# Methods for Risk Evaluation part 1

Jingchen (Monika) Hu

Vassar College

Statistical Data Privacy

# Outline

# Outline

# Recap

- Lecture 6:
  - Global utility
  - Two measures: *pMSE* (propensity score) and ECDF
- Lecture 7:
  - Analysis-specific utility
  - Combining rules and interval overlap

# Plan for this lecture

- Two general types of disclosure: identification and attribute (Lecture 1)

1. Identification disclosure: The intruder correctly identifies records of interest in the released synthetic data
2. Attribute disclosure: The intruder correctly infers the true confidential values of the synthetic records using information from the released synthetic data

# Plan for this lecture

- Two general types of disclosure: identification and attribute (Lecture 1)

1. Identification disclosure: The intruder correctly identifies records of interest in the released synthetic data
2. Attribute disclosure: The intruder correctly infers the true confidential values of the synthetic records using information from the released synthetic data

- In this lecture, we focus on identification risk evaluation methods, with illustrations to the synthetic CE from Lectures 4 & 5 and the synthetic ACS from Lecture 5

# Overview

- Identification disclosure:
  - The intruder correctly identifies records of interest in the released synthetic data
  - Only exist in partially synthetic data
- We will introduce two general approaches
  - Matching-based approaches
  - Record-linkage approaches (next lecture)

Discussion question: Suppose in the CE sample (five variables: `UrbanRural`, `Income`, `Race`, `Expenditure`, `KidsCount`), the synthetic data has `Expenditure` synthesized. Now suppose you know someone who's in the CE survey, and you know their `UrbanRural` and `Race`, how would you go about finding that person in the synthetic data?

# Outline

# Overview

- In the matching-based approaches, we

1. Make assumptions about the knowledge possessed by the intruder for a confidential record $i$
2. Use the assumed knowledge to investigate which records in the synthetic data are matched with record $i$
3. Evaluate whether the true match is among the matched records, how unique the true match is, among other things

# Overview

- In the matching-based approaches, we

1. Make assumptions about the knowledge possessed by the intruder for a confidential record $i$
2. Use the assumed knowledge to investigate which records in the synthetic data are matched with record $i$
3. Evaluate whether the true match is among the matched records, how unique the true match is, among other things

- We present a basic version of Reiter and Mitra (2009), which is a form of **Bayesian probabilistic matching**

# Notation and setup

In the sample $S$ of $n$ units and $r$ variables:

- $y_{ij}$ refers to the $j$-th variable of the $i$-th unit, where $i = 1, \cdots, n$ and $j = 0, 1, \cdots, r$
- The column $j = 0$ contains some unique identifiers (such as name or Social Security Number), which are never released
- Among the $r$ variables:
  - some are available to users from external databases, denoted by $\mathbf{y}_i^A$
  - others are unavailable to users except in the released data, denoted by $\mathbf{y}_i^U$
- We therefore have the vector response of the $i$-th unit, $\mathbf{y}_i = (y_{i1}, \cdots, y_{ir}) = (\mathbf{y}_i^A, \mathbf{y}_i^U)$

We also have the matrix $\mathbf{Y} = (\mathbf{Y}^A, \mathbf{Y}^U)$ representing the confidential values of all $n$ units

# Notation and setup

- On the confidential data holder side, suppose it releases all $n$ units of the sample $S$
- Similar to the split of $\mathbf{y}^i$, we have $\mathbf{z}_i = (z_{i1}, \cdots, z_{ir}) = (\mathbf{z}_i^A, \mathbf{z}_i^U)$
- Among the available variables, we further split them into:
  - $\mathbf{z}_i^{A_s}$ the synthesized variables
  - $\mathbf{z}_i^{A_{us}}$ the unsynthesized variables
- We therefore have $\mathbf{z}_i = (\mathbf{z}_i^{A_{us}}, \mathbf{z}_i^{A_s}, \mathbf{z}_i^U)$, and we let $\mathbf{Z} = (\mathbf{Z}^{A_{us}}, \mathbf{Z}^{A_s}, \mathbf{Z}^U)$ be the matrix of all released data
- We also let $\mathbf{Y}^{A_s}$ be all $n$ units' confidential values of the synthesized variables (in some cases, the agency might only release $n' \leq n$ units of the sample (Reiter (2005)))

# Notation and setup

- On the intruder side, let $\mathbf{t}$ to be the vector of information available to the intruder.
- $\mathbf{t}$ may or may not be in $\mathbf{Z}$, but we assume $\mathbf{t} = \mathbf{y}_i^A$ for some unit in the population
- This vector $\mathbf{t}$ only contains unsynthesized and synthesized variables (no unavailable variables as in $\mathbf{y}_i$ and $\mathbf{z}_i$), thus we have $\mathbf{t} = (\mathbf{t}^{A_{us}}, \mathbf{t}^{A_s})$
- When $z_{i0} = t_0$, the intruder will declare a match of $\mathbf{t}$ with unit $i$ in $\mathbf{Z}$; when $z_{i0} \neq t_0$, the intruder will declare no match of $\mathbf{t}$ with any unit in $\mathbf{Z}$

# Notation and setup

- Additionally, two other pieces of information can be available to the intruder
- Let $S$ represent the meta-data released about the simulation models used to generate the synthetic data
- Let $R$ represent the meta-data released about the reason why records were selected for synthesis
- Either $S$ or $R$ could be empty

# Notation and setup

- There are $n$ released units in $\mathbf{Z}$
- Let $I$ be the random variable that equals to $i$ when $z_{i0} = t_0$ for $i \in \mathbf{Z}$ and equals $n+1$ when $z_{i0} = t_0$ for some $i \notin \mathbf{Z}$
- The intruder intends to calculate $Pr(I = i \mid \mathbf{t}, \mathbf{Z}, S, R)$ for $i = 1, \cdots, n+1$
- The intruder is particularly interested in learning whether any of the calculated identification probabilities for $i = 1, \cdots, n$ are large enough to declare an identification

# Notation and setup

- The intruder intends to calculate $Pr(I = i \mid \mathbf{t}, \mathbf{Z}, S, R)$ for $i = 1, \cdots, n+1$.
- Based on the split of $\mathbf{Z} = (\mathbf{Z}^{A_{us}}, \mathbf{Z}^{A_s}, \mathbf{Z}^{U})$, we re-write the probability as

$$Pr(I = i \mid \mathbf{t}, \mathbf{Z}^{A_{us}}, \mathbf{Z}^{A_s}, \mathbf{Z}^{U}, S, R). \tag{1}$$

# Notation and setup

- The intruder intends to calculate $Pr(I = i \mid \mathbf{t}, \mathbf{Z}, S, R)$ for $i = 1, \cdots, n+1$.
- Based on the split of $\mathbf{Z} = (\mathbf{Z}^{A_{us}}, \mathbf{Z}^{A_s}, \mathbf{Z}^{U})$, we re-write the probability as

$$Pr(I = i \mid \mathbf{t}, \mathbf{Z}^{A_{us}}, \mathbf{Z}^{A_s}, \mathbf{Z}^{U}, S, R). \tag{1}$$

- In the current basic version, we consider $S$ and $R$ to be both empty

# Notation and setup

- This suggests that the intruder will perform matching records in the synthetic data with a confidential, target record $i$, based on
  - a set of available (known to the intruder), unsynthesized variables
  - the true, confidential values of the synthesized variables

Discussion question: For example, in the CE sample where `Expenditure` is synthesized, if the intruder knows the unsynthesized `UrbanRural` and `Race` and the true confidential value of the synthesized `Expenditure` information about a target record, how will they go about finding matched records in the synthetic data?

# Three risk summaries

- Regardless of the type(s) of the synthetic variable(s), there are three widely-used summaries of identification disclosure risks based on matching: **the expected match risk**, **the true match rate**, and **the false match rate**

# The expected match risk

- The expected match risk measures on average how likely it is to find the correct match for each record, and for the sample as a whole. It is defined as:

$$\sum_{i=1}^{n} \frac{T_i}{c_i}, \tag{2}$$

where $T_i = 1$ if the true match is among the $c_i$ units and $T_i = 0$ otherwise, and $c_i$ is the number of records with the highest match probability for the target record $i$

Discussion question: When $T_i = 1$ and $c_i > 1$, what does $T_i/c_i$ represent? What happens to $\sum_{i=1}^{n} \frac{T_i}{c_i}$ when for record $i$, $T_i = 0$?

# The expected match risk

- Each $T_i/c_i$ is a record-level probability $\in [0, 1]$
- The sum $\sum_{i=1}^{n} T_i/c_i$ is a sample-level summary of the expected match risk, which is $\in [0, n]$
- The higher the expected match risk $\sum_{i=1}^{n} T_i/c_i$, the higher the identification disclosure risk for the sample, and vise versa

# The true match rate

- The true match rate considers how large is a percentange of true unique matches that exist. It is defined as:

$$\sum_{i=1}^{n} \frac{K_i}{N}, \tag{3}$$

where $K_i = 1$ if the true match is the unique match (i.e., $c_i T_i = 1$) and $K_i = 0$ otherwise, and $N$ is the total number of target records out of $n$ total records

# The true match rate

- The true match rate considers how large is a percentange of true unique matches that exist. It is defined as:

$$\sum_{i=1}^{n} \frac{K_i}{N}, \tag{3}$$

where $K_i = 1$ if the true match is the unique match (i.e., $c_i T_i = 1$) and $K_i = 0$ otherwise, and $N$ is the total number of target records out of $n$ total records

- $\sum_{i=1}^{n} K_i/N$ is the percentage of true unique matches among the target records

Discussion question: The higher the true match rate, the lower or the higher the identification risk for the sample?

# The false match rate

- The false match rate considers how large is a percentage of unique matches that are actually false matches. It is defined as:

$$\sum_{i=1}^{n} \frac{F_i}{s}. \tag{4}$$

where $F_i = 1$ if there is a unique match but it is not the true match (i.e., $c_i(1 - T_i) = 1$) and $F_i = 0$ otherwise, and $s$ is the number of uniquely matched records (i.e., $\sum_{i=1}^{n}(c_i = 1)$)

# The false match rate

- The false match rate considers how large is a percentage of unique matches that are actually false matches. It is defined as:

$$\sum_{i=1}^{n} \frac{F_i}{s}. \tag{4}$$

  where $F_i = 1$ if there is a unique match but it is not the true match (i.e., $c_i(1 - T_i) = 1$) and $F_i = 0$ otherwise, and $s$ is the number of uniquely matched records (i.e., $\sum_{i=1}^{n}(c_i = 1)$)

- $\sum_{i=1}^{n} F_i/s$ is the percentage of false matches among unique matches
- The lower the false match rate, the higher the identification disclosure risk for the sample, and vise versa

# Summary and dicussion

- Higher expected match risk, higher true match rate, and lower false match rate indicate higher identification disclosure risk for the sample
- When $m > 1$ synthetic datasets are generated, we can calculate the three summaries on each synthetic dataset, and for each summary compute its averages across $m$ samples
- Each measure is providing one aspect of the identification risk; we should consider them as a whole when comparing different synthetic datasets

# Example of the ACS sample

We use the matching-based approach to evaluate the identification disclosure risks of the synthetic ACS sample from Lecture 5, where `DIS`, `HICOV` are synthesized and the other variables remain unsynthesized. The synthesis model is the DPMPM model with the NPBayesImputeCat R package. We assume that the intruder knows `SEX`, `RACE`, `MAR` of reach record.

# Example of the ACS sample

- Loading datasets

```
ACSdata <- data.frame(readr::read_csv(file = "ACSdata.csv"))
n <- dim(ACSdata)[1]

ACSdata_syn <- data.frame(readr::read_csv(file = "ACSdata_syn.csv"))
## make sure variables are in the same ordering
ACSdata_syn <- ACSdata_syn[, names(ACSdata)]
ACSdata_con <- ACSdata
```

- We assume that the intruder knows each record's `SEX`, `RACE`, `MAR`, and trying to use this information to identify records in the synthetic ACS data in `ACSdata_syn`

# Example of the ACS sample: calculate key quantities

```
CalculateKeyQuantities_cat <- function(condata, syndata,
                                       known.vars, syn.vars, n){
  condata <- condata
  syndata <- syndata
  n <- n

  c_vector <- rep(NA, n)
  T_vector <- rep(NA, n)

  for (i in 1:n){
    match <- (eval(parse(text=paste("condata$",syn.vars,
                                    "[i]==syndata$",syn.vars,
                                    sep="",collapse="&")))&
                eval(parse(text=paste("condata$",known.vars,
                                      "[i]==syndata$",
                                      known.vars,sep="",
                                      collapse="&"))))
    match.prob <- ifelse(match, 1/sum(match), 0)
```

# Example of the ACS sample: calculate key quantities

```
    if (max(match.prob) > 0){
      c_vector[i] <- length(match.prob[match.prob == max(match.prob)])
    }
    else
      c_vector[i] <- 0
      T_vector[i] <- is.element(i, rownames(condata)
                                [match.prob == max(match.prob)])
  }

  K_vector <- (c_vector * T_vector == 1)
  F_vector <- (c_vector * (1 - T_vector) == 1)
  s <- length(c_vector[c_vector == 1 & is.na(c_vector) == FALSE])

  res_r <- list(c_vector = c_vector,
                T_vector = T_vector,
                K_vector = K_vector,
                F_vector = F_vector,
                s = s
  )
  return(res_r)
}
```

# Example of the ACS sample: calculate key quantities

- On the synthetic ACS sample, we obtain the key quantities as below
- Recall that there are two synthesized variables: `DIS`, `HICOV`, which are assigned to `syn.vars`
- The known variables are assumed to be: `SEX`, `RACE`, `MAR`, which are assigned to `known.vars`

```
KeyQuantities_ACS <- CalculateKeyQuantities_cat(condata = ACSdata_con,
                                                syndata = ACSdata_syn,
                                                known.vars = c("SEX",
                                                               "RACE",
                                                               "MAR"),
                                                syn.vars = c("DIS",
                                                             "HICOV"),
                                                n = dim(ACSdata)[1])
```

# Example of the ACS sample: calculate three risk summaries

- We create the function `IdentificationRiskCal()` which takes the previously calculated key quantities, and `N`, the number of target inputs as the inputs

```
IdentificationRiskCal <- function(c_vector, T_vector,
                                  K_vector, F_vector,
                                  s, N){

  nonzero_c_index <- which(c_vector > 0)

  exp_match_risk <- sum(1/c_vector[nonzero_c_index] *
                          T_vector[nonzero_c_index])
  true_match_rate <- sum(na.omit(K_vector))/N
  false_match_rate <- sum(na.omit(F_vector))/s

  res_r <- list(exp_match_risk = exp_match_risk,
                true_match_rate = true_match_rate,
                false_match_rate = false_match_rate
  )
  return(res_r)
```

# Example of the ACS sample: calculate three risk summaries

- On the synthetic ACS sample, we extract the key quantities from `KeyQuantities` and pass them into the `IdentificationRiskCal()` function to calculate the three summary measures
- Note that we assume each record is a target, therefore `N = n`

```
ACS_res <- IdentificationRiskCal(c_vector = KeyQuantities_ACS[["c_vector"]],
                                 T_vector = KeyQuantities_ACS[["T_vector"]],
                                 K_vector = KeyQuantities_ACS[["K_vector"]],
                                 F_vector = KeyQuantities_ACS[["F_vector"]],
                                 s = KeyQuantities_ACS[["s"]],
                                 N = n)
```

# Example of the ACS sample: calculate three risk summaries

```
ACS_res[["exp_match_risk"]]
```

```
## [1] 64.78361
```

```
ACS_res[["true_match_rate"]]
```

```
## [1] 7e-04
```

```
ACS_res[["false_match_rate"]]
```

```
## [1] 0.72
```

```
KeyQuantities_ACS[["s"]]
```

```
## [1] 25
```

Discussion question: What are the results telling us?

# Example of the ACS sample: results on the confidential data

- Practice using the functions on the confidential data
- The results are
  - Expected match risk: 173
  - True match rate: 0.003
  - False match rate: 0
  - Unique matches: 30

Discussion question: How do the results compare?

# Using the IdentificationRiskCalculation R package

- Check out the IdentificationRiskCalculation R package (Hornby and Hu (2020))
- Hornby and Hu (2021) provides examples
- Your results should match with the output from the package

# When $m > 1$

- When dealing with `m > 1`, we calculate the three summaries in each synthetic dataset, and then take the average to obtain three final summaries
- The R script can be updated by specifying the `c_vector, T_vector, K_vector, F_vector` as matrices, and `exp_match_risk, true_match_rate, false_match_rate` as vectors
- See the hidden code for example code

# Example of the CE sample

We use the matching-based approaches to evaluate the identification disclosure risks of a synthetic CE sample in Lecture 4, where `Expenditure` is synthesized using a Bayesian simple linear regression synthesizer with `Income` as a predictor. We assume the intruder knows `UrbanRural, Race` of each record.

# Example of the CE sample

- Loading datasets

```
CEdata <- readr::read_csv(file = "CEdata.csv")
n <- dim(CEdata)[1]

CEdata_syn_SLR <- data.frame(readr::read_csv(file = "CEdata_syn_SLR.csv"))

CEdata_con <- CEdata
CEdata_syn <- CEdata_syn_SLR
```

- We proceed with `CEdata_con` and `CEdata_syn`
- We assume that the intruder knows each records' `UrbanRural`, `Race`, and trying to use this information to identify records in the synthetic CE data in `CEdata_syn`

# Example of the CE sample: calculate key quantities

- In the categorical case

```
match <- (eval(parse(text=paste("condata$",syn.vars,
                                "[i]==syndata$",syn.vars,
                                sep="",collapse="&")))&
          eval(parse(text=paste("condata$",known.vars,
                                "[i]==syndata$",
                                known.vars,sep="",
                                collapse="&"))))
```

- When dealing with continuous variables, checking exact match might not be meaningful
- We can consider using a radius to declare a match

# Example of the CE sample: calculate key quantities

- Main change from the categorical case to the continuous case below

```
radius <- r*eval(parse(text=paste("condata$",syn.vars,"[i]")))
match <- (eval(parse(text=paste("syndata$",syn.vars,
                                "<=condata$",syn.vars,
                                "[i]+",radius,sep="",
                                collapse="&")))&
          eval(parse(text=paste("syndata$",syn.vars,
                                ">=condata$",syn.vars,
                                "[i]-",radius,sep="",
                                collapse="&")))&
          eval(parse(text=paste("condata$",known.vars,
                                  "[i]==syndata$",
                                  known.vars,sep="",
                                  collapse="&"))))
```

- See hidden code for the entire script for `CalculateKeyQuantities_cont`

# Example of the CE sample: calculate key quantities

```
KeyQuantities_CE <- CalculateKeyQuantities_cont(condata = CEdata_con,
                                                syndata = CEdata_syn,
                                                known.vars = c("UrbanRural"
                                                               "Race"),
                                                syn.vars = c("Expenditure")
                                                n = dim(CEdata)[1],
                                                r = 0.2)
```

# Example of the CE sample: calculate three risk summaries

- On the synthetic CE sample, we extract the key quantities from `KeyQuantities` and pass them into the `IdentificationRiskCal()` function to calculate the three summary measures
- Note that we assume each record is a target, therefore `N = n`

```
CE_res <- IdentificationRiskCal(c_vector = KeyQuantities_CE[["c_vector"]],
                                T_vector = KeyQuantities_CE[["T_vector"]],
                                K_vector = KeyQuantities_CE[["K_vector"]],
                                F_vector = KeyQuantities_CE[["F_vector"]],
                                s = KeyQuantities_CE[["s"]],
                                N = n)
```

# Example of the CE sample: calculate three risk summaries

```
CE_res[["exp_match_risk"]]
```

```
## [1] 10.5975
```

```
CE_res[["true_match_rate"]]
```

```
## [1] 0.0003896357
```

```
CE_res[["false_match_rate"]]
```

```
## [1] 0.9230769
```

```
KeyQuantities_CE[["s"]]
```

```
## [1] 26
```

Discussion question: What are the results telling us?

# Example of the CE sample: results on the confidential data

- Practice using the functions on the confidential data
- The results are
  - Expected match risk: 101.41
  - True match rate: 0.0045
  - False match rate: 0
  - Unique matches: 23

Discussion question: How do the results compare?

# Using the IdentificationRiskCalculation R package

- Check out the IdentificationRiskCalculation R package (Hornby and Hu (2020))
- Hornby and Hu (2021) provides examples
- Your results should match with the output from the package

# Outline

# Summary

- Matching-based approaches for identification disclosure risk evaluations
  - Three risk summaries
  - Comparison between the synthetic data risk and confidential data risk

# Summary

- Matching-based approaches for identification disclosure risk evaluations
  - Three risk summaries
  - Comparison between the synthetic data risk and confidential data risk

- No homework! But next week you will present global utility and analysis-specific utility results of your your simulated synthetic data for your project
- Lecture 9: Methods for risk evaluation part 2
  - R package `reclin` (record-linkage)
  - Baillargeon and Charest (2020) (CAP statistic)

# References I

Baillargeon, M., and A. Charest. 2020. "A Closer Look at the CAP Risk Measure for Synthetic Datasets." Privacy in Statistical Databases (E-Proceedings).

Hornby, R., and J. Hu. 2020. IdentificationRiskCalculation: Calculating the Identification Risk in Partially Synthetic Microdata. https://github.com/RyanHornby/IdentificationRiskCalculation.

———. 2021. "Identification Risks Evaluation of Partially Synthetic Data with the Identificationriskcalculation R Package." Transactions of Data Privacy 14 (1): 37–52.

Reiter, J. P. 2005. "Estimating Risks of Identification Disclosure in Microdata." Journal of the American Statistical Association 100: 1103–12.

Reiter, J. P., and R. Mitra. 2009. "Estimating Risks of Identification Disclosure in Partially Synthetic Data." The Journal of Privacy and Confidentiality 1 (1): 99–110.