7 Detection frequency  0.1100469

$params
  parameters       values
1          k 1.600000e+01
2          h 2.000000e+00
3          m 8.000000e+00
4          p 5.000000e-01
5          q 7.500000e-01
6          f 5.000000e-01
7          N 1.000000e+04
8      alpha 2.777778e-03
```

Fig. 5 Privacy parameters used



```
$metrics
$metrics$sample_size
[1] 10000

$metrics$allocated_mass
[1] 0.7483

$metrics$num_detected
[1] 8

$metrics$explained_var
[1] 0.504

$metrics$missing_var
[1] -952625.3
```

Fig. 6 Results/ metrics

In Fig. 6, the metric $num_detected represents the number of unique candidate strings obtained after decoding. When the sample size is less, this metric will mostly be zero. This means the server could not decode any survey responses and hence statistical analysis on that data cannot be performed. Therefore, it is often recommended to increase the sample size to few thousands before encoding. For the purpose of this demonstration, I have used 10000 samples. (Refer Fig. 2)

Reference:

Source code obtained from: https://github.com/google/rappor.