September 11, 2024

PSEUDONYMISATION AND K-ANONYMITY

PETs
Pseudonymisation

Raw Data

Name	UPI	Age	Gender	Disease
Jhon	dgre6789	29	M	Cancer
Kate	spea6271	22	F	Infection
Alice	jgib9592	35	F	AIDS
Ellen	hbar4405	38	F	AIDS
Mary	mhan6405	41	F	Cancer

Pseudonymised Data

ID	Age	Gender	Disease
P001	29	M	Cancer
P002	22	F	Infection
P003	35	F	AIDS
P004	38	F	AIDS
P005	41	F	Cancer

Pseudonym

ID	Name	UPI
P001	Jhon	dgre6789
P002	Kate	spea6271
P003	Alice	jgib9592
P004	Ellen	hbar4405
P005	Mary	mhan6405

Mapping Table

PETs

Pseudonymisation - Steps

1. Preprocess data
2. Identify direct identifiers
3. Replace direct identifiers with a pseudonym column
4. Create a mapping table if needed
5. Secure the mapping table

PETs

Pseudonymisation – Step 2 Generate a Pseudonym Column

- Counter
- Random Number Generator (RNG)
- Cryptographic Hashing
- Message Authentication Code (MAC)
- Encryption

Which one does not need a mapping table?

Let’s Code

PETs
K-Anonymity

ID	Age	Zip	Gender	Disease
P001	22	2141	M	Cancer
P002	24	2141	F	Infection
P003	31	2138	F	AIDS
P004	32	2139	F	AIDS
P005	41	2243	M	Cancer
P006	41	2245	M	Infection
P007	48	6534	M	Infection

$\underbrace{\text{Age, Zip, Gender}}_{\text{Quasi Identifiers}}$

Sensitive Data

2-Anonymous Data (k=2)

ID	Age	Zip	Gender	Disease
P001	21 – 30	2141	Human	Cancer
P002	21 – 30	2141	Human	Infection
P003	31 – 35	213*	F	AIDS
P004	31 – 35	213*	F	AIDS
P005	41 - 50	*	M	Cancer
P006	41 - 50	*	M	Infection
P007	41 - 50	*	M	Infection

Suppression

Generalisation

k = 2 => **at least** 2 rows in each group

QIDs {Age, Zip, Gender} in each group is the same

PETs

K-Anonymity - Steps

1. Preprocess data
2. Identify Quasi Identifiers (QIDs)
3. Partition the dataset : Each group at least K rows
4. Anonymise values
   - For each QID
     - For each partition

ID	Age
P001	22
P002	24
P003	31
P004	32

Partition 1: P001, P002

Partition 2: P003, P004

QID

Step 3 →

ID	Age
P001	20 - 25
P002	20 - 25
P003	30 - 35
P004	30 - 35

PETs

K-Anonymity - Step 3 Partitioning

How would you partition this for 3-Anonymity?

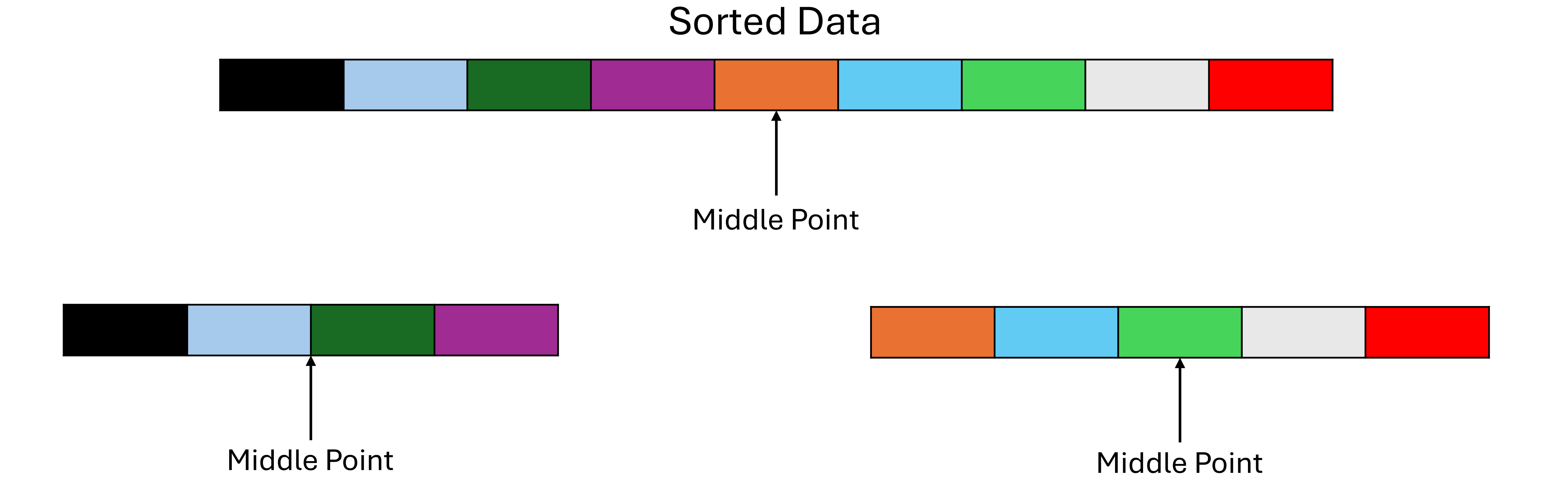
ID	Age
ID01	24
ID02	23
ID03	21
ID04	20
ID05	22
ID06	24
ID07	20
ID08	21
ID09	22
ID10	23

- Sort ‘Age’ -> loop -> break into size 3 groups?
- Sort ‘Age’ -> loop -> break into groups >= 3?
- What happens when you have multiple QIDs?

We need a fairer way to partition

PETs

Mondrian Algorithm



Stop when a condition is met

PETs

K-Anonymity - Step 3 Partitioning (**Mondrian Algorithm)**

A. Select a partition **P** (at the start : all the rows)

If size(**P**) >= 2 * **K**

B. Select the most diverse QID in the partition (why?)

Find the median of the QID values (**M**)

Split the **P** into 2 sub partitions

LHS < **M** and RHS >= **M**

If size(LHS) < **K** OR size(RHS) < **K**

If more QIDs exist

Step B for **P**: remaining QIDs

else

Add **P** to final_partitions

else

Step A for **P** = LHS and **P** = RHS

else

Add **P** to final_partitions

Row Index	ID	Age	Gender
0	ID01	24	M
1	ID02	23	M
2	ID03	21	M
3	ID04	20	F
4	ID05	22	M
5	ID06	24	F
6	ID07	20	M
7	ID08	21	F
8	ID09	22	F
9	ID10	23	F

QIDs

PETs

K-Anonymity - **Mondrian Algorithm**

Median

1. Numerical Data (e.g., Age)
   - Sort
   - Find the middle point
2. Categorical Data (e.g., Gender)
   - Find the unique values ['M', 'F']
   - Divide the unique values into 2 groups

     G1 =['M'] G2 =['F']
   - Match the rows into the correct group

     G1 =['M'] [0,1,2,4,6]

     G2 =['F'] [3,5,7,8,9]

Row Index	ID	Age	Gender
0	ID01	24	M
1	ID02	23	M
2	ID03	21	M
3	ID04	20	F
4	ID05	22	M
5	ID06	24	F
6	ID07	20	M
7	ID08	21	F
8	ID09	22	F
9	ID10	23	F

QIDs

PETs

Partitioning - Visualise

Assume K = 3

Row Index	ID	Age	Gender
0	ID01	24	M
1	ID02	23	M
2	ID03	21	M
3	ID04	20	F
4	ID05	22	M
5	ID06	24	F
6	ID07	20	M
7	ID08	21	F
8	ID09	22	F
9	ID10	23	F

A. Partition (**P**) :

0	1	2	3	4	5	6	7	8	9

If Size (**P**) >= 2 * 3 **true**

B. Select the most diverse QID in the partition

Age – 5/10 unique values (more diverse)
Gender – 2/10 unique values

qid_variability = [‘Age’, ‘Gender’]

Select ‘Age’

Continue....

PETs

Partitioning - Visualise

Assume K = 3

Row Index	ID	Age	Gender
0	ID01	24	M
1	ID02	23	M
2	ID03	21	M
3	ID04	20	F
4	ID05	22	M
5	ID06	24	F
6	ID07	20	M
7	ID08	21	F
8	ID09	22	F
9	ID10	23	F

Find the median (M)

row index	0	1	2	3	4	5	6	7	8	9
Age	24	23	21	20	22	24	20	21	22	23

	3	6	2	7	4	8	1	9	0	5
sorted	20	20	21	21	22	22	23	23	24	24

M = 22

Split the **P** into 2 sub partitions

LHS < 22

3	6	2	7

RHS >= 22

4	8	1	9	0	5

Continue....

PETs

Partitioning - Visualise

Assume K = 3

Row Index	ID	Age	Gender
0	ID01	24	M
1	ID02	23	M
2	ID03	21	M
3	ID04	20	F
4	ID05	22	M
5	ID06	24	F
6	ID07	20	M
7	ID08	21	F
8	ID09	22	F
9	ID10	23	F

LHS

3	6	2	7

RHS

4	8	1	9	0	5

If size(LHS) <3 OR size(RHS) <3 = **false**

Step A for **P** = LHS and **P** = RHS

Continue....

PETs

Partitioning - Visualise

Assume K = 3

Row Index	ID	Age	Gender
0	ID01	24	M
1	ID02	23	M
2	ID03	21	M
3	ID04	20	F
4	ID05	22	M
5	ID06	24	F
6	ID07	20	M
7	ID08	21	F
8	ID09	22	F
9	ID10	23	F

Step A for P = LHS

1

LHS

3	6	2	7

If size(LHS) >= 2 * 3
4 >= 6
false

Add **P** to final_partitions

3	6	2	7

PETs

Partitioning - Visualise

Assume K = 3

Row Index	ID	Age	Gender
0	ID01	24	M
1	ID02	23	M
2	ID03	21	M
3	ID04	20	F
4	ID05	22	M
5	ID06	24	F
6	ID07	20	M
7	ID08	21	F
8	ID09	22	F
9	ID10	23	F

Step A for P = LHS 1

LHS

3	6	2	7

If size(LHS) >= 2 * 3
4 >= 6
false

Add **P** to final_partitions

3	6	2	7

Step A for P = RHS 2

RHS

4	8	1	9	0	5

If size(RHS) >= 2 * 3
6 >= 6
true

B. Select the most diverse QID in RHS
Age – 3/6 unique values
Gender – 2/6 unique values

qid_vaiability = [‘Age’, ‘Gender’]
Select ‘Age’

Continue....

PETs

Partitioning - Visualise

Assume K = 3

Row Index	ID	Age	Gender
0	ID01	24	M
1	ID02	23	M
2	ID03	21	M
3	ID04	20	F
4	ID05	22	M
5	ID06	24	F
6	ID07	20	M
7	ID08	21	F
8	ID09	22	F
9	ID10	23	F

Step A for P = LHS

1

LHS

3	6	2	7

If size(LHS) >= 2 * 3
4 >= 6
false

Add P to final_partitions

3	6	2	7

Step A for P = RHS

2

Find the median (M)

RHS

4	8	1	9	0	5
22	22	23	23	24	24

M = 23

Split the RHS into 2 partitions

LHS < 23

4	8

RHS > = 23

1	9	0	5

If size(LHS) < 3 OR size(RHS) < 3 = **true**

If more QIDs exist = **true**

Step B on RHS

Continue....

PETs

Partitioning - Visualise

Assume K = 3

Row Index	ID	Age	Gender
0	ID01	24	M
1	ID02	23	M
2	ID03	21	M
3	ID04	20	F
4	ID05	22	M
5	ID06	24	F
6	ID07	20	M
7	ID08	21	F
8	ID09	22	F
9	ID10	23	F

Step A for P = LHS 1

LHS

3	6	2	7

If size(LHS) >= 2 * 3
4 >= 6
false

Add **P** to final_partitions

3	6	2	7

Step A for P = RHS 2

B. Select the most diverse QID in RHS

qid_vaiability = [~~'Age'~~, 'Gender']

Select 'Gender'

Find the median (M) of RHS

Continue....

PETs

Partitioning - Visualise

Assume K = 3

Row Index	ID	Age	Gender
0	ID01	24	M
1	ID02	23	M
2	ID03	21	M
3	ID04	20	F
4	ID05	22	M
5	ID06	24	F
6	ID07	20	M
7	ID08	21	F
8	ID09	22	F
9	ID10	23	F

Step A for P = LHS 1

LHS

3	6	2	7

If size(LHS) >= 2 * 3
4 >= 6
false

Add **P** to final_partitions

3	6	2	7

Step A for P = RHS 2

Find the median (M)

RHS

4	8	1	9	0	5
M	F	M	F	M	F

unique_genders = ['M', 'F']

mask_lhs = ['M']   mask_rhs = ['F']

Split the RHS into 2 sub partitions

LHS has mask_lhs

4	1	0

RHS has mask_rhs

8	9	5

If size(LHS) < 3 OR size(RHS) < 3 = **false**

Step A for **P** = LHS and **P** = RHS

Continue....

PETs

Partitioning - Visualise

Assume K = 3

Row Index	ID	Age	Gender
0	ID01	24	M
1	ID02	23	M
2	ID03	21	M
3	ID04	20	F
4	ID05	22	M
5	ID06	24	F
6	ID07	20	M
7	ID08	21	F
8	ID09	22	F
9	ID10	23	F

1

Step A for P = LHS

LHS

3	6	2	7

If size(LHS) >= 2 * 3
4 >= 6
false

Add P to final_partitions

3	6	2	7

Step A for P = LHS

2

LHS

4	1	0

If size(LHS) >= 6
3 >= 6
false

Add LHS to final_partitions

4	1	0

Step A for P = RHS

RHS

8	9	5

If size(RHS) >= 6
3 >= 6
false

Add LHS to final_partitions

8	9	5

Stop the Partitioning Process

PETs
K-Anonymity

Assume K = 3

final_partitions

3	6	2	7

4	1	0

8	9	5

original

Row Index	ID	Age	Gender
0	ID01	24	M
1	ID02	23	M
2	ID03	21	M
3	ID04	20	F
4	ID05	22	M
5	ID06	24	F
6	ID07	20	M
7	ID08	21	F
8	ID09	22	F
9	ID10	23	F

partitioned

Row Index	ID	Age	Gender
3	ID04	20	F
6	ID07	20	M
2	ID03	21	M
7	ID08	21	F
4	ID05	22	M
1	ID02	23	M
0	ID01	24	M
8	ID09	22	F
9	ID10	23	F
5	ID06	24	F

anonymised

Row Index	ID	Age	Gender
3	ID04	20.5	*
6	ID07	20.5	*
2	ID03	20.5	*
7	ID08	20.5	*
4	ID05	23	M
1	ID02	23	M
0	ID01	23	M
8	ID09	23	F
9	ID10	23	F
5	ID06	23	F

PETs

In the Jupyter Exercise

def mondrian(k)

A. Select a partition **P** (at the start : all the rows)

If size(**P**) >= 2 * **K**

B. Select the most diverse QID in the partition → **def get_qid_variability(partition)**

Find the median of the QID values (**M**)

Split the partition into 2 sub partitions

LHS < **M** and RHS >= **M**

→ **def split(partition, column)**

If size(LHS) < **K** OR size(RHS) < **K**

If more QIDs exist

Step B for **P**: for remaining QIDs

else

Add **P** to final_partitions

else

Step A for **P** = LHS and **P** = RHS

else

Add **P** to final_partitions

Let’s Code