Privacy and Confidentiality

Daniela Hochfellner, Julia Lane and Frauke Kreuter

Overview

Intro to Privacy and Confidentiality
Why Data Access is Important
Legal Framework
Operational framework
How to Maintain Confidentiality
Big Data Challenges

What is …

Privacy

includes the famous “right to be left alone,” and the ability to share information selectively but not publicly (White House 2014)

Confidentiality

means “preserving authorized restrictions on information access and disclosure, including means for protecting personal privacy and proprietary information” (McCallister, Grance, and Scarfone 2010).

Why confidentiality important

Promise to respondents

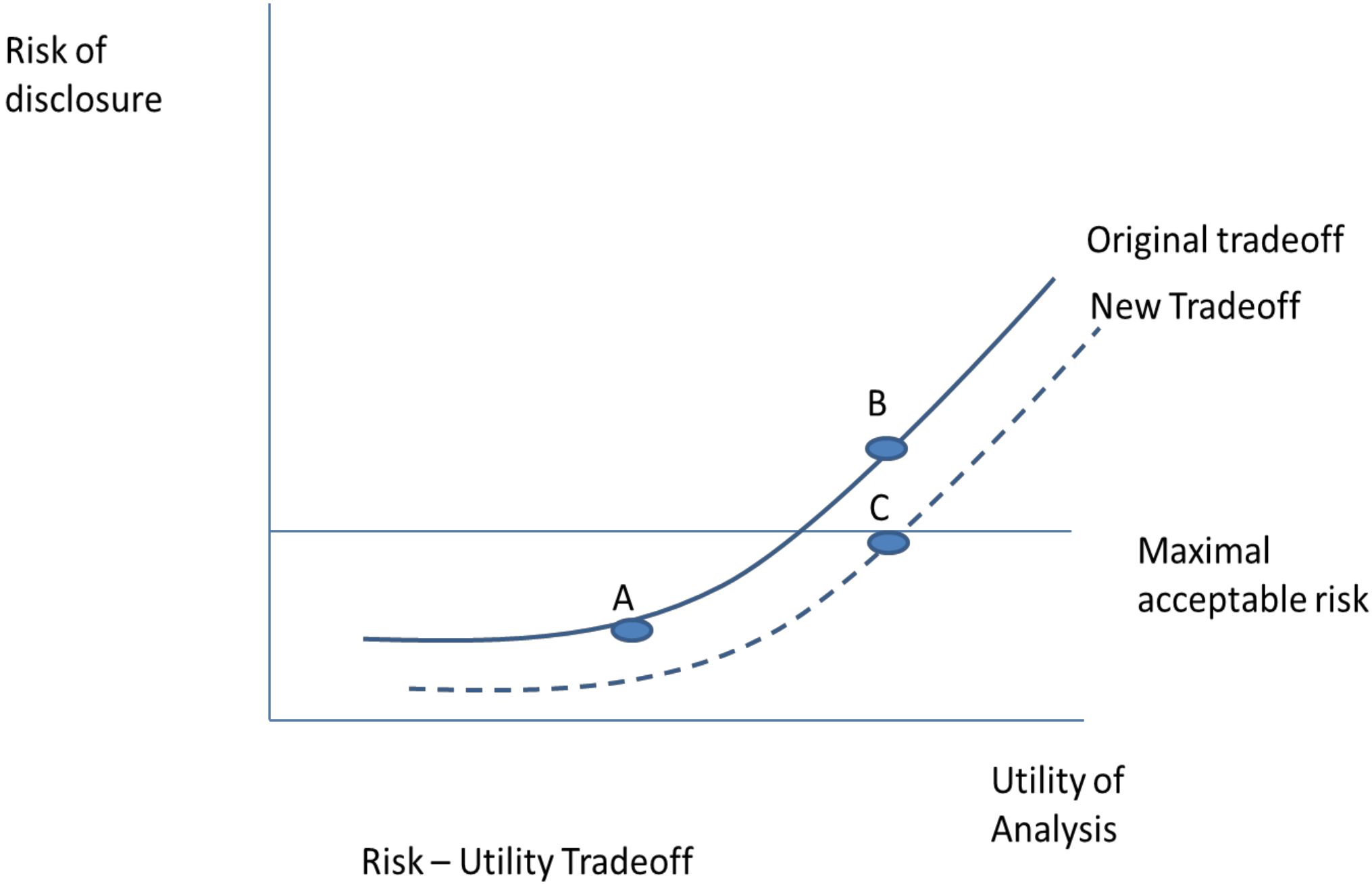
Ethical requirement

Legal requirement

Practical implications

Challenge

How to balance the *risk* of providing access with the associated utility?



Risk – Utility Tradeoff

Why is Access Important?

Research

Linkage validation

Replication

Building knowledge infrastructure

Legal framework

Data controlled by statistical agencies

- Title 26
- Title 13
- CIPSEA

Other frameworks

- HIPAA
- FERPA

Twin pillars of anonymization and consent

Operational framework

Valid research purpose
- statistical purpose
- need “research benefit”

Trusted researchers

Limits on data use
- Remote access to secure results
- Disclosure control of results

safe projects
+ safe people
+ safe setting
+ safe outputs

⇒ **safe use**

What We Did:

ADRF

FedRAMP

HOME ABOUT US PARTICIPATE MARKETPLACE RESOURCES LEARN BLOG

Cloud solutions allow for faster processing and more elasticity in computing in an on demand, more efficient platform. However, incorporating the cloud into our Federal IT infrastructure has proven difficult. Currently, there is a redundant, inconsistent, time consuming, costly, and inefficient risk management approach to cloud adoption. In addition, there is little incentive to leverage existing Authorizations to Operate (ATOs) among agencies. The Federal Government spends hundreds of millions of dollars a year securing the use of IT systems.

The solution? FedRAMP.

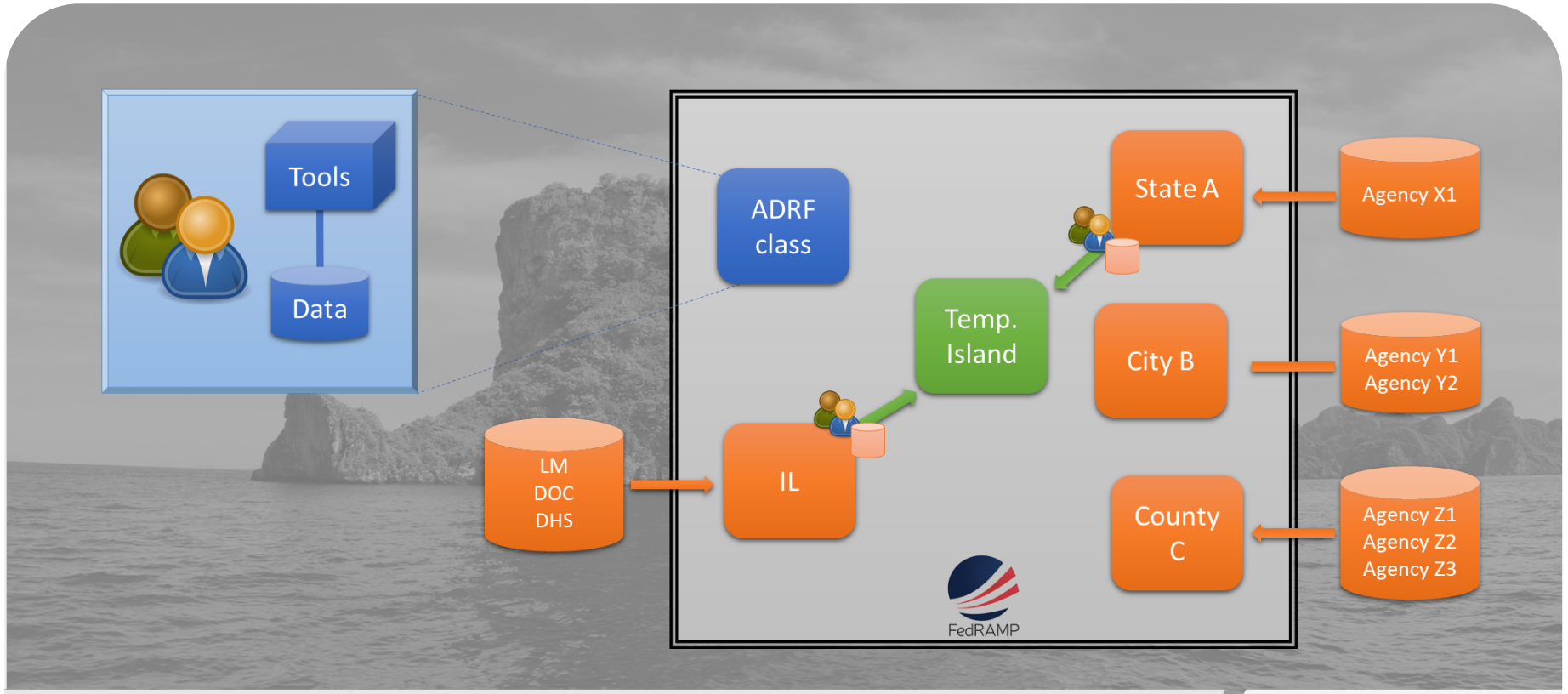
The Federal Risk and Authorization Management Program, or FedRAMP, is a government-wide program that provides a standardized approach to security assessment, authorization, and continuous monitoring for cloud products and services. This approach uses a "do once, use many times" framework that saves an estimated 30-40% of government costs, as well as both time and staff required to conduct redundant agency security assessments. FedRAMP is the result of close collaboration with cybersecurity and cloud experts from the General Services Administration (GSA), National Institute of Standards and Technology (NIST), Department of Homeland Security (DHS), Department of Defense (DOD), National Security Agency (NSA), Office of Management and Budget (OMB), the Federal Chief Information Officer (CIO) Council and its working groups, as well as private industry.

Administrative Data Research Facility (ADRF)

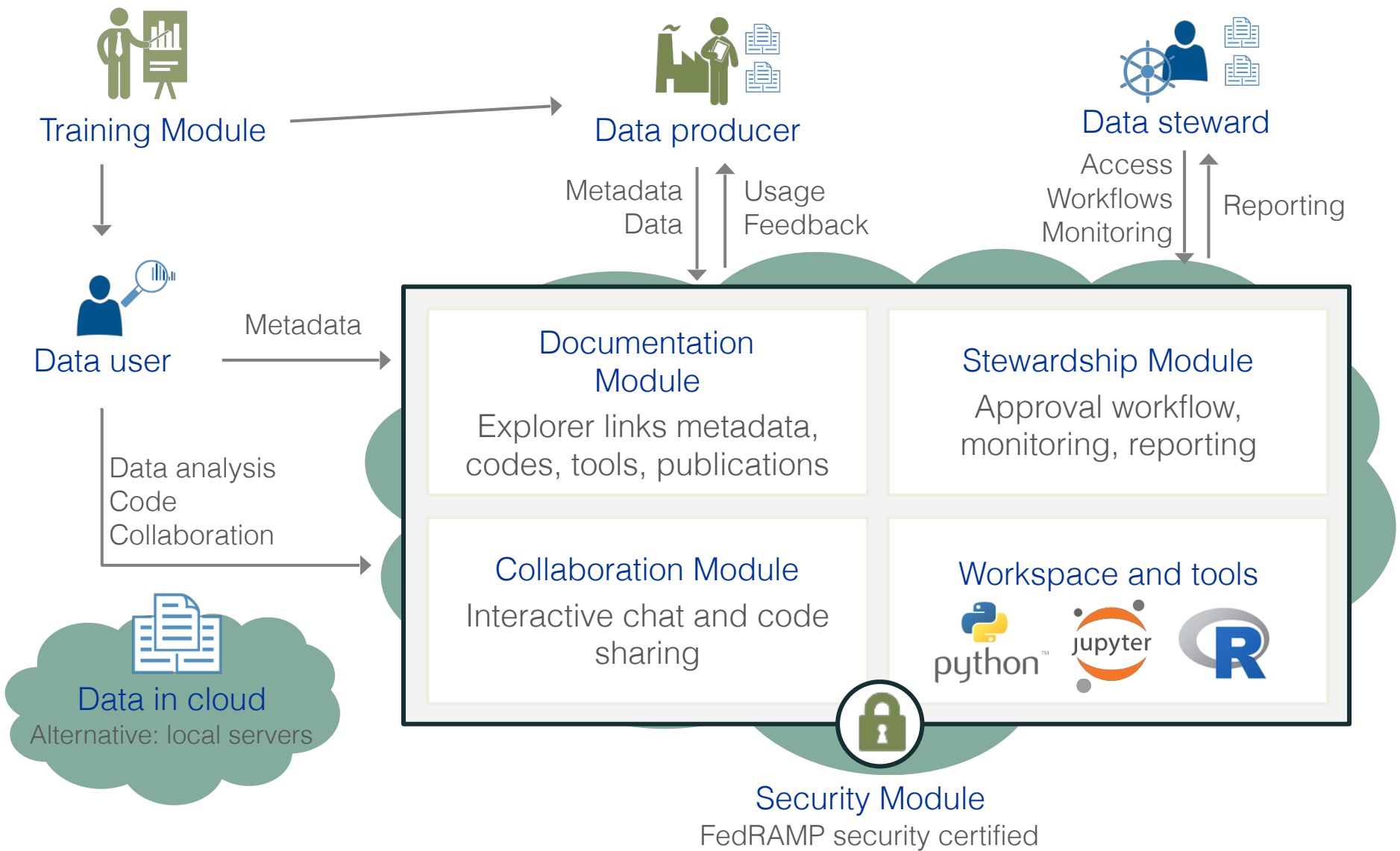
The (ADRF) provides a secure platform to host confidential micro-data. It is developed at New York University (NYU)

ADRF combines the business workflow of a research data centre with potentially interesting new ideas how to enhance user experience and engage researchers to contributing information about data

Data Security



The ADRF User Centric Approach



De-identification

De-identification procedure that replaces direct identifiers with artificial IDs that do not allow for re-identification

Applying an algorithm (hash function) according to NIST, FIPS Secure Hash Standards (FIPS PUB 180-4)

Adding a salt (secret or public)

Example

Original Data: Name: Peter Miller, SSN: 234-56-295, DOB: 07/12/76

Hashed Data:

Name:
9660ea4d6a0953035372678dd36da57a3e6f5c165605cbf73b6cfd778b7724a7

SSN:
6462c7069be22fc103f2edbad09b5763e0f1866af12ad9b578b8f9e1a89f8d87

DOB:
6c2c2cca5715ec5cd08bb7889dd12837976ead3b04201d88bc34173df52f7b11

Requires Pre-processing

Make sure IDs are unique, especially if there are different datasets that are supposed to be linked afterwards

Name harmonization through name standardization tables

Harmonize date formats

```
import datetime
import hashlib

class HashCache( object ):

    SALT_APPEND_RIGHT = "right"
    SALT_APPEND_LEFT = "left"

    def __init__( self, *args, **kwargs ):

        self.string_to_hash_map = {}
        self.cache_hit_count = 0
        print( "Cache initialized at " + str( datetime.datetime.now() ) )

    #-- END __init__() method --#

    def get_hash( self, string_to_hash_IN, salt_IN = "", append_salt_to_IN = SALT_APPEND_RIGHT ):

        '''
        Accepts string to hash and optional salt value.  If salt, appends it to the right of the string
            Then, checks to see if that string is already in the hash map.  If so, retrieves hash of it
            If not, hashes it and caches the hash.  Returns the hash, else None if there was an error.
        '''

        # return reference
        hash_OUT = ""

        # declare variables
        working_string = ""
        temp_hash = ""

        # pull string into working string:
        working_string = string_to_hash_IN

        # empty?
        if ( ( working_string is not None ) and ( working_string != "" ) and ( working_string != "NaN"

            # Got a value.  hash it.

            # is there a salt?
            if ( ( salt_IN is not None ) and ( salt_IN != "" ) ):

                # yes - use it.  Append to right or left?
                if ( append_salt_to_IN == self.SALT_APPEND_RIGHT ):

                    # right.
                    working_string = working_string + salt_IN

                else:

                    # if not right, left.
                    working_string = salt_IN + working_string

                #-- END check to see if right or left append --#

            #-- END check to see if salt. --#

            # check to see if hash in cache
            if ( working_string in self.string_to_hash_map ):

                # cached - record cache hit.
                self.cache_hit_count += 1

            else:

                # not cached.  Hash and cache.

                # encode to UTF-8
                temp_hash = working_string.encode( "utf-8" )

                # hash
                temp_hash = hashlib.sha256( temp_hash ).hexdigest()

                # cache
                self.string_to_hash_map[ working_string ] = temp_hash

            #-- END check to see if cached. --#

            # retrieve hash from cache.
            hash_OUT = self.string_to_hash_map[ working_string ]

        else:

            # No string in.  Leave empty.
            hash_OUT = ""

        #-- END check to see if empty --#


        return hash_OUT

    #-- END function get_hash() --#

#-- END class HashCache --#

print( "Object HashCache defined at " + str( datetime.datetime.now() ) )
```

Practicalities of Disclosure Control

The aim of disclosure control is to ensure that no unauthorized individual, technically competent with public data and private information could:

- Identify any information not already public knowledge with a reasonable degree of confidence, and
- Associate that information with the supplier of the information

What is Disclosure?

Identity disclosure occurs when an individual can be identified from the released output, leading to information being provided about that identified subject.

Attribute disclosure occurs when confidential information is revealed and can be attributed to an individual. It is not necessary for a specific individual to be identified or for a specific value to be given for attribute disclosure to occur. For example, publishing a narrow range for the salary of persons exercising a particular profession in one region may constitute a disclosure.

Residual disclosure can occur when released information can be combined to obtain confidential data.). Care must be taken to examine all output to be released. While a table on its own might not disclose confidential information, disclosure can occur by combining information from several sources, including external ones. (e.g., suppressed data in one table can be derived from other tables).

Historical Approach

1. Aggregated tabular data
2. Public use files
3. Licensing
4. Synthetic Data
5. Research Data Centers

Examples

Traditional approaches – tables

- cell suppression
- controlled tabular adjustment
- rounding
- cell perturbation

Traditional approaches – microdata

- local suppression
- global recoding
- top coding
- sampling
- rounding
- swapping
- added noise
- data shuffling

Specific Examples

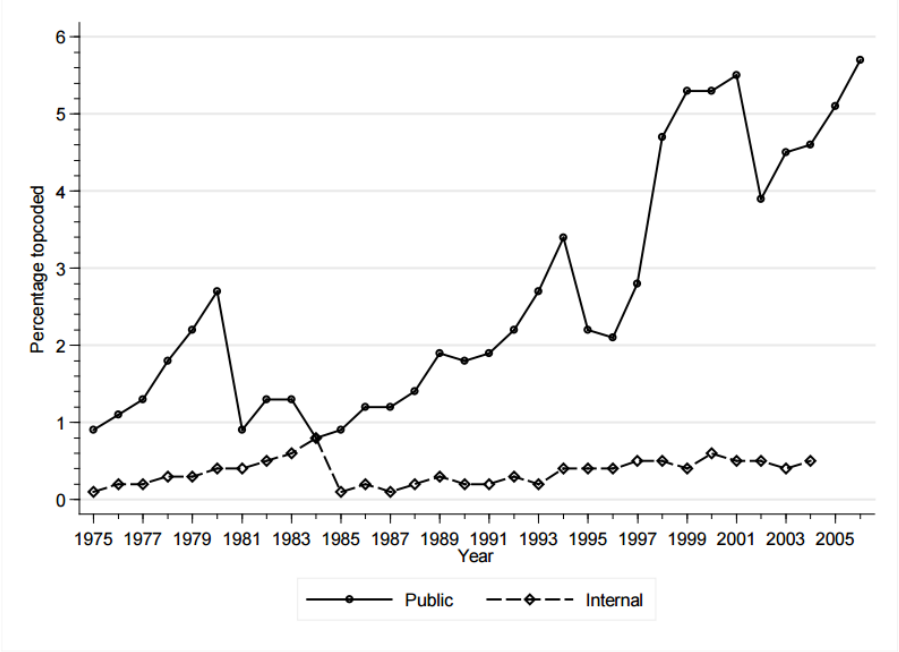
- Topcoding
  - Upper limit on values of a given variable, all cases above a certain part of the distribution are placed into one single category (wages)
  - Mean corrected topcoding: choose the value for topcoded cells that the mean of the distribution is correct
- Noise addition
  - Multiplying or adding a stochastic or randomized number
  - Multiplicative noise: generating random numbers with mean=1
  - Differential Privacy

Specific examples (contd)

- Grouping, aggregating
  - Geographic population thresholds
  - Sensitive variables (nationality
- Rounding (Age)
- Data Swapping
  - Introduce uncertainty, does not change the marginal distribution, but it distorts joint distributions of swapped and unswapped variables

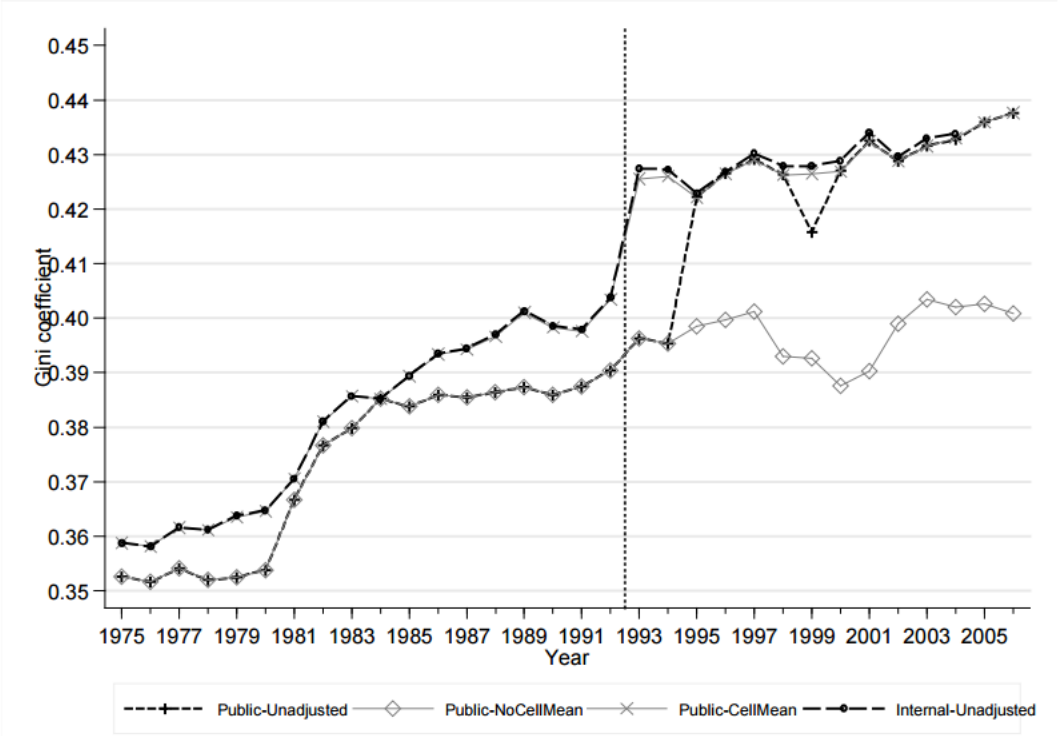
Consequences

Figure 1: Percentage of Individuals with Censored Household Income in March CPS, by Year



Source: authors' calculations from internal and public use data files of March CPS. Internal data were not available for years after 2004.

Figure 2: Gini Coefficient Estimates Derived Using Four Censoring Adjustment Methods



Source: authors' calculations from internal and public use data files of the March CPS. There was a major change in CPS data collection methods between 1992 and 1993. Internal data were not available for years after 2004. See text for definitions of the series.

Burkhauser, Richard V., Feng, Shuaizhang, Jenkins, Stephen P. and Larrimore, Jeff (2010) Estimating trends in US income inequality using the Current Population Survey: the importance of controlling for censoring. The Journal of Economic Inequality, 9 (3). pp. 393-415 ISSN 1569- 1721 DOI: 10.1007/s10888-010-9131-6

Practicalities of Disclosure Control

Primary
looking at individual cells

Secondary
combining data from different tables and sources
using non-suppressed information to infer things

Most disclosure control is very context-specific

Practicalities of disclosure control: primary disclosure

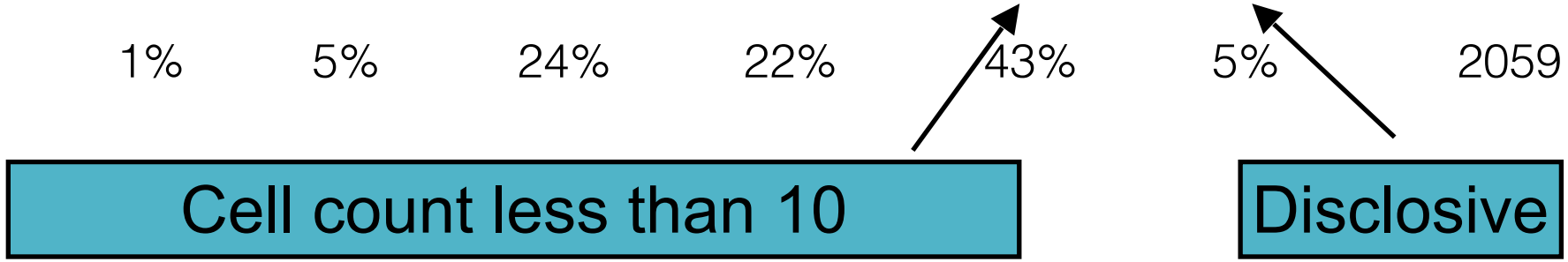
1) Threshold Rule
   no cells with less than 10 units (individuals/enterprises)
   Note: local unit analysis must show the enterprise count
   This rule is applied even when there is no information associated with each cell

Example: manufacturing firms with over 1,000 employees by region

Region	Number of firms
North	152
South	8
East	12
West	6

% breakdown of hourly earnings by occupation

Pay bands: per hour	$5 to $6	$6 to $7	$7 to $8	$8 to $9	$9 to $10	>$10	Total numbers
Mechanics	15%	13%	32%	25%	10%	5%	1846
Nurses	13%	22%	57%	7%	1%	0%	949
Bankers	1%	5%	24%	22%	43%	5%	2059



Class disclosure: 0% values and 100% values are both problematic

Variable	Obs	Mean	Std. Dev	Min	Max
% PC users	3439	0.32	0.341	0	1
employees	3439	1413.7	5379.95		
Sales	3439	183323.7	694490.9		
Firm age	3439	6.5	2.08	1	15

Remove cells

?

No max/min unless shown to be uninformative

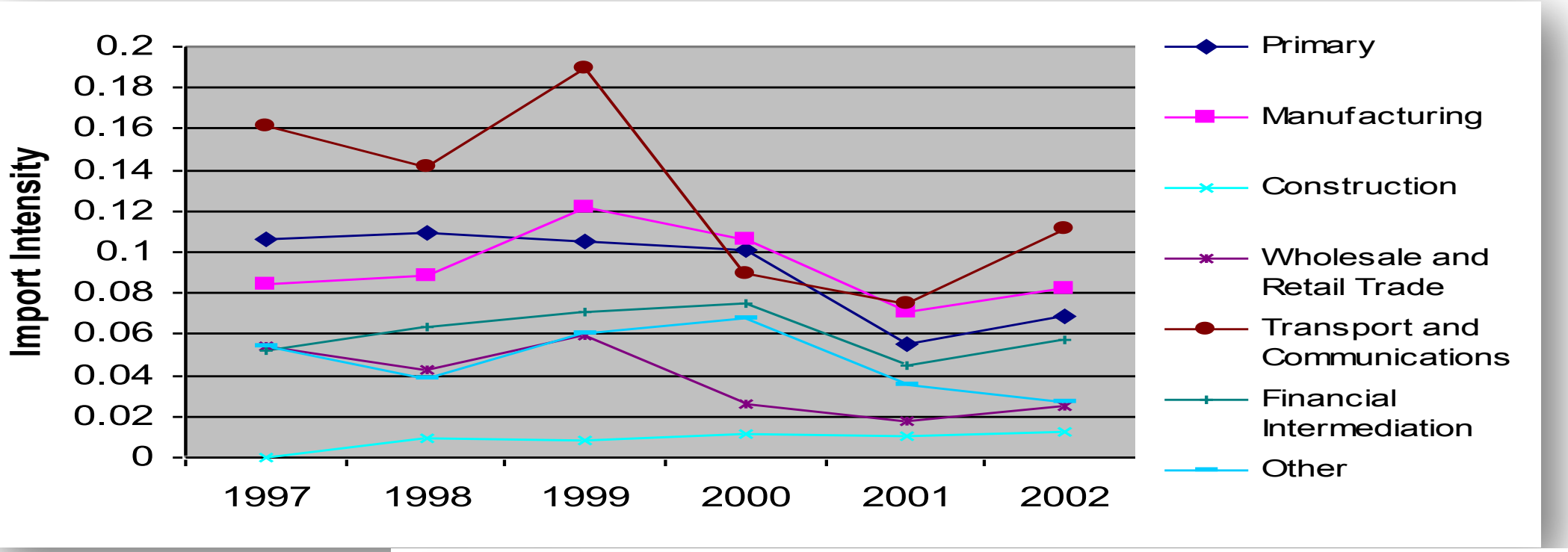


Table of frequencies

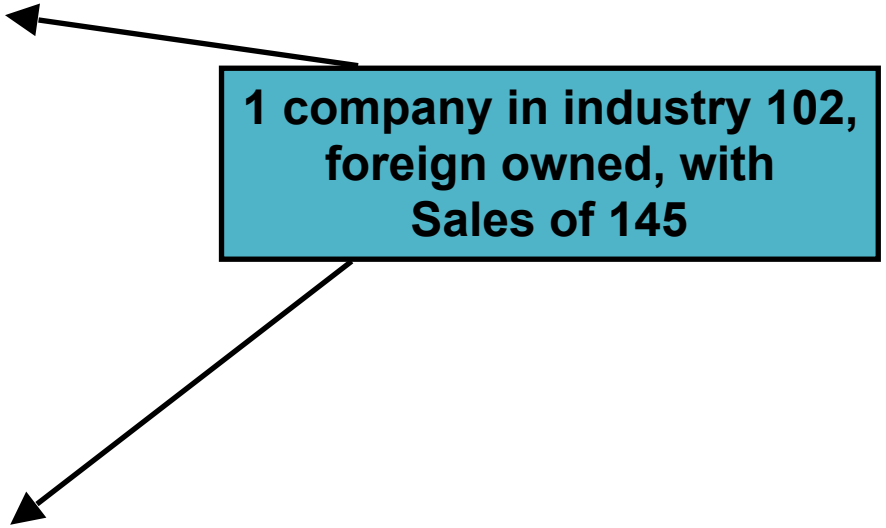
	1997	1998	1999	2000	2001	2002
Primary	69	89	88	76	170	157
Manufacturing	2,764	2,149	1,570	1,756	3,863	3,850
Construction	395	377	480	418	382	410
Wholesale and Retail Trade	209	319	487	494	1,314	1,301
Transport and Communications	23	19	18	35	84	111
Financial Intermediation	987	973	1,223	1,182	2,198	2,364
Other	83	97	137	142	409	398

Sales: US-based companies

Industry	Companies	Sales
101	11	12003
102	10	5434
103	15	45644

Sales: All companies

Industry	Companies	Sales
101	14	16013
102	11	5579
103	19	65744



Sales: by Region			
Country	Industry	Companies	Sales
North	101	26	16013
	102	31	74379
	103	50	60321
South	101	11	7284
	102	12	20301
	103	15	15124
East	101	14	8742
	102	19	28554
	103	20	20199

Sales: All companies		
Industry	Companies	Sales
101	51	32003
102	63	124434
103	85	95644

One company in West in industry 102 with sales of 1200...

Average Earnings by occupation: all job types		
Occupation	Av. Earnings	Number of Individuals
Plumbers	18,000	98
Dentists	39,500	11
Lawyers	47,000	54

Average Earnings by occupation: Full-time jobs		
Occupation	Av. Earnings	Number of Individuals
Plumbers	19,500	70
Dentists	42,500	10
Lawyers	47,000	54

some simple math to work out what the part-time dentist earns:

11 * 39.5 = 434.5 10 * 42.5 = 425 434.5 – 425 = **9.5**

What about Regressions?

Regressions are generally safe

Regressions could be worrisome if:

- Only on dummies = a table
- Potentially disclosive by differencing
- Hiding coefficients makes linear and non-linear estimation completely non-disclosive
- Panel models inherently safe

In Big Data Era

Most data no longer collected by the government (internet search logs, Twitter, supermarket scanners…)

Question how to share collected information without violating privacy guarantees becomes more relevant

Additional Problems

What is the legal framework when the ownership of data is unclear?

Collection and analysis often no longer within same entity. Ownership of data less clear.

Who has the legal authority to make decisions about permission, access and dissemination and under what circumstances?

The challenge in the case of big data is that data sources are often combined, collected for one purpose and used for another and users often have no good understanding of it or how their data will be used.

=> Concepts Out of Date

Notification is either comprehensive or comprehensible, but not both. (Nissenbaum 2011)

Understanding of the nature of harm has diffused over time..

Consumers value their own privacy in variously flawed ways. (Acquisti 2014)

Solution: Differential Privacy?

Differential privacy is a rigorous mathematical definition of privacy

An algorithm is said to be differentially private if by looking at the output, one cannot tell whether any individual's data was included in the original dataset or not.

The guarantee of a differentially private algorithm is that its behavior hardly changes when a single individual joins or leaves the dataset

This guarantee holds for any individual and any dataset

What is the DP guarantee?

Researchers selected a sample of individuals to participate in a survey exploring the relationship between socioeconomic status and medical outcomes across a number of U.S. cities. Individual respondents were asked to complete a questionnaire covering topics such as where they live, their finances, and their medical history. One of the participants, John, is aware that individuals have been re-identified in previous releases of de-identified data and is concerned that personal information he provides about himself, such as his HIV status or annual income, could one day be revealed in de-identified data released from this study. If leaked, the personal information John provides in response to the questionnaire used in this tudy could lead to an increase in his life insurance premium or an adverse decision on a mortgage application he submits in the future.

Differential Privacy can be used to address John's concerns

Study is designed to be differentially private if John is guaranteed that even though his information used the outcome will not disclose anything that is specific to him

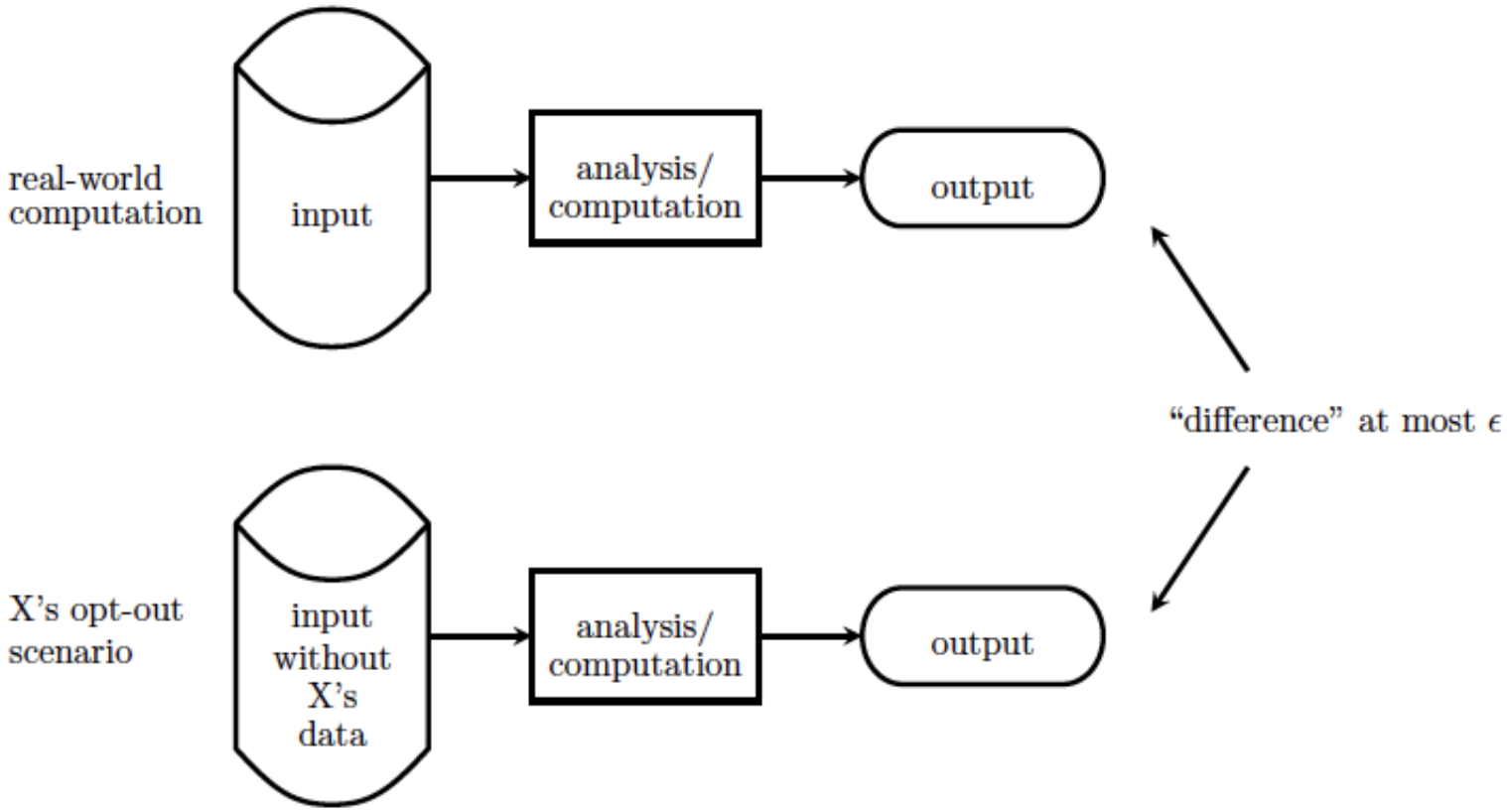
John's opt out scenario: The analysis is performed without including John's data.

Real world scenario: The analysis is performed with all people's data

DP: protect John in the real world scenario in a way that mimics the privacy protection of his opt out scenario

Is achieved by adding randomness -> output is not exact but approximation

How Do We Add Randomness



Epsilon: privacy loss parameter

Captures deviation between opt-out and real world scenario

The effect of each individual's information on the output of the analysis

Smaller value is more privacy (0 = opt-out scenario)

Consider computing an estimate of the number of HIV-positive individuals in a sample, where the sample contains $n = 10,000$ individuals of whom $m = 38$ are HIV-positive. In a differentially private version of the computation, random noise Y is introduced into the count so as to hide the contribution of a single individual. That is, the result of the computation would be $m' = m + Y = 38 + Y$ instead of $m = 38$.

A researcher uses the estimate m', as defined in the previous example, to approximate the fraction p of HIV-positive people in the population. The computation would result in the estimate

$$p' = \frac{m'}{n} = \frac{38 + Y}{10,000}.$$

For instance, suppose the sampled noise is $Y = 4.2$. Then, the estimate would be

$$p' = \frac{38 + Y}{10,000} = \frac{38 + 4.2}{10,000} = \frac{42.2}{10,000} = 0.42\%,$$

whereas without added noise, the estimate would have been $p = 0.38\%$.