

southpark-sdc-key

EDS213

2023-05-10

South Park Elementary School Data



Our Clients

Mayor McDaniels and Peter Charles (aka PC Principal) are concerned that even after removing direct identifiers such as names, SSNs, and IDs, students may still be easily re-identified in the yearly assessment dataset and have their math and reading scores revealed. For example, everyone in school knows that Tolkien Williams is the wealthiest kid in the whole town, whereas Kenny and his sister Karen are from a very poor family.

They have requested our assistance to compute this risk of disclosure, implement strategies to minimize it, and determine information loss for the anonymized dataset they would like to make public to other school board members*. They asked for our help, and we will be using the `sdcmicro` package for this purpose.

In summary, our client has three main questions to for us (and none of them involve finding out who keeps killing Keny and how come he keeps coming back to life):

Q1. What is the level of disclosure risk associated with this dataset?

Q2. How can the risk of re-identification be significantly reduced?

Q3. What would be the utility and information loss after implementing the anonymization strategies?

*Caveat: We have a relative small dataset for this exercise (rows and columns, so we can't strive for some of the thresholds recommended in the literature.

Packages & Data

```
# Required Package
library(sdcMicro)

# Dataset
data <- read.csv("southpark-sdc.csv")
```

Taking a closer look at the variables included in this dataset

```
# Read the CSV dataset into a data frame
df <- read.csv("southpark-sdc.csv")

# Show the list of variable names
options(scipen = 999) #remove scientific notation, if any
head(df)
```

```
##      zip      stu_id      ssn      name      dob age sex race
## 1 80220 8206630976 998126245    Stan Marsh 10/19/2012  10 Male White
## 2 80220 6555504757 807281100 Kyle Broflovski 05/26/2012  10 Male White
## 3 80220 5737953702 890807948 Kenny McCormick 03/12/2011  11 Male White
## 4 80220 5705942436 991920659    Eric Cartman 07/01/2012  10 Male White
## 5 80220 2809004240 921479968    Butters Scotch 11/11/2012  10 Male White
## 6 80220 4486369132 804989533    Clyde Donovan 04/10/2012  10 Male White
##      ethn snap      income learn_dis phys_dis math_sc read_sc
## 1 Non-hispanic    0 200,000-249,999      0      0      299      300
## 2 Non-hispanic    0 100,000-149,999      0      0      209      209
## 3 Non-hispanic    1  10,000-24,999      0      0      200      201
## 4 Non-hispanic    0  75,000-99,999      0      0      211      215
## 5 Non-hispanic    0  75,000-99,999      0      0      224      230
## 6 Non-hispanic    0  75,000-99,999      0      0      213      227
```

```
str(df)
```

```
## 'data.frame':    100 obs. of  15 variables:
## $ zip      : int  80220 80220 80220 80220 80220 80220 80220 80221 80220 80220 ...
## $ stu_id   : num  8206630976 6555504757 5737953702 5705942436 2809004240 ...
## $ ssn      : int  998126245 807281100 890807948 991920659 921479968 804989533 854569
146 761499326 925072083 772439783 ...
## $ name     : chr  "Stan Marsh" "Kyle Broflovski" "Kenny McCormick" "Eric Cartman"
...
## $ dob      : chr  "10/19/2012" "05/26/2012" "03/12/2011" "07/01/2012" ...
## $ age      : int  10 10 11 10 10 10 9 10 10 10 ...
## $ sex      : chr  "Male" "Male" "Male" "Male" ...
## $ race     : chr  "White" "White" "White" "White" ...
## $ ethn     : chr  "Non-hispanic" "Non-hispanic" "Non-hispanic" "Non-hispanic" ...
## $ snap     : int  0 0 1 0 0 0 0 0 0 0 ...
## $ income   : chr  "200,000-249,999" "100,000-149,999" "10,000-24,999" "75,000-99,99
9" ...
## $ learn_dis: int  0 0 0 0 0 0 0 0 0 1 ...
## $ phys_dis : int  0 0 0 0 0 0 0 0 0 1 ...
## $ math_sc  : int  299 209 200 211 224 213 204 202 202 205 ...
## $ read_sc  : int  300 209 201 215 230 227 210 214 222 225 ...
```

Data Prep - Converting variables

As we can see, we will need to convert some of the variables first.

The stu-id, SSN, name and dob will be removed soon from the dataset as they are direct identifiers.

Let's focus on the remaining ones that should be converted before we can proceed.

```
fname = "southpark-sdc.csv"
file <- read.csv(fname)
file <- varToFactor(obj=file, var=c("zip","age", "sex","race","ethn", "snap", "income",
"learn_dis","phys_dis"))
file <- varToNumeric(obj = file, var=c("math_sc","read_sc"))
```

Q1. What is the level of disclosure risk associated with this dataset?

To answer this question we have to set up an SDC problem. In other words we must select variables and create an object of class *sdcMicroObj* for the SDC process in *R*.

```
# Select variables for creating sdcMicro object
# All variable names should correspond to the names in the data file
# select categorical key variables - aka quasi-identifiers

sdcInitial <- createSdcObj(dat=file,
                           keyVars=c("zip","age", "sex","race","ethn", "snap", "income", "le
arn_dis","phys_dis"),
                           numVars=c("stu_id","math_sc","read_sc"),
                           weightVar=NULL,
                           hhId=NULL,
                           strataVar=NULL,
                           pramVars=NULL,
                           excludeVars=c("ssn","name","dob"),
                           seed=0,
                           randomizeRecords=FALSE,
                           alpha=c(1))

# Summary of object
sdcInitial
```

```
## The input dataset consists of 100 rows and 12 variables.
##
## The following variables have been deleted are not available in the output dataset:
## --> ssn
## --> name
## --> dob
##
##
## --> Categorical key variables: zip, age, sex, race, ethn, snap, income, learn_dis,
phys_dis
## --> Numerical key variables: stu_id, math_sc, read_sc
## -----
```

```
## Information on categorical key variables:
##
## Reported is the number, mean size and size of the smallest category >0 for recoded va
riables.
## In parenthesis, the same statistics are shown for the unmodified data.
## Note: NA (missings) are counted as seperate categories!
```

```
## Key Variable Number of categories      Mean size      Size of smallest (>0)
##      zip      3 (3)      33.333 (33.333)      25
##      age      6 (6)      16.667 (16.667)      1
##      sex      2 (2)      50.000 (50.000)      42
##      race     4 (4)      32.333 (32.333)      10
##      ethn     2 (2)      50.000 (50.000)      9
##      snap     2 (2)      50.000 (50.000)      5
##      income    5 (5)      20.000 (20.000)      1
##      learn_dis  2 (2)      50.000 (50.000)      12
##      phys_dis  2 (2)      50.000 (50.000)      6
##
## (25)
## (1)
## (42)
## (10)
## (9)
## (5)
## (1)
## (12)
## (6)
```

```
## -----
```

```
## Infos on 2/3-Anonymity:
##
## Number of observations violating
## - 2-anonymity: 70 (70.000%)
## - 3-anonymity: 84 (84.000%)
## - 5-anonymity: 90 (90.000%)
##
## -----
```

```
## Numerical key variables: stu_id, math_sc, read_sc
##
## Disclosure risk is currently between [0.00%; 100.00%]
##
## Current Information Loss:
## - IL1: 0.00
## - Difference of Eigenvalues: 0.000%
## -----
```

Check the results below, and the number of observations that violate 2-5 anonymity. What does that mean?

Time to calculate the risk of re-identification for the entire dataset

```
# The treshold depends on the size of the dataset and the access control (conservative number for large surveys are 0.04)
sdcInitial@risk$global$risk
```

```
## [1] 0.81
```

Over 81%? That is terrible! Let's see if we can get that lowered to less than 15% and a $k=5$.

We have to get some work done to reduce that. But that would be the first answer to our clients.

We can inspect this issue a little further before moving to the second question.

Which observations/subjects have a higher risk to be re-identified?

```
sdcInitial@risk$individual
```

##		risk	fk	Fk
##	[1,]	0.5000000	2	2
##	[2,]	0.2000000	5	5
##	[3,]	1.0000000	1	1
##	[4,]	0.2000000	5	5
##	[5,]	0.2000000	5	5
##	[6,]	0.2000000	5	5
##	[7,]	1.0000000	1	1
##	[8,]	1.0000000	1	1
##	[9,]	1.0000000	1	1
##	[10,]	1.0000000	1	1
##	[11,]	1.0000000	1	1
##	[12,]	0.3333333	3	3
##	[13,]	1.0000000	1	1
##	[14,]	1.0000000	1	1
##	[15,]	1.0000000	1	1
##	[16,]	0.5000000	2	2
##	[17,]	0.3333333	3	3
##	[18,]	0.5000000	2	2
##	[19,]	0.5000000	2	2
##	[20,]	0.2000000	5	5
##	[21,]	1.0000000	1	1
##	[22,]	1.0000000	1	1
##	[23,]	0.5000000	2	2
##	[24,]	1.0000000	1	1
##	[25,]	1.0000000	1	1
##	[26,]	1.0000000	1	1
##	[27,]	1.0000000	1	1
##	[28,]	0.5000000	2	2
##	[29,]	1.0000000	1	1
##	[30,]	1.0000000	1	1
##	[31,]	0.2000000	5	5
##	[32,]	1.0000000	1	1
##	[33,]	1.0000000	1	1
##	[34,]	1.0000000	1	1
##	[35,]	1.0000000	1	1
##	[36,]	0.3333333	3	3
##	[37,]	1.0000000	1	1
##	[38,]	1.0000000	1	1
##	[39,]	1.0000000	1	1
##	[40,]	0.5000000	2	2
##	[41,]	1.0000000	1	1
##	[42,]	1.0000000	1	1
##	[43,]	1.0000000	1	1
##	[44,]	1.0000000	1	1
##	[45,]	1.0000000	1	1
##	[46,]	0.2000000	5	5
##	[47,]	1.0000000	1	1
##	[48,]	1.0000000	1	1
##	[49,]	1.0000000	1	1
##	[50,]	1.0000000	1	1
##	[51,]	1.0000000	1	1

```

## [52,] 1.0000000 1 1
## [53,] 0.2000000 5 5
## [54,] 1.0000000 1 1
## [55,] 1.0000000 1 1
## [56,] 1.0000000 1 1
## [57,] 1.0000000 1 1
## [58,] 0.5000000 2 2
## [59,] 1.0000000 1 1
## [60,] 1.0000000 1 1
## [61,] 1.0000000 1 1
## [62,] 1.0000000 1 1
## [63,] 0.5000000 2 2
## [64,] 1.0000000 1 1
## [65,] 1.0000000 1 1
## [66,] 0.3333333 3 3
## [67,] 1.0000000 1 1
## [68,] 0.5000000 2 2
## [69,] 1.0000000 1 1
## [70,] 1.0000000 1 1
## [71,] 1.0000000 1 1
## [72,] 1.0000000 1 1
## [73,] 1.0000000 1 1
## [74,] 1.0000000 1 1
## [75,] 1.0000000 1 1
## [76,] 0.5000000 2 2
## [77,] 1.0000000 1 1
## [78,] 0.5000000 2 2
## [79,] 1.0000000 1 1
## [80,] 1.0000000 1 1
## [81,] 1.0000000 1 1
## [82,] 1.0000000 1 1
## [83,] 1.0000000 1 1
## [84,] 0.5000000 2 2
## [85,] 1.0000000 1 1
## [86,] 1.0000000 1 1
## [87,] 1.0000000 1 1
## [88,] 1.0000000 1 1
## [89,] 1.0000000 1 1
## [90,] 1.0000000 1 1
## [91,] 0.2000000 5 5
## [92,] 0.3333333 3 3
## [93,] 1.0000000 1 1
## [94,] 1.0000000 1 1
## [95,] 0.2000000 5 5
## [96,] 1.0000000 1 1
## [97,] 1.0000000 1 1
## [98,] 0.5000000 2 2
## [99,] 1.0000000 1 1
## [100,] 0.3333333 3 3

```

How many combinations of key variables each record have?

```
#Categorical variable risk
#Frequency of the particular combination of key variables (quasi-identifiers) for each record in the sample
freq(sdcInitial, type = 'fk')
```

```
##      [1] 2 5 1 5 5 5 1 1 1 1 1 3 1 1 1 2 3 2 2 5 1 1 2 1 1 1 1 2 1 1 5 1 1 1 1 3 1
##      [38] 1 1 2 1 1 1 1 1 5 1 1 1 1 1 1 5 1 1 1 1 2 1 1 1 1 2 1 1 3 1 2 1 1 1 1 1 1
##      [75] 1 2 1 2 1 1 1 1 1 2 1 1 1 1 1 1 5 3 1 1 5 1 1 2 1 3
```

Q2. How can the risk of re-identification be significantly reduced?

We learned that there are different techniques to de-identify and anonymize datasets.

First, let's use some non-perturbative methods such as global recoding and top and bottom coding techniques.

Income

As mentioned before, the household income of some students may pose a risk to their privacy in this dataset. So let's see if using top and bottom recoding could help reducing that risk.

```
# Frequencies of income before recoding
table(sdcInitial@manipKeyVars$income)
```

```
##
##      10,000-24,999 100,000-149,999 200,000-249,999      500,000+      75,000-99,999
##              2              39              24              1              34
```

```
## Recode variable income (top coding)
sdcInitial <- groupAndRename(obj= sdcInitial, var= c("income"), before=c("200,000-249,999", "500,000+"), after=c("200,000+"))

## Recode variable income (bottom coding)
sdcInitial <- groupAndRename(obj= sdcInitial, var= c("income"), before=c("10,000-24,999", "75,000-99,999"), after=c("10,000-99,999"))
```

Age

```
# Frequencies of age before recoding
table(sdcInitial@manipKeyVars$age)
```

```
##
##      8   9 10 11 12 13
##      1 28 50 19   1   1
```



```
#Recode Age
```

```
sdInitial <- groupAndRename(obj= sdInitial, var= c("age"), before=c("8","9","10"), after=c("8-10"))  
sdInitial <- groupAndRename(obj= sdInitial, var= c("age"), before=c("11","12","13"), after=c("11-13"))
```

Note: Undoing things

```
# Important note: If the results are reassigned to the same sdcMicro object, it is possible to undo the last step in the SDC process. Using:  
# sdInitial <- undolast(sdInitial)  
# It might be helpful to tune some parameters. The results of the last step, however, will be lost after undoing that step.  
# We can also choose to assign results to a new sdcMicro object this time, using:  
# sdc1 <- functionName(sdInitial) specially if you anticipate creating multiple sdc problems to test out. Otherwise, you can delete the object and re-run the code when needed
```

Let's see if those steps lowered the risk of re-identification of subjects.

```
sdInitial@risk$global$risk
```

```
## [1] 0.6941667
```

```
sum(sdInitial@risk$individual[,1] > 0.05)
```

```
## [1] 100
```

```
print(sdInitial, 'kAnon')
```

```
## Infos on 2/3-Anonymity:  
##  
## Number of observations violating  
## - 2-anonymity: 56 (56.000%) | in original data: 70 (70.000%)  
## - 3-anonymity: 69 (69.000%) | in original data: 84 (84.000%)  
## - 5-anonymity: 87 (87.000%) | in original data: 90 (90.000%)  
##  
## -----
```

Only a tiny improvement compared to the original dataset. Let's try something else.

Time for a more powerful technique. Let's use the k-anonymization function!

```
#Local suppression to obtain k-anonymity
sdcInitial <- kAnon(sdcInitial, k=c(5))

# Setting the parameters that we are aiming for at least 5 observations sharing the same
attributes in the dataset.

#Alternatively, we could have set the order of importance for each keyvariables
#sdcInitial <- kAnon(sdcInitial, importance=c(9,5,6,7,8,4,3,1,2), k=c(5))
```

More on importance (pg. 50): <https://cran.r-project.org/web/packages/sdcMicro/sdcMicro.pdf> (<https://cran.r-project.org/web/packages/sdcMicro/sdcMicro.pdf>)

Time to check it again:

```
sdcInitial@risk$global$risk
```

```
## [1] 0.1030652
```

```
print(sdcInitial, 'kAnon')
```

```
## Infos on 2/3-Anonymity:
##
## Number of observations violating
## - 2-anonymity: 0 (0.000%) | in original data: 70 (70.000%)
## - 3-anonymity: 0 (0.000%) | in original data: 84 (84.000%)
## - 5-anonymity: 0 (0.000%) | in original data: 90 (90.000%)
##
## -----
```

Alright! We managed lower the risk of identification from 81% to about 10% and now we have 0 observations violating 5-anonymity! We can tell our clients we used some recoding, but suppression via k-anonymity was necessary to improve the privacy level of this dataset.

Q3. What would be the utility and information loss after implementing anonymization strategies?

Time to measure the utility and information loss for the anonymized dataset.

```
#First we retrieve the total number of suppressions for each categorical key variable
print(sdcInitial, 'ls')
```

```
## Local suppression:
```

```
##      KeyVar | Suppressions (#) | Suppressions (%)
##      zip   |          9      |          9.000
##      age   |          2      |          2.000
##      sex   |          2      |          2.000
##      race  |         23      |         23.000
##      ethn  |          6      |          6.000
##      snap  |          5      |          5.000
##      income|         62      |         62.000
##  learn_dis |          4      |          4.000
##  phys_dis  |          6      |          6.000
```

```
## -----
```

```
#We can also compare the number of NAs before and after our interventions
# Store the names of all categorical key variables in a vector
namesKeyVars <- names(sdcInitial@manipKeyVars)

# Matrix to store the number of missing values (NA) before and after anonymization
NAcount <- matrix(NA, nrow = 2, ncol = length(namesKeyVars))
colnames(NAcount) <- c(paste0('NA', namesKeyVars)) # column names
rownames(NAcount) <- c('initial', 'treated') # row names

# NA count in all key variables (NOTE: only those coded NA are counted)
for(i in 1:length(namesKeyVars)) {
  NAcount[1, i] <- sum(is.na(sdcInitial@origData[,namesKeyVars[i]]))
  NAcount[2, i] <- sum(is.na(sdcInitial@manipKeyVars[,i]))}

# Show results
NAcount
```

```
##      NAzip NAage NAsex NArace NAethn NAsnap NAINcome NAlearn_dis NAPHYS_dis
## initial    0     0     0     3     0     0         0             0
## treated    9     2     2    26     6     5        62             4
```

Based on the results we can tell PC Principal and the Mayor that the suppression greatly reduced the level of detail about the income and the race of the students. We could continue exploring removing other less relevant variables and explore other functions in this package or even considering different ways of recoding that variable. But let's call the day for today, and export the anonymized dataset we produced.

Creating a new random number to replace the student ID

```
## Adding a new randomized ID-variable
sdcInitial <- createNewID(sdcInitial, newID="ID", withinVar="stu_id")
```

Exporting the anonymized dataset

```
writeSafeFile(obj=sdcInitial, format="csv", randomizeRecords="no", col.names=TRUE, sep
=",", dec=".", fileOut="southpark-anon.csv")
```