
 Universidad de los Andes Facultad de Ingeniería	Departamento de Ingeniería de Sistemas y Computación Curso: MINE 4204 Semestre 2018-2	 Engineering Accreditation Commission
---	---	--

Laboratorio - Control estadístico de divulgación de datos

En este laboratorio verá la forma de aplicar la protección de datos a nivel tabular, de forma que aprenda a disminuir, por medio de métodos estadísticos, el riesgo de re-identificación de ciertas observaciones que para casos prácticos pueden ser los usuarios de sus aplicaciones, sus clientes, sus empleados, etc. En el laboratorio usaremos el programa R de software estadístico y, en particular, las librerías `sdcMicro` y `sdcTable`. [Techniques]

Protección de Microdatos

1. Configuración del ambiente de trabajo

- Abra RStudio y en la ventana de la consola ejecute el siguiente comando para instalar la librería `sdcMicro` (Statistical Disclosure Control Methods for Anonymization of Microdata):

```
install.packages("sdcMicro")
```
- Cree un nuevo script de R haciendo click en el botón de la parte superior izquierda:



- En el script creado, incluya el siguiente comando para cargar la librería previamente instalada:

```
library(sdcMicro)
```
- Puede encontrar información adicional de la librería en el manual:
<https://cran.r-project.org/web/packages/sdcMicro/sdcMicro.pdf>

2. Anonimización de variables categóricas

Es importante conocer métodos que nos permitan anonimizar de forma efectiva la información asociada a atributos casi-identificadores, puesto que un intruso puede tener información de los valores asociados a algunos atributos para un individuo particular y, mientras que los atributos de identificación directa, como las identificaciones, los nombres, las direcciones y los números de seguridad social, generalmente no se difunden, las combinaciones de identificadores indirectos o casi-identificadores, como rangos de salario, categoría de edad, nivel socioeconómico, entre otros, pueden usarse para vincular fuentes de datos e identificar unidades estadísticas.

Aún más, como se ha visto en clase, si se logra re-identificar una observación, el intruso conoce todas las entradas de esta unidad estadística en el conjunto de datos y algunas entradas pueden contener información confidencial o sensible.

- Frecuencias y riesgos de divulgación**

En primer lugar, es necesario saber qué unidades estadísticas son más susceptibles de ser re-identificadas, de forma que el proceso de anonimización sea justificado y coherente.

Sea f_k con $k = 1 \dots n$ la frecuencia de aparición obtenida por tabulación cruzada y sea F_k la frecuencia de aparición con respecto a la población; entonces, si $f_k = 1$, la observación correspondiente es única en la muestra. Si $F_k = 1$, entonces la observación es única en la población.

Sin embargo, F_k se desconoce ya que generalmente se recopila información sobre muestras, no sobre la población completa. Por lo tanto, dado que en el procedimiento que seguiremos necesitamos este valor para saber qué tanto se repiten nuestras observaciones, vamos a estimarlo por medio de métodos estadísticos.

En este caso, vamos a estimar F_k utilizando el método de los pesos muestrales. Cuando una observación tiene un peso de muestreo igual a 100, se puede suponer que 100 observaciones tienen las mismas características en la población relacionada. De esta forma, el estimador \widehat{F}_k se calcula como sigue:

$$\widehat{F}_k = \sum_{i \in \{j \mid x_j = x_k\}} w_i \text{ donde } x_i \text{ es la } i\text{-ésima observación}$$

Para este laboratorio usaremos un dataset de prueba que viene integrado con la librería y, a partir de allí, vamos a estimar F_k con la función `freqCalc()`. Para ello, adicione las siguientes líneas a su script:

```
data(franccat)
x <- franccat[,c(2,4,5,6,8)]
print(x)

ff <- freqCalc(x, keyVars=c(1,2,3,4), w=5)
print(cbind(x, ff$fk, ff$Fk))
```

Las primeras 3 instrucciones importan el dataset y seleccionan las columnas 2,4,5,6 y 8. Posteriormente, la función `freqCalc` calcula las frecuencias que necesitamos (tanto f_k como F_k). Note que agregamos 3 argumentos: el primero es la matriz de datos, el segundo es un arreglo que indica en qué columnas están las variables, y el tercero es el índice de la columna que tiene los pesos asociados.

Los valores de las observaciones 1 y 8 son iguales en los atributos `key1` a `key4`. Entonces, $f_1 = 2$ y $f_8 = 2$.

La frecuencia en la población \widehat{F}_1 y \widehat{F}_8 se puede estimar con la suma de sus ponderaciones/pesos muestrales, w_1 y w_8 , que es igual a 110. Por lo tanto, hay dos observaciones con $x_k = (1, 2, 5, 1)$ en la muestra y se espera que 110 observaciones con estos valores existan en la población.

Actividad 1: Verifique manualmente que los resultados generados para `ff$fk` y para `ff$Fk` corresponden a lo esperado.

- **Medidas de riesgo global**

Una medida global del riesgo de re-identificación viene dada por la “exclusividad” que se produce tanto en la muestra como en la población. En este caso, se debe asumir una cierta distribución de F_k para formular una medida realista del riesgo global. Del conjunto de modelos existentes vamos a utilizar el modelo Bennedetti-Franconi [BF],[FP]. Hay otros modelos que pueden ser usados, dependiendo de las características de la población.

A partir de diferentes análisis se encontró entonces que una buena aproximación estadística es la siguiente:

$$f_k \mid F_k \sim \text{Poisson}(N\pi_k) \text{ donde se supone que } N \text{ es conocido}$$

Con π_k : probabilidad de inclusión (probabilidad de que un elemento de una población de tamaño N se elija en una muestra de tamaño n).

Con base en el modelo, el riesgo se calcula de la siguiente manera:

$$\hat{r}_k = \frac{\hat{p}_k}{f_k - (1 - \hat{p}_k)} \text{ donde } \hat{p}_k = \frac{f_k}{\hat{F}_k}$$

Para calcular el riesgo sobre el dataset previamente generado y con las frecuencias calculadas, use la siguiente función:

```
rk <- indivRisk(ff)$rk
print (cbind(x, ff$fk, ff$Fk, rk ))
```

La función `indivRisk()` tiene solo un parámetro que debe ser un objeto de tipo `freqCalc` como el que se calculó previamente.

La salida del ejemplo anterior muestra la frecuencia de aparición en la tabla, la frecuencia de aparición en la población y el riesgo individual de cada observación.

Actividad 2: ¿Cuál es la observación de mayor riesgo? ¿por qué cree que obtenemos este resultado?

Note que para poder utilizar esta función es necesario que se cumpla el supuesto: $f_k | F_k \sim \text{Poisson}(N\pi_k)$ y esto depende directamente de los datos usados.

- **Recodificación global y supresión local**

Por lo general los atributos con categorías (valores) que incluyen solo unas pocas entradas conducen a la singularidad y al alto riesgo. Por lo tanto, la recodificación en categorías más amplias (por ejemplo, rangos de edad en vez de edad) o la combinación de categorías (por ejemplo, la combinación de dos niveles de 2 dígitos) reduce el riesgo.

El riesgo individual estimado debe reducirse. Por ejemplo, mediante la recodificación global de los valores de un atributo. Es decir, las categorías de ciertos atributos se asignan a categorías más amplias (Generalización). De esta forma, se busca que después de la recodificación, haya más observaciones iguales para los atributos casi-identificadores, por lo que \hat{F}_k aumentaría y \hat{r}_k disminuiría.

En este caso, la recodificación de los valores del primer atributo a cuatro valores (1 a 4) podría reducir el riesgo, considerando que solo una observación tiene valor 6. Esto se realiza en R por medio de la función `globalRecode()` como se muestra a continuación:

```
x[,1] <- globalRecode(x[,1] , breaks=c(0 ,1 ,2 ,3,6) ,
labels=c(1 ,2 ,3 ,4))
print(x)

ff <- freqCalc(x, keyVars=c(1,2,3,4) , w=5)
print(cbind(x, ff$fk, ff$Fk))

rk <- indivRisk( ff )$rk
print (cbind(x, ff$fk , ff$Fk , rk ))
```

Al tomar $x[,1]$ le estamos indicando que debe tomar la primera columna. Posterior a la ejecución de la función debemos recalculamos nuestros valores para ver si se redujo el riesgo. Sin embargo, en este caso, puesto que las observaciones como una entidad conservan el mismo valor de f_k y F_k no se observa una reducción.

En estos casos (generalmente), se aplica algún mecanismo de supresión. En este método, ciertas observaciones se remueven para reducir el riesgo. Esto debe hacerse de una manera óptima, es decir, para remover la menor cantidad posible de valores por un lado y para garantizar un bajo riesgo de re-identificación por el otro.

Para ello vamos a usar k-anonymity:

```
localsupx <- kAnon(x, keyVars=1:4, k=2)
```

Los parámetros en este caso son la matriz de observaciones (x), el índice de las variables categóricas (keyVars) y el umbral correspondiente al método de k-anonymity que se implementa en la librería.

Recuerde que con k-anonymity un conjunto de datos anonimizados cumple que cada registro es similar a por lo menos k-1 registros sobre los atributos casi-identificadores (potencialmente identificables). Por ejemplo, si $k = 5$ y los casi-identificadores son edad y género, entonces un conjunto de datos anonimizados k tiene al menos 5 registros para cada combinación de edad y género. En este caso, tomamos $k=2$ por el volumen de datos que tenemos.

A continuación, genere una gráfica que indica la cantidad de valores suprimidos por cada variable categórica y posteriormente observe la matriz de observaciones resultante y compárela con la original para ver qué valores se eliminaron.

```
plot(localsupx)
print(localsupx$xAnon)
```

Ahora, para verificar que el método funcionó, calcule nuevamente las frecuencias y el riesgo asociado:

```
newX <- cbind(localsupx$xAnon, x$w);

newff <- freqCalc(newX, keyVars=c(1,2,3,4) , w=5)
print(cbind(newX,newff$fk,newff$Fk))

newrk <- indivRisk( newff )$rk
print (cbind(newX, newff$fk , newff$Fk , newrk ))
```

Actividad 3: ¿para qué observaciones se redujo el riesgo?

- **Agregación**

La implementación de la función globalRecode en la librería sdcMicro funciona para variables categóricas con valores numéricos. Para variables categóricas con valores nominales es necesario recurrir a otras funciones. La implementación de la función kAnon en la librería sdcMicro remueve automáticamente valores que no cumplen con el k seleccionado, por lo que en algunos casos puede no ser apropiada.

Por tanto, para algunos datasets será necesario revisar cuidadosamente los valores de ciertos atributos y tomar la decisión de agregarlos. En estos casos puede usar la función `groupAndRename()`; en el manual encontrará los parámetros y valor de retorno de esta función.

Actividad 4: Cargue el archivo `data4agr.txt` en un frame, revise el contenido del frame (print o use la interfaz de RStudio para revisar el contenido de la variable). A continuación, agregue los datos. Para ellos ejecute el comando que se indica a continuación y revise el resultado.

```
<miFrame> <- read.table(<miArchivo>, header=TRUE)
<miFrameagregado> <- groupAndRename(<miFrame>, var = "condicion",
  before=c("neumonia","gripa"), after = c("respiratoria") )
ffagr <- freqCalc(<miFrameagregado>, keyVars=c(2,3,4))
print(ffagr$fk)
```

Nota: Tenga en cuenta que en todos los comandos con variables entre los signos <>, de ahora en adelante, debe sustituir los nombres entre <>, por nombres significativos para sus variables.

3. Anonimización de variables numéricas

Casi todas las combinaciones de variables continuas son únicas en una muestra. Por lo tanto, el concepto de singularidad usado con variables categóricas ya no funciona para asegurar la anonimización. Sin embargo, sigue siendo indispensable asegurarnos de ello, puesto que un intruso de datos puede tener información sobre el valor de una unidad estadística y si este valor coincide con los datos anónimos, entonces él puede estar bastante seguro de que la re-identificación fue exitosa. Asimismo, un intruso también puede usar técnicas de vinculación de registros (*statistical matching*) para identificar valores ya perturbados. Por ende, si un valor no se perturba lo suficiente, entonces es posible llegar a re-identificarlo.

- **Rank swapping y microagregación**

El **Rank swapping** ordena las entradas de una variable por sus valores numéricos (ranking). Cada valor clasificado se intercambia con otro valor que se ha elegido al azar dentro de un rango restringido. El rango de dos valores intercambiadas no puede diferir en más del p por ciento del número total de observaciones. Este método debe aplicarse a cada variable por separado y, por lo tanto, la estructura de datos multivariable no se conserva muy bien. En la implementación de la librería `sdcmicro`, se implementa por columnas a través de la siguiente función:

```
rankSwap(x , variables, P)
```

donde x es la matriz de observaciones, $variables$ indica el nombre o índices de las variables sobre las que se quiere realizar el rank swapping y P es el rango como porcentaje del total de la muestra. Es decir que dos observaciones son elegibles para el intercambio (*swapping*) si sus rangos, i y j , respectivamente, satisfacen que $|i - j| \leq \frac{PN}{100}$ donde N es el tamaño total de la muestra.

Actividad 5: Cargue el archivo `data4swap.txt` en un frame, revise el contenido del frame (print o use la interfaz de RStudio para revisar el contenido de la variable). A continuación, intente ejecutar un swapping. Para ellos ejecute el comando que se indica a continuación y revise el resultado.

```
<miFrame> <- read.table(<miArchivo>, header=TRUE)
<miResultado> <- rankSwap(<miFrame>, variables=c("tarea5","tarea7"),
  TopPercent=20, BottomPercent=20, P = 15)
```

```
print(<miResultado>)
```

Por otra parte, la **microagregación** consiste en agrupar registros según una medida de proximidad de las variables de interés y, posteriormente, utilizar estos grupos pequeños para calcular agregados de esas variables. De esta forma, se publican los agregados en lugar de los registros individuales. Como es de esperarse, la elección de la medida de "proximidad" es la parte más desafiante e importante de la microagregación. Ello, puesto que la estructura multivariable de los datos solo se conserva si se agregan observaciones similares. Por lo tanto, se han desarrollado múltiples métodos de los cuales se destacan los siguientes (implementados por la librería *sd*

Micro):

- Single: ordena los datos según una sola variable en orden ascendente o descendente.
- Influence: clasifica las observaciones de cada grupo (después de hacer un proceso de clustering) por la variable más influyente del mismo.
- MDAV: distancia máxima al vector promedio (por sus siglas en inglés). Basado en las distancias Euclidianas en un espacio multivariable.
- ... (ver manual)

En cualquier caso, para realizar microagregación sobre un conjunto de datos se utiliza la siguiente función:

```
microaggregation(x, method='METHOD' , aggr=3)
```

Donde *x* es la matriz con las observaciones, *method* es el método escogido para medir la proximidad y *aggr* es el nivel de agregación que se desea obtener (por defecto se tienen 3 niveles).

Actividad 6: Construya un frame con base en los datos *francdat*, usando solo las columnas 2, 4, 5 y 6. (no usaremos los pesos) y aplique microagregación. Use los métodos *single* y *mdav*. ¿cómo varía el resultado?

```
<suFrame> <- francdat[,c(2,4,5,6)]  
print(<suFrame>)  
<suResultado> <- microaggregation(<suFrame>, method = "single", aggr = 3)  
print(<suResultado>$mx)
```

- **Método para añadir ruido**

La estrategia más usada es agregar ruido aditivo a cada variable numérica:

$$d_i = t_i + n_i$$

donde construimos una distribución normal para los t_i , con media μ y varianza σ^2 , estimadas a partir de los registros. Con la librería propuesta, es posible añadir ruido utilizando la función *addNoise(x, method)*. Los parámetros son la matriz de observaciones y, de forma opcional, el método que se desea usar (el cual depende de las características particulares, en términos estadísticos, que se desean preservar a pesar del ruido añadido).

Por otra parte, es importante notar que pueden existir conjuntos de datos no homogéneos, que incluyen valores atípicos que por su naturaleza son más fáciles de identificar (incluso añadiendo ruido). Por ello, estos valores deben estar más protegidos que el resto. En el paquete *sdMicro* se implementa un método llamado *outdetect()* que considera este hecho.

Actividad 7: Construya un frame con base en los datos francdat, usando solo las columnas 1, 3 y 7 y adicione ruido. ¿Cómo varían las observaciones?

```
<suFrame> <- francdat[,c(1,3,7)]
print(<suFrame>)
<suResultado> <- addNoise(<suFrame>,method="additive")
print(<suResultado>$xm)
```

- **Pérdida de información:**

Luego de aplicar ciertas transformaciones a los datos (como adicionar ruido o usar microagregación) es importante calcular el nivel de pérdida de información para verificar que los datos que efectivamente se van a divulgar, incluso cuando están anonimizados, siguen siendo de utilidad para los análisis de quien esté interesado.

Por ejemplo, una medida de la pérdida de información, llamada IL1, se basa en las distancias agregadas de los puntos originales a los valores correspondientes de los datos perturbados, dividida por la desviación estándar para cada variable. Por ser uno de los métodos más comunes, se muestra su definición en términos más formales:

$$IL1(X, X') = \frac{1}{VM} \sum_{i=1}^M \sum_{j=0}^V \frac{|x_{ij} - x'_{ij}|}{\sqrt{2} S_j}$$

Donde V es el número de atributos, M el número de observaciones, x_{ij} denota el valor de la observación i para el atributo j, x'_{ij} el mismo, pero para la versión perturbada y S_j la desviación estándar del j-ésimo atributo de los datos originales. Es importante notar que esta medida es grande incluso si solo un valor atípico está altamente perturbado, pero todos los demás valores son exactamente iguales a los del conjunto de datos original.

Por otra parte, se tienen medidas de pérdida de información que comparan las estadísticas univariadas de los datos originales y los datos perturbados (por ejemplo, la suma de las diferencias de la media o las medianas).

En el caso de la librería sdcMicro, se tiene la siguiente función para verificar la pérdida de información:

```
dUtility(x, xm, method='METHOD')
```

Donde x son los datos originales, xm son los datos perturbados y method es alguno de los métodos de pérdida de información soportados.

Actividad 8: vamos a calcular el nivel de pérdida de información asociado con la adición de ruido para luego compararlo con el que se obtiene al realizar microagregación.

```
dUtility(obj=<framesinRuido>, xm=<suFrameconRuido>$xm, method="IL1")
```

Note que con esta última función se miden las distancias estandarizadas de los valores de datos perturbados a los originales. En particular, como se explicó previamente, la medida IL1 mide las distancias entre los valores originales y los perturbados, escalados por la desviación estándar. Por lo tanto, entre mayor sea el número, mayor será el nivel de pérdida de información arrojado por la función.

Ejecute la función dUtility para las modificaciones con adición de ruido y microagregación. En su caso ¿cuál operación es más conveniente? (es decir, cuál genera una pérdida menor de información)

- **Riesgo de divulgación**

El riesgo de divulgación (Disclosure Risk – DR) mide el riesgo de re-identificación, que se puede dar con respecto a las unidades estadísticas completas o a información que puede ser inferida sobre los atributos de una unidad estadística particular.

Utilizando la librería propuesta, es posible calcular este riesgo con la siguiente función:

```
dRisk(x, xm)
```

Donde x son los datos originales y x_m son los datos perturbados. En este caso, la estimación se realiza por medio de intervalos que se calculan alrededor de las observaciones perturbadas con base en la desviación estándar. Luego de construir los intervalos, la función observa si los valores originales están en estos intervalos o no.

Otra función disponible es `dRiskRMD(x, xm)`. Esta función usa distancias de Mahalanobis. Este método es más robusto y considera la presencia de valores atípicos. La distancia Mahalanobis mide el número de desviaciones estándar que hay desde un punto P a la media de una distribución D (es una medida de la distancia entre P y D) y tiene en cuenta las correlaciones existentes en el conjunto de datos.

Actividad 9: Mida el riesgo para las modificaciones con adición de ruido y microagregación. En su caso, ¿cuál operación es más conveniente?

Nota:

Se recomienda guardar un script R con todos los comandos para consultas posteriores.

El comando para exportar un dataset a un archivo .txt es:

```
write.table(mydata, "c:/mydata.txt", sep="\t")
```

4. Tarea (para entregar en sícua+)

Se le entregará un dataset con información sobre el desempeño obtenido en el área de matemáticas por estudiantes de dos escuelas secundarias de Portugal. Los atributos de los datos incluyen calificaciones de los estudiantes, características demográficas, sociales y relacionadas con la escuela, y se recopilaron mediante el uso de informes y cuestionarios escolares. [UC-Irvine]

Los atributos están etiquetados en inglés y se indican a continuación:

Attributes for both student-mat.csv (Math course) and student-por.csv (Portuguese language course) datasets:

1 school - student's school (binary: 'GP' - Gabriel Pereira or 'MS' - Mousinho da Silveira)

2 sex - student's sex (binary: 'F' - female or 'M' - male)

3 age - student's age (numeric: from 15 to 22)

4 address - student's home address type (binary: 'U' - urban or 'R' - rural)

5 famsize - family size (binary: 'LE3' - less or equal to 3 or 'GT3' - greater than 3)

6 Pstatus - parent's cohabitation status (binary: 'T' - living together or 'A' - apart)

7 Medu - mother's education (numeric: 0 - none, 1 - primary education (4th grade), 2 - 5th to 9th grade, 3 - secondary education or 4 - higher education)

8 Fedu - father's education (numeric: 0 - none, 1 - primary education (4th grade), 2 - 5th to 9th grade, 3 - secondary education or 4 - higher education)

9 Mjob - mother's job (nominal: 'teacher', 'health' care related, civil 'services' (e.g. administrative or police), 'at_home' or 'other')

10 Fjob - father's job (nominal: 'teacher', 'health' care related, civil 'services' (e.g. administrative or police), 'at_home' or 'other')

11 reason - reason to choose this school (nominal: close to 'home', school 'reputation', 'course' preference or 'other')

12 guardian - student's guardian (nominal: 'mother', 'father' or 'other')

13 traveltime - home to school travel time (numeric: 1 - <15 min., 2 - 15 to 30 min., 3 - 30 min. to 1 hour, or 4 - >1 hour)

14 studytime - weekly study time (numeric: 1 - <2 hours, 2 - 2 to 5 hours, 3 - 5 to 10 hours, or 4 - >10 hours)

15 failures - number of past class failures (numeric: n if 1<=n<3, else 4)

16 schoolsup - extra educational support (binary: yes or no)

17 famsup - family educational support (binary: yes or no)

18 paid - extra paid classes within the course subject (binary: yes or no)

19 activities - extra-curricular activities (binary: yes or no)

20 nursery - attended nursery school (binary: yes or no)

21 higher - wants to take higher education (binary: yes or no)

22 internet - Internet access at home (binary: yes or no)

23 romantic - with a romantic relationship (binary: yes or no)

24 famrel - quality of family relationships (numeric: from 1 - very bad to 5 - excellent)

25 freetime - free time after school (numeric: from 1 - very low to 5 - very high)

26 goout - going out with friends (numeric: from 1 - very low to 5 - very high)

27 Dalc - workday alcohol consumption (numeric: from 1 - very low to 5 - very high)

28 Walc - weekend alcohol consumption (numeric: from 1 - very low to 5 - very high)

29 health - current health status (numeric: from 1 - very bad to 5 - very good)

30 absences - number of school absences (numeric: from 0 to 93)

these grades are related with the course subject:

31 G1 - first period grade (numeric: from 0 to 20)

32 G2 - second period grade (numeric: from 0 to 20)

33 G3 - final grade (numeric: from 0 to 20, output target)

34 w - pesos de muestreo (sampling weights)

Note que se trata de información sensible (tanto por los atributos relacionados como por el hecho de que se trata de menores de edad en su mayoría) y, por este motivo, su trabajo es anonimizar este dataset para reducir el riesgo de re-identificación de las unidades estadísticas (en este caso estudiantes).

Adelante el siguiente procedimiento:

1. Estudiar los atributos que se proporcionan. Responda (justifique brevemente todas sus respuestas):
 - a. ¿Hay relaciones entre algunos de ellos?
 - b. ¿Es necesario suprimir algún atributo?
 - c. ¿Cuáles son deducibles a partir de otros?
 - d. ¿Cuáles son categóricos y cuáles numéricos?
2. Para los atributos binarios y nominales que así lo requieran, cambie las etiquetas de texto por identificadores numéricos (e.g. en los binarios, cambie “yes” por 1 y “no” por 0).
 - a. Indique los cambios realizados.
3. Cargue el archivo resultante a RStudio. Para ello, puede utilizar el siguiente comando:


```
<suFrame> =
```

```
read.table("C:/la_ruta_a_su_archivo.csv", sep=";", header=TRUE)
```

4. Puesto que ya identificó qué variables son categóricas, puede proceder con la anonimización de dichas variables. Empiece identificando las frecuencias y riesgos de divulgación. Una vez tenga conocimiento de la existencia de combinaciones únicas de variables, utilice los métodos de recodificación global y supresión local para reducir el riesgo asociado. Puede graficar este último y valerse de los métodos previamente presentados.

- a. ¿Hay observaciones únicas? ¿Cuántas?
- b. ¿El riesgo de re-identificación es muy alto?
- c. Copie los comandos ejecutados para cumplir con este punto.

Nota: Puede suponer que se cumple el supuesto de distribución Poisson de f_k y F_k , porque los datos fueron especialmente preparados para ello.

5. Es necesario continuar el proceso con las variables numéricas. Considere las posibilidades de añadir ruido, o efectuar rank swapping sobre alguno(s) de ellos. Posteriormente, utilice el procedimiento de microagregación para perturbar el conjunto de datos. Por último, no olvide evaluar estas transformaciones verificando la pérdida de información y el riesgo de divulgación. Note que estos dos últimos representan un trade-off a nivel del proceso de anonimización, porque el riesgo se disminuye sustancialmente a medida que aumenta el nivel de pérdida de información, pero, como es natural, no es deseable que este nivel crezca exponencialmente; razón por la cual es necesario encontrar un equilibrio entre ambos.

- a. Copie los comandos ejecutados para cumplir con este punto.

6. Una vez haya verificado que el riesgo ha sido efectivamente reducido en ambos casos (variables categóricas y numéricas), unifique todo en un único dataset anonimizado que usted divulgaría sin temor de re-identificación de los estudiantes portugueses del estudio.

- a. Copie los comandos ejecutados para cumplir con este punto.

Entrega:

Cada grupo debe entregar en sicua+, en el enlace para tal fin: (1) un documento con las respuestas a las preguntas planteadas; incluyendo los comandos R ejecutados para cumplir con cada punto (en formato texto), (2) un archivo texto con el dataset anonimizado.

Referencias

- [BF] Statistical and technological solutions for controlled data dissemination. R Benedetti, L Franconi. En Pre-proceedings of New Techniques and Technologies for Statistics. 1998. Pgs. 225-232
- [FP] Individual Risk Estimation in μ -Argus: A Review. Franconi, Luisa and Poletti, Silvia. In Privacy in Statistical Databases, 2004. Publisher=Springer Berlin Heidelberg. Pgs. 262—272.
- [UC-Irvine] University of California, Irvine. (2014). *Student Performance Data Set* [Data file]. Retrieved from <http://archive.ics.uci.edu/ml/datasets/Student+Performance#>
- [Techniques] Practical Applications in Statistical Disclosure Control Using R. Mathias Templ y Bernhard Meindl. En Privacy and Anonymity in Information Management Systems: New Techniques for New Practical Problems. J. Nin y J. Herranz. 2010. Pgs. 31 – 60.