INTRODUCTION TO DATA SCIENCE

JMCT

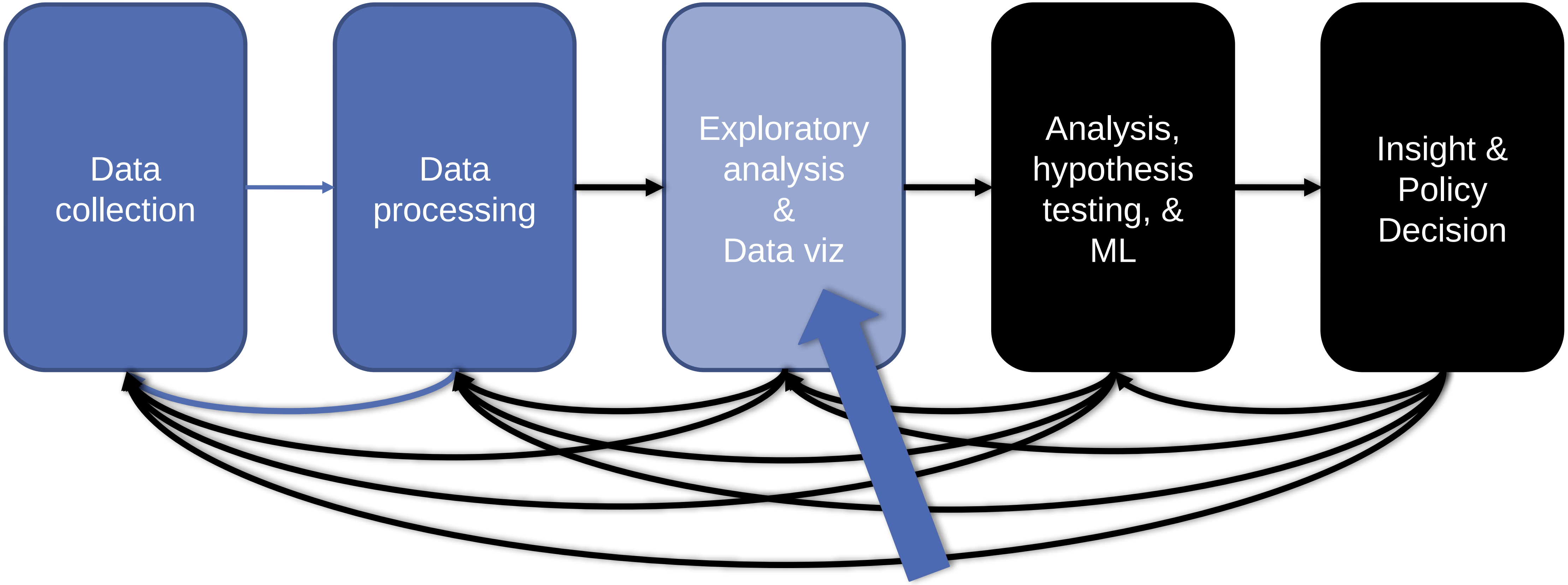
Lecture #12+13 – 03/02/2022

CMSC320
Mondays & Wednesdays
5:00pm – 6:15pm

https://cmsc320.github.io/

COMPUTER SCIENCE
UNIVERSITY OF MARYLAND

THE DATA LIFECYCLE


Data collection
Data processing
Exploratory analysis & Data viz
Analysis, hypothesis testing, & ML
Insight & Policy Decision

(Mainly.)

TODAY'S LECTURE

Missing Data …

- What is it?
- Simple methods for imputation

… with a tiny taste of Stats/ML lecturers to come.

Wild MISSINGNO.
appeared!

Thanks to John Atwood and Wenjiang Fu

MISSING DATA

Missing data is information that we want to know, but don’t

It can come in many forms, e.g.:

- People not answering questions on surveys
- Inaccurate recordings of the height of plants that need to be discarded
- Canceled runs in a driving experiment due to rain

Could also consider missing columns (no collection at all) to be missing data …

[JA]

KEY QUESTION

Why is the data missing?

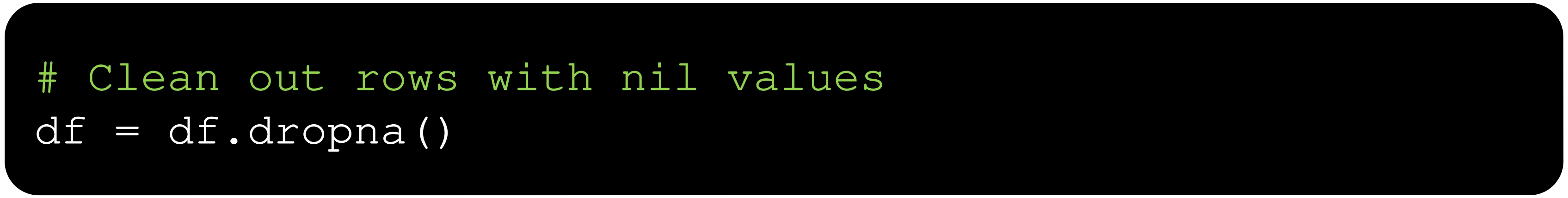
- What mechanism is it that contributes to, or is associated with, the probability of a data point being absent?
- Can it be explained by our observed data or not?

The answers drastically affect what we can ultimately do to compensate for the missing-ness

[JA]

COMPLETE CASE ANALYSIS

Delete all tuples with any missing values at all, so you are left only with observations with all variables observed



```
# Clean out rows with nil values
df = df.dropna()
```

Default behavior for libraries for analysis (e.g., regression)

- We’ll talk about this much more during the Stats/ML lectures

This is the simplest way to handle missing data. In some cases, will work fine; in others, ?????????????:

- Loss of sample will lead to variance larger than reflected by the size of your data
- May bias your sample

[JA]

RECALL: SAMPLES, POPULATION, & MODELS

Quick aside (remember back to Jon Snow & cholera ..)!

We want to build:

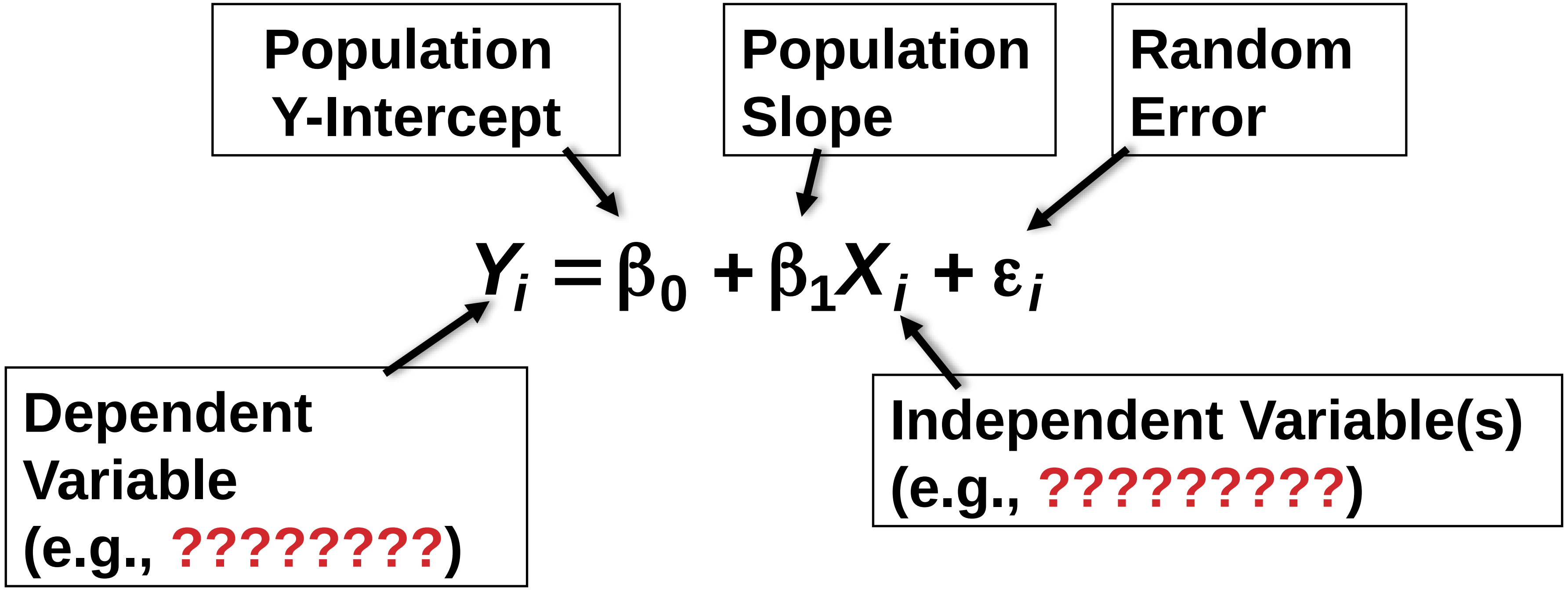
- **Statistical models – given some statistical assumptions, what can we say about the generation of sample data (and, hopefully, generalize to population)**
- **Often want to use these statistical models as, or to build, predictive models – use statistics to predict some unknown event (e.g., in the future)**

Spoiler: missing data can influence these models, *and* models can be used to “fill in” aka impute missing data (more later)

LINEAR REGRESSION

Assumption: relationship between variables is linear:

- (We'll relax linearity, study in more depth later.)



Population Y-Intercept

Population Slope

Random Error

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$$

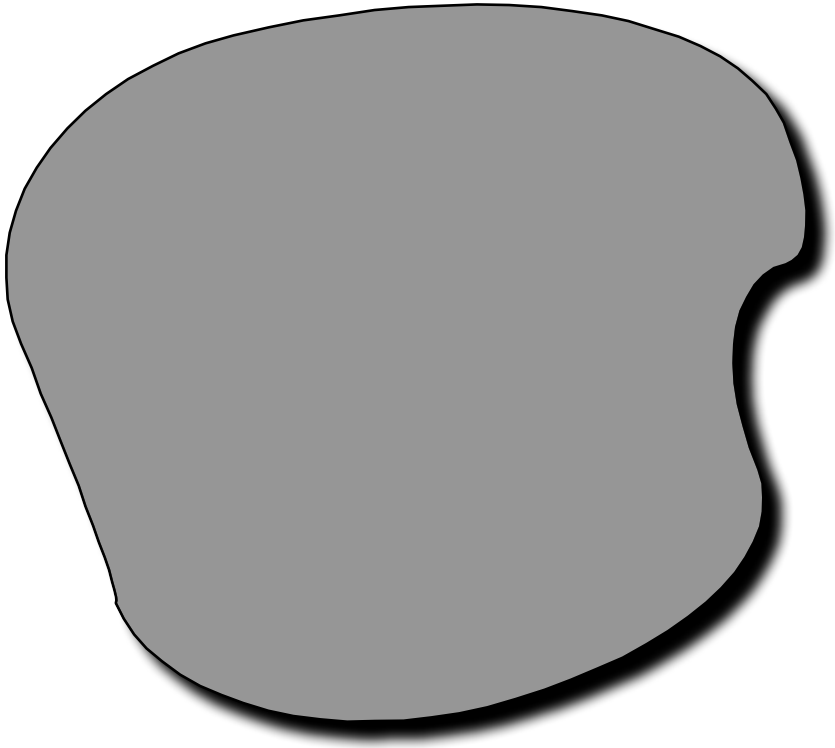
Dependent Variable (e.g., ????????)

Independent Variable(s) (e.g., ?????????)

[WF]

POPULATION & SAMPLE REGRESSION MODELS

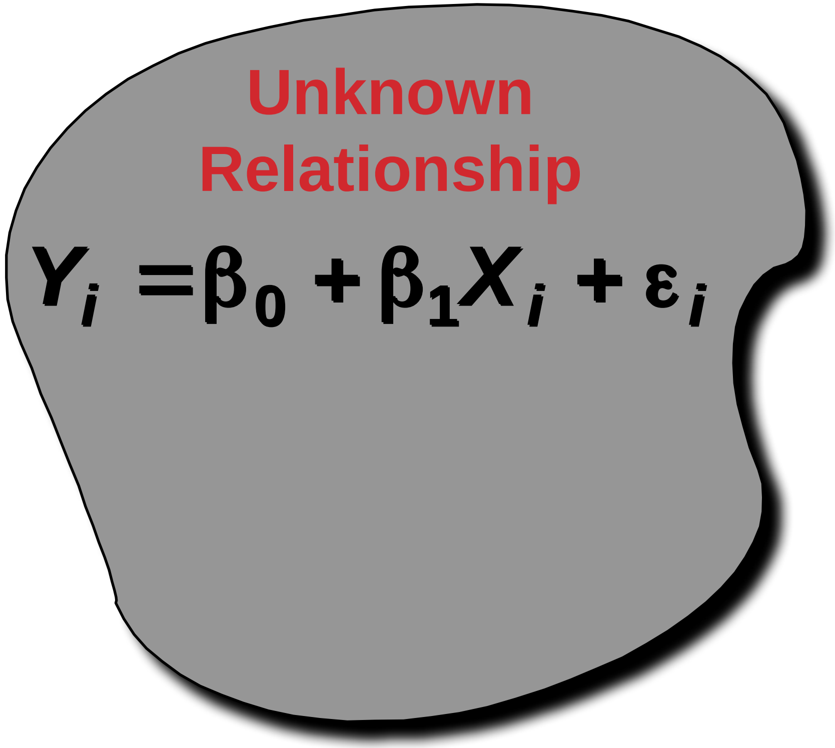
Population



[WF]

POPULATION & SAMPLE REGRESSION MODELS

Population

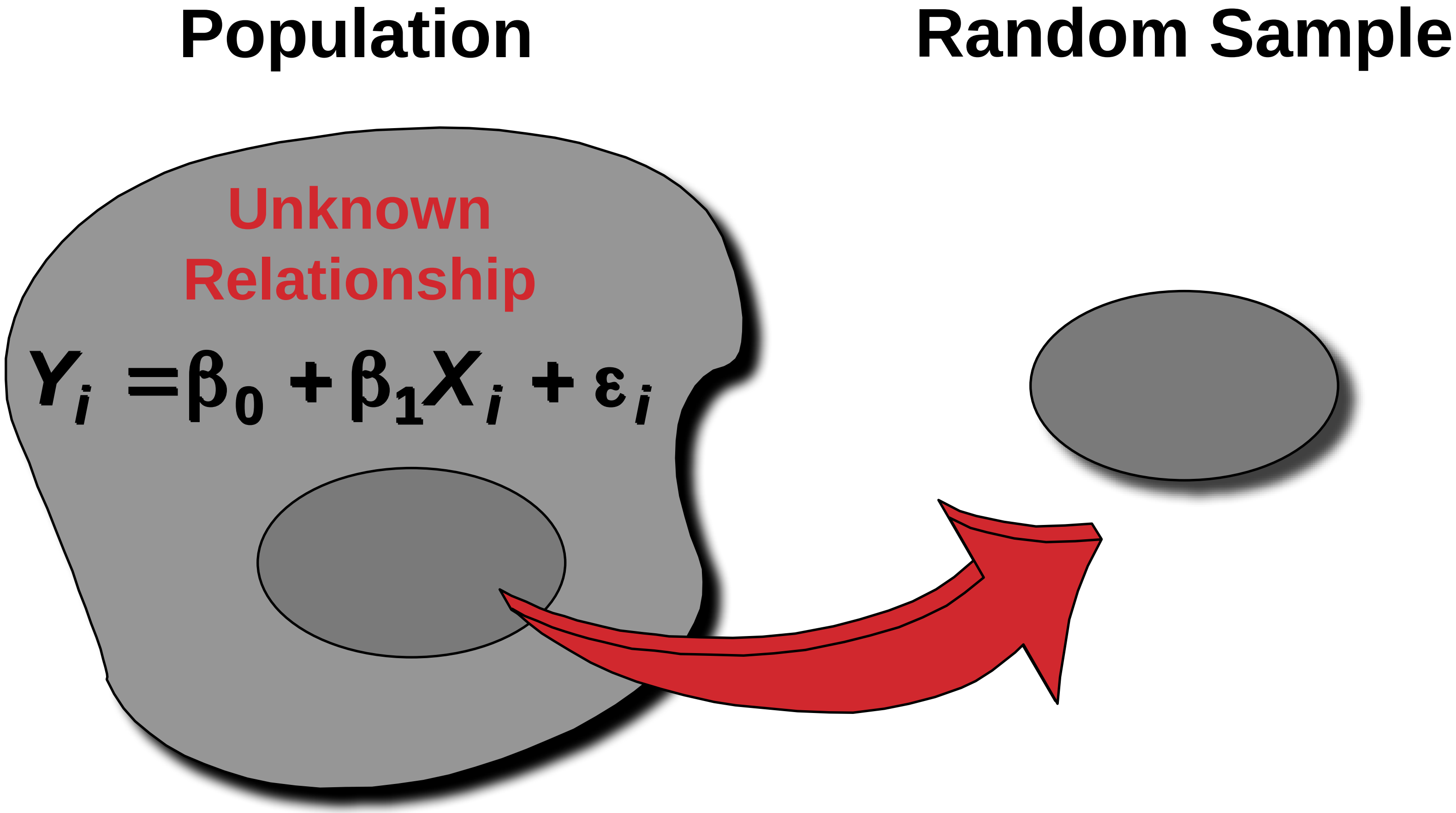


Unknown Relationship

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$$

[WF]

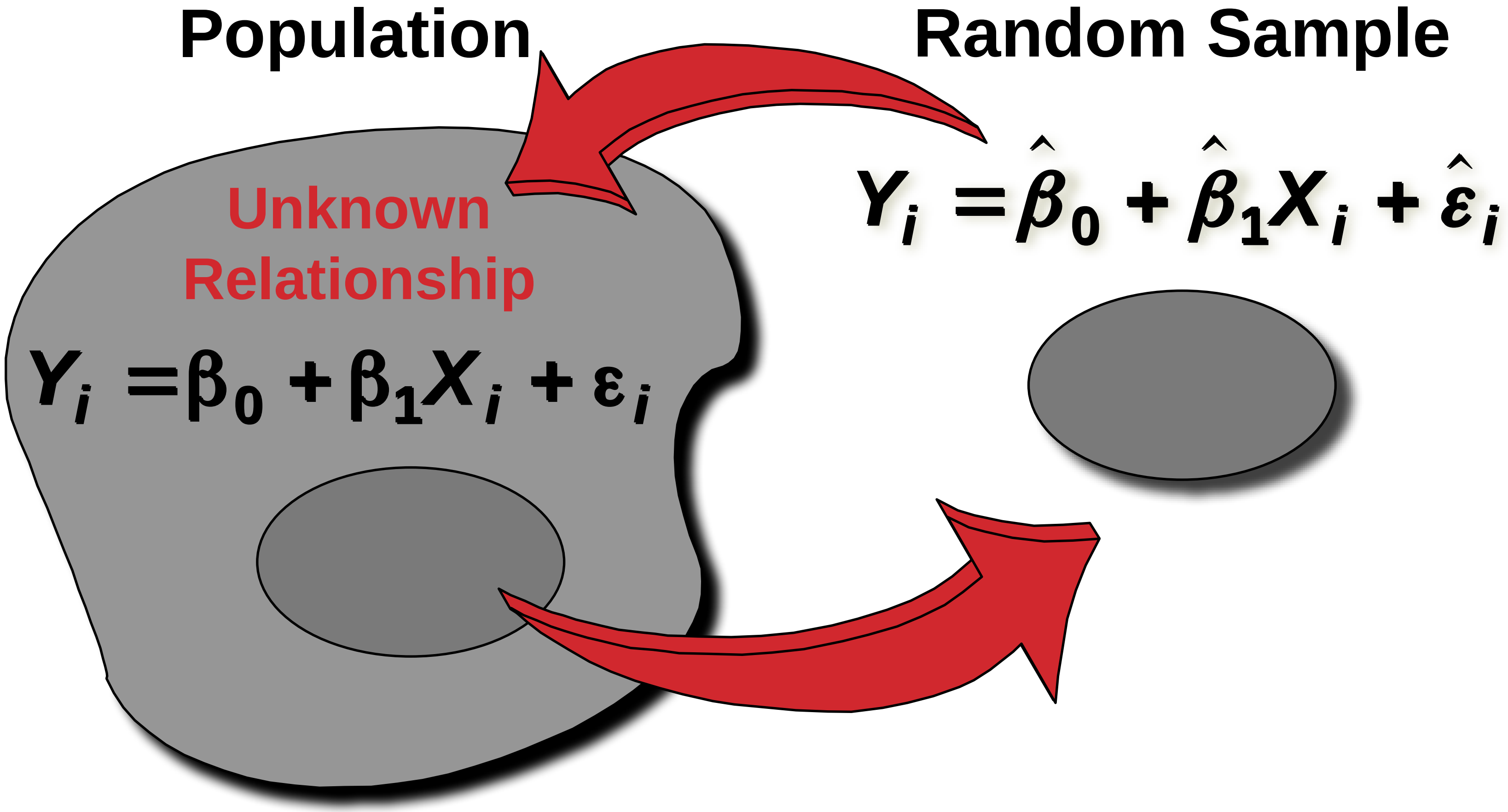
POPULATION & SAMPLE REGRESSION MODELS



$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$$

[WF]

POPULATION & SAMPLE REGRESSION MODELS

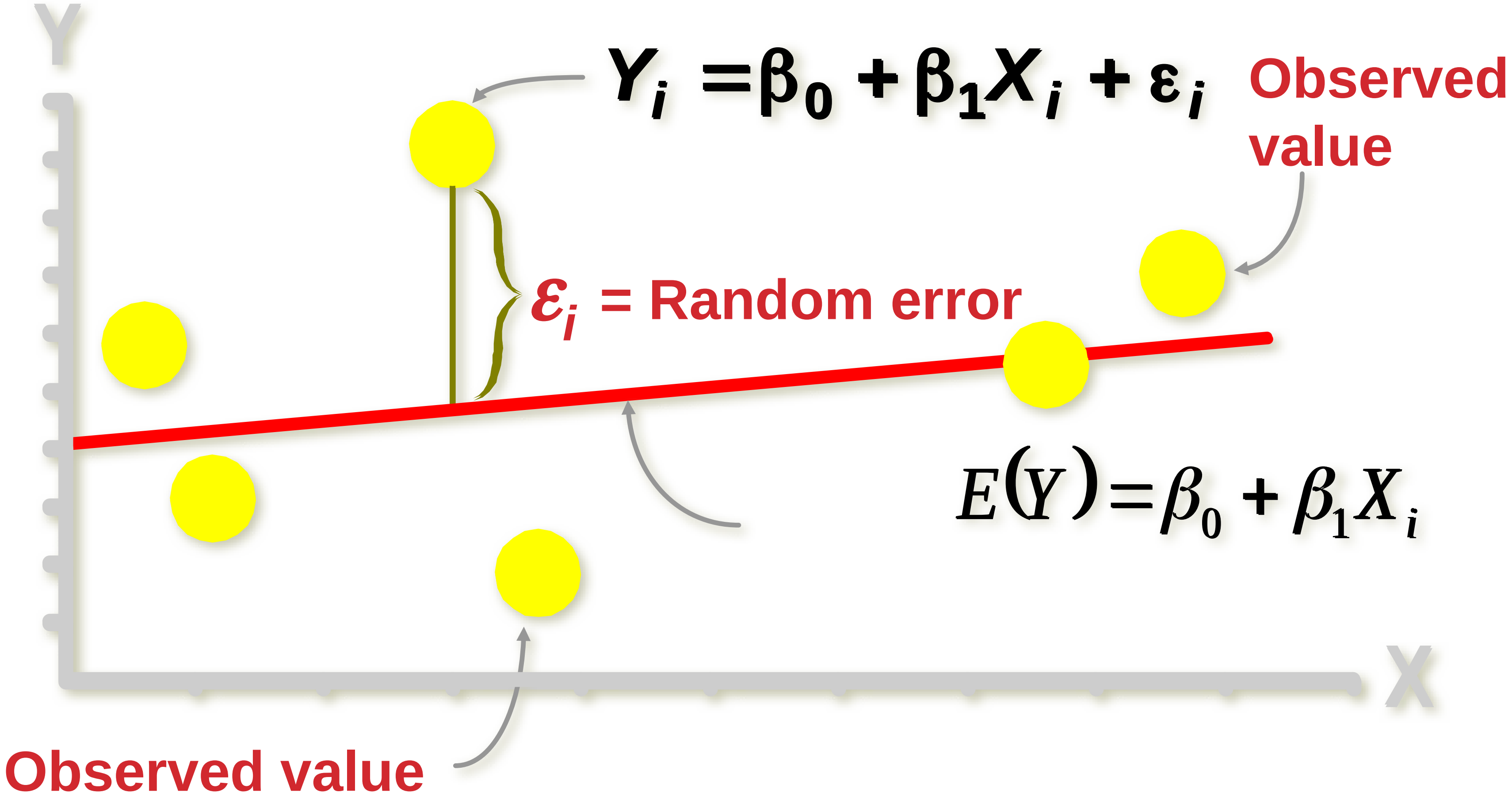


$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$$

$$Y_i = \hat{\beta}_0 + \hat{\beta}_1 X_i + \hat{\varepsilon}_i$$

[WF]

LINEAR REGRESSION



$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$$

Observed value

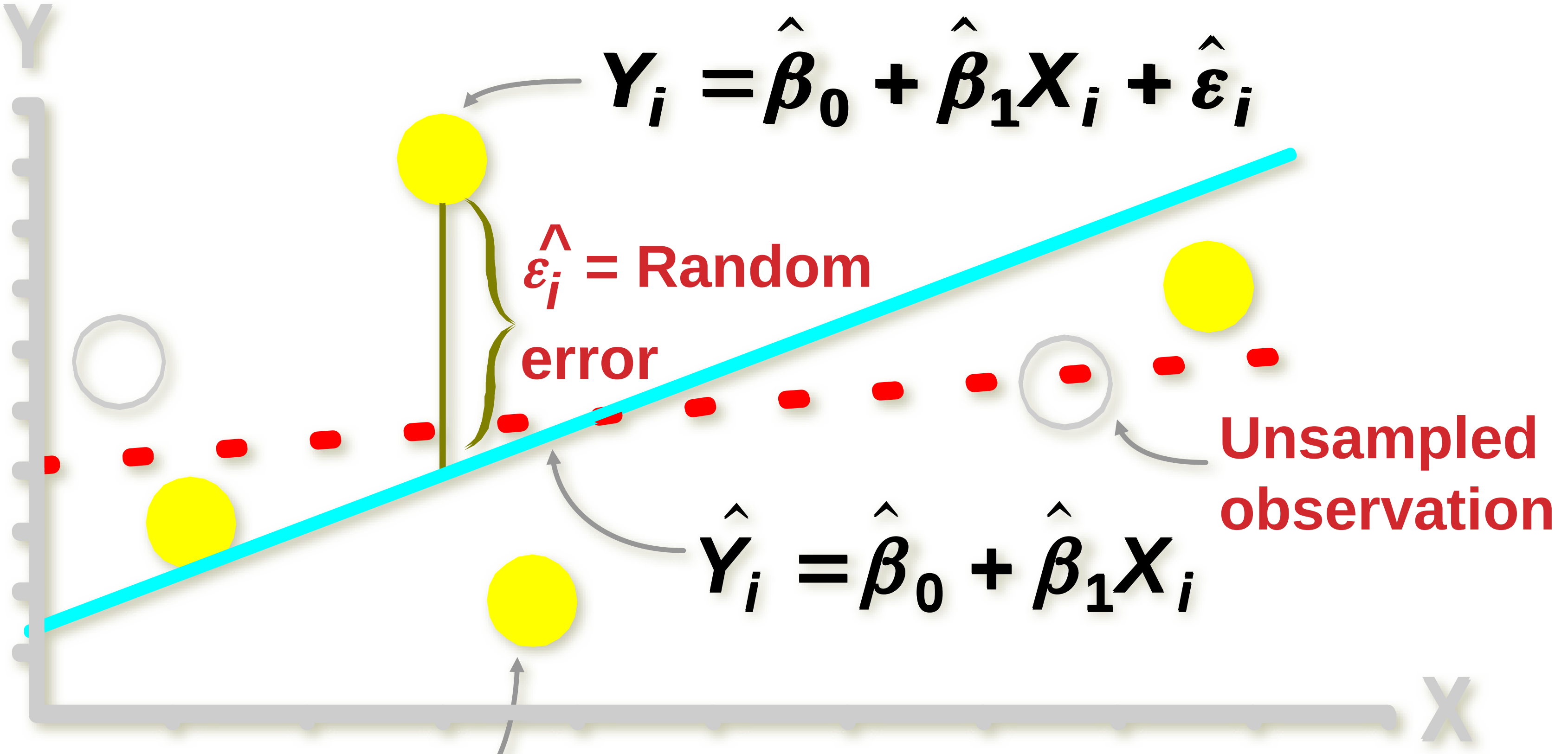
ε_i = Random error

$$E(Y) = \beta_0 + \beta_1 X_i$$

Observed value

[WF]

SAMPLE LINEAR REGRESSION MODEL



$$Y_i = \hat{\beta}_0 + \hat{\beta}_1 X_i + \hat{\varepsilon}_i$$

$\hat{\varepsilon}_i$ = Random error

$$\hat{Y}_i = \hat{\beta}_0 + \hat{\beta}_1 X_i$$

Unsampled observation

Observed value

[WF]

ESTIMATING PARAMETERS: LEAST SQUARES METHOD

?

LEAST SQUARES

Best fit: difference between the true Y-values and the estimated Y-values is minimized:

- Positive errors offset negative errors …
- … square the error!

$$\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2 = \sum_{i=1}^{n}\hat{\varepsilon}_i^2$$

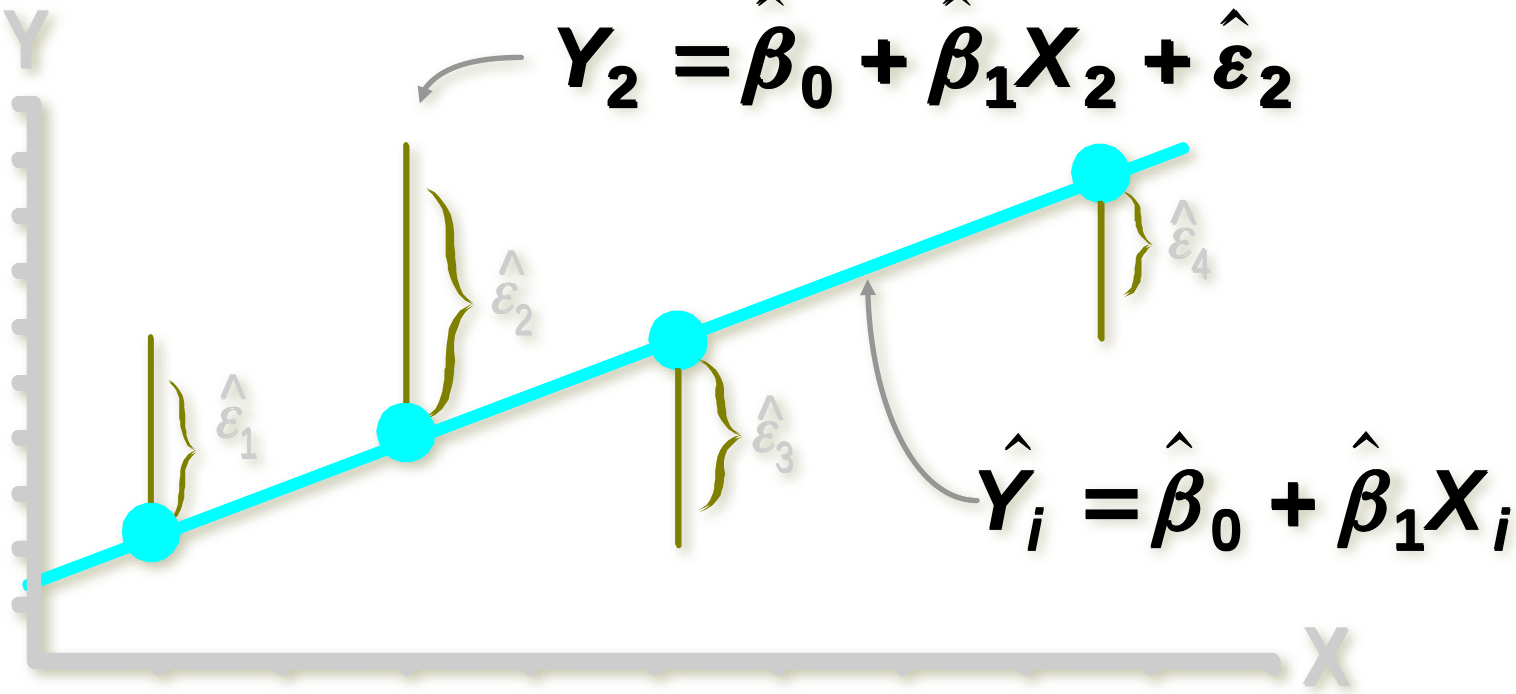
Least squares minimizes the sum of the squared errors

- Why squared? We'll cover this in more depth in a few weeks.
- Until then: http://www.benkuhn.net/squared

[WF]

LEAST SQUARES, GRAPHICALLY

LS minimizes $\sum_{i=1}^{n}\hat{\varepsilon}_i^2 = \hat{\varepsilon}_1^2 + \hat{\varepsilon}_2^2 + \hat{\varepsilon}_3^2 + \hat{\varepsilon}_4^2$


Y
X
$Y_2 = \hat{\beta}_0 + \hat{\beta}_1 X_2 + \hat{\varepsilon}_2$
$\hat{\varepsilon}_1$
$\hat{\varepsilon}_2$
$\hat{\varepsilon}_3$
$\hat{\varepsilon}_4$
$\hat{Y}_i = \hat{\beta}_0 + \hat{\beta}_1 X_i$

[WF]

INTERPRETATION OF COEFFICIENTS

Slope ($\hat{\beta}_1$):

- Estimated *Y* changes by $\hat{\beta}_1$ for each unit increase in *X*
- If $\hat{\beta}_1$ = 2, then *Y* is expected to increase by 2 for each 1 unit increase in *X*

Y-Intercept ($\hat{\beta}_0$)

- Average value of *Y* when *X* = 0
- If $\hat{\beta}_0$ = 4, then average *Y* is expected to be 4 when *X* Is 0

In-depth analysis
of less naïve
approaches to
come!

[WF]

NOW, BACK TO MISSING DATA …

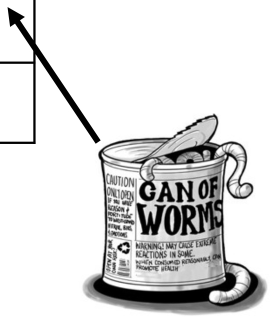
EXAMPLE

Question: Does the circumference of {Neck, Forearm, Wrist} & {Age} predict BF %?

Assumption: BF% is a linear function of measurements of {Neck, Forearm, Wrist, Age} …

Analysis: Results from a regression model with BF% …

Predictor	Estimate	S.E.	p-value
Age	0.0626	0.0313	0.0463
Neck	-0.4728	0.2294	0.0403
Forearm	0.45315	0.1979	0.0229
Wrist	-1.6181	0.5323	0.0026



(Interpretation ???????????)

[JA]

WHAT IF DATA WERE MISSING?

In this case, the dataset is complete:

- But what if 5 percent of the participants had missing values? 10 percent? 20 percent?

What if we performed complete case analysis and removed those who had missing values?

Let’s consider the effect if we do this if when the data is missing completely at random (MCAR)

- Removed cases at random, rerun analysis
- No systematic difference exist between rows (e.g., participants) with missing data and those without unbiased estimates
- Still losing sample size, increasing variance, etc. Also, MCAR is very unlikely to occur in reality!

[JA]

TYPES OF MISSING-NESS

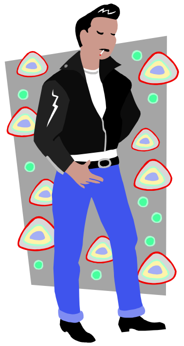
Missing Completely at Random (MCAR)

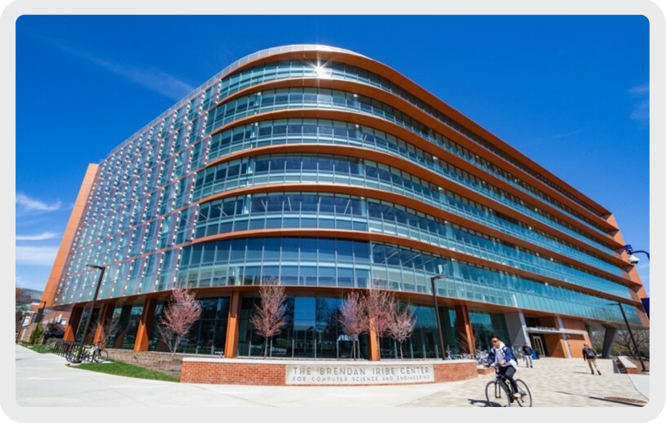
Missing at Random (MAR)

Missing Not at Random (MNAR)

WHAT DISTINGUISHES EACH TYPE OF MISSING-NESS?

Suppose you're loitering outside of Iribe one day …





Students just received their mid-semester grades

You start asking passing students their CMSC131 grades

- **You don't force them to tell you or anything**
- **You also write down their height (>6ft or not) and hair color**

(Please don't do this.)

YOUR SAMPLE

Hair Color	> 6ft	Grade
Red	Y	A
Brown	N	A
Black	N	B
Black	Y	A
Brown	Y	
Brown	Y	
Brown	N	
Black	Y	B
Black	Y	B
Brown	N	A
Black	N	
Brown	N	C
Red	Y	
Red	N	A
Brown	Y	A
Black	Y	A

Summary:

- 7 students received As
- 3 students received Bs
- 1 student received a C

Nobody is failing!

- But 5 students did not reveal their grade …

[JA]

WHAT INFLUENCES A DATA POINT’S PRESENCE?

Same dataset, but the values are replaced with a “0” if the data point is observed and “1” if it is not

Question: for any one of these data points, what is the probability that the point is equal to “1” …?

What type of missing-ness do the grades exhibit?

Hair Color	>6ft	Grade
0	0	0
0	0	0
0	0	0
0	0	0
0	0	1
0	0	1
0	0	1
0	0	0
0	0	0
0	0	0
0	0	1
0	0	0
0	0	1
0	0	0
0	0	0
0	0	0

MCAR: MISSING COMPLETELY AT RANDOM

If this probability is not dependent on any of the data, observed or unobserved, then the data is Missing Completely at Random (MCAR)

Suppose that X is the observed data and Y is the unobserved data. Call our "missing matrix" R

Then, if the data are MCAR, P(R|X,Y) = ??????????

$$P(R|X,Y) = P(R)$$

Probability of those rows missing is independent of anything.

[JA]

EXAMPLES?

?

APPLICABILITY OF MCAR

A completely random mechanism for generating missing-ness in your data set just isn't very realistic

Usually, missing data is missing for a reason:

- **Maybe older people are less likely to answer web-delivered questions on surveys**
- **In longitudinal studies people may die before they have completed the entire study**
- **Companies may be reluctant to reveal financial information**

MAR: MISSING AT RANDOM

Missing at Random (MAR): probability of missing data is dependent on the observed data but not the unobserved data

Suppose that X is the observed data and Y is the unobserved data. Call our “missing matrix” R

Then, if the data are MAR, P(R|X,Y) = ??????????

$$P(R|X,Y) = P(R|X)$$

Not exactly random (in the vernacular sense).

- There is a probabilistic mechanism that is associated with whether the data is missing
- Mechanism takes the observed data as input

EXAMPLES?

?

MAR: KEY POINT

We can model that latent mechanism and compensate for it

Imputation: replacing missing data with substituted values

- Models today will assume MAR

Example: if age is known, you can model missing-ness as a function of age

Whether or not missing data is MAR or the next type, Missing Not at Random (MNAR), is not* testable.

- Requires you to “understand” your data

*unless you can get the missing data (e.g., post-study phone calls)

MNAR: MISSING NOT AT RANDOM

MNAR: missing-ness has something to do with the missing data itself

Examples: ??????????

- Do you binge drink? Do you have a trust fund? Do you use illegal drugs? What is your sexuality? Are you depressed?

Said to be "non-ignorable":

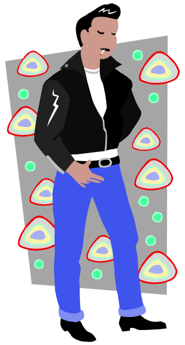
- Missing data mechanism must be considered as you deal with the missing data
- Must include model for why the data are missing, and best guesses as to what the data might be

BACK TO IRIBE …

Is the the missing data:

- **MCAR;**
- **MAR; or**
- **MNAR?**

???????????



Hair Color	> 6ft	Grade
Red	Y	A
Brown	N	A
Black	N	B
Black	Y	A
Brown	Y	
Brown	Y	
Brown	N	
Black	Y	B
Black	Y	B
Brown	N	A
Black	N	
Brown	N	C
Red	Y	
Red	N	A
Brown	Y	A
Black	Y	A

ADD A VARIABLE

Bring in the GPA:

Does this change anything?

Hair Color	GPA	> 6ft	Grade
Red	3.4	Y	A
Brown	3.6	N	A
Black	3.7	N	B
Black	3.9	Y	A
Brown	2.5	Y	
Brown	3.2	Y	
Brown	3.0	N	
Black	2.9	Y	B
Black	3.3	Y	B
Brown	4.0	N	A
Black	3.65	N	
Brown	3.4	N	C
Red	2.2	Y	
Red	3.8	N	A
Brown	3.8	Y	A
Black	3.67	Y	A

HANDLING MISSING DATA …

SINGLE IMPUTATION

Mean imputation: imputing the average from observed cases for all missing values of a variable

Hot-deck imputation: imputing a value from another subject, or “donor,” that is most like the subject in terms of observed variables

- Last observation carried forward (LOCF): order the dataset somehow and then fill in a missing value with its neighbor

Cold-deck imputation: bring in other datasets

Old and busted:

- All fundamentally impose too much precision.
- Have uncertainty over what unobserved values actually are
- Developed before cheap computation

MULTIPLE IMPUTATION

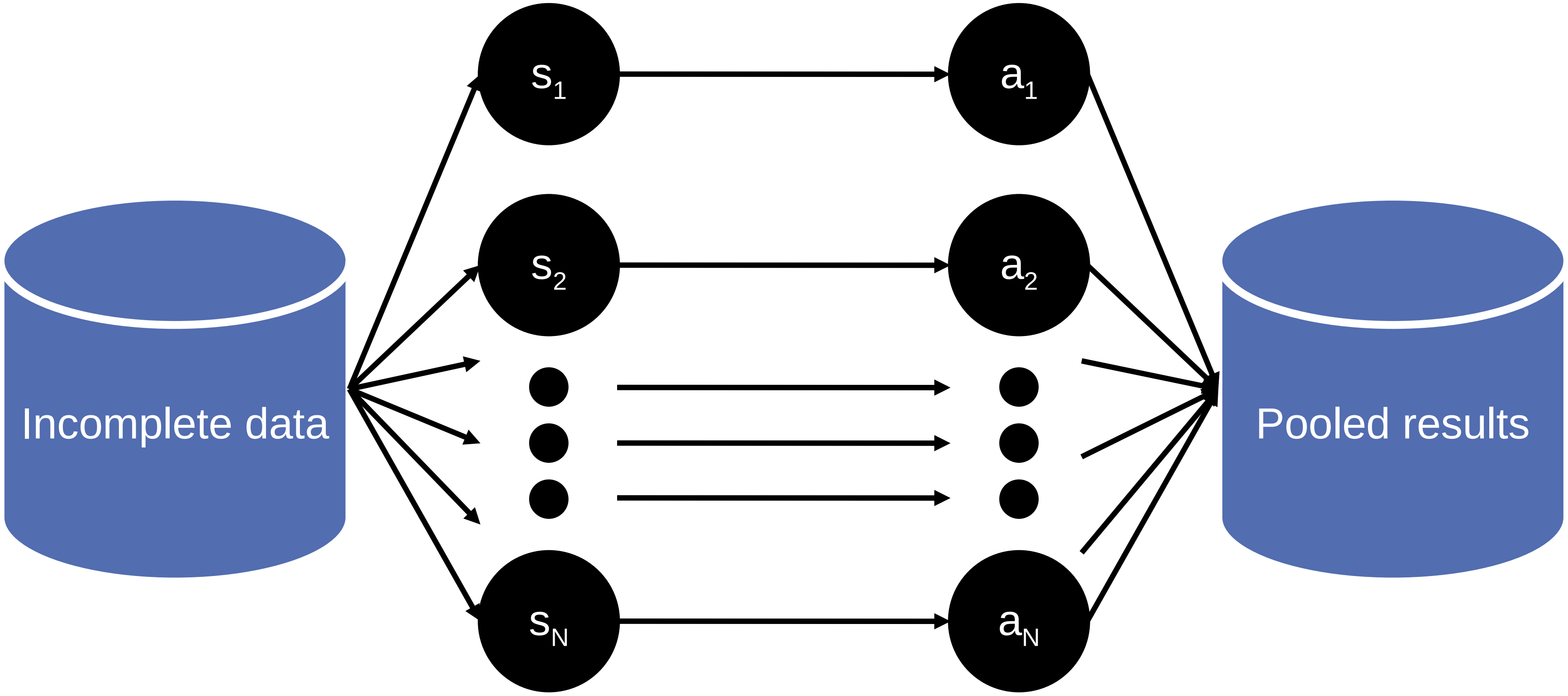
Developed to deal with noise during imputation

- Impute once   treats imputed value as observed

We have uncertainty over what the observed value would have been

Multiple imputation: generate several random values for each missing data point during imputation

IMPUTATION PROCESS



Impute N times

Analysis performed on each imputed set

TINY EXAMPLE

X	Y
32	2
43	?
56	6
25	?
84	5

Independent variable: X
Dependent variable: Y
We assume Y has a linear relationship with X

LET'S IMPUTE SOME DATA!

Use a predictive distribution of the missing values:

- Given the observed values, make random draws of the observed values and fill them in.
- Do this N times and make N imputed datasets

X	Y
32	2
43	5.5
56	6
25	8
84	5

X	Y
32	2
43	7.2
56	6
25	1.1
84	5

For very large values of N=2 …

INFERENCE WITH MULTIPLE IMPUTATION

Now that we have our imputed data sets, how do we make use of them?   ???????????

- Analyze each of the separately

X	Y
32	2
43	5.5
56	6
25	8
84	5

X	Y
32	2
43	7.2
56	6
25	1.1
84	5

Slope	-0.8245
Standard error	6.1845

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$$

Slope	4.932
Standard error	4.287

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$$

POOLING ANALYSES

Pooled slope estimate is the average of the N imputed estimates

Our example, β_{1p} = = (4.932-.8245) x 0.5 = 2.0538

The pooled slope variance is given by

) x β_{1p})^2

Where Z_i is the standard error of the imputed slopes

Our example: (4.287 + 6.1845)/2 + (3/2)*(16.569) = 30.08925

Standard error: take the square root, and we get 5.485

PREDICTING THE MISSING DATA GIVEN THE OBSERVED DATA

Given events A, B; and P(A) > 0 …

Bayes’ Theorem:

In our case:

Probability of seeing evidence given the hypothesis

Prior probability of hypotheses

Posterior probability of the hypothesis given the evidence

Prior over the evidence

BAYESIAN IMPUTATION

Establish a prior distribution:

- **Some distribution of parameters of interest θ before considering the data, *$P(\theta)$***
- **We want to estimate θ**

Given θ, can establish a distribution *$P(X_{obs}|\theta)$*

Use Bayes Theorem to establish *$P(\theta|X_{obs})$* …

- **Make random draws for θ**
- **Use these draws to make predictions of Y_{miss}**

HOW BIG SHOULD N BE?

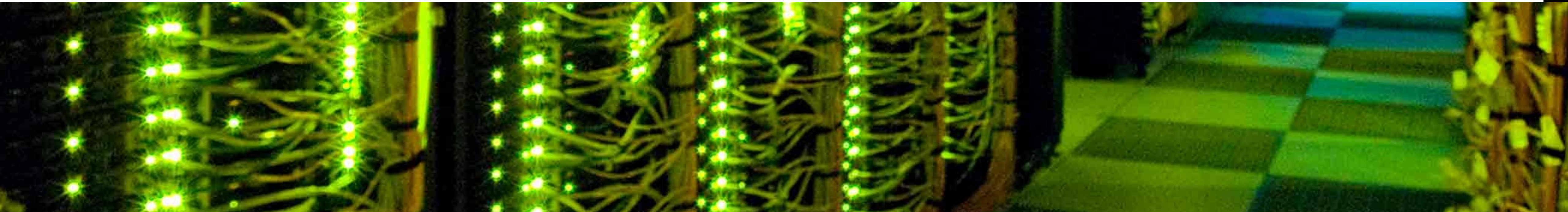
Number of imputations N depends on:

- Size of dataset
- Amount of missing data in the dataset

Some research finds small N is sufficient for efficiency of the estimates, based on:

- (1 +)-1 N is the number of imputations and λ is the fraction of missing information for the term being estimated [Schaffer 1999]

More recent research claims that a good N is actually higher in order to achieve higher power [Graham et al. 2007] – computation is cheap now, too!

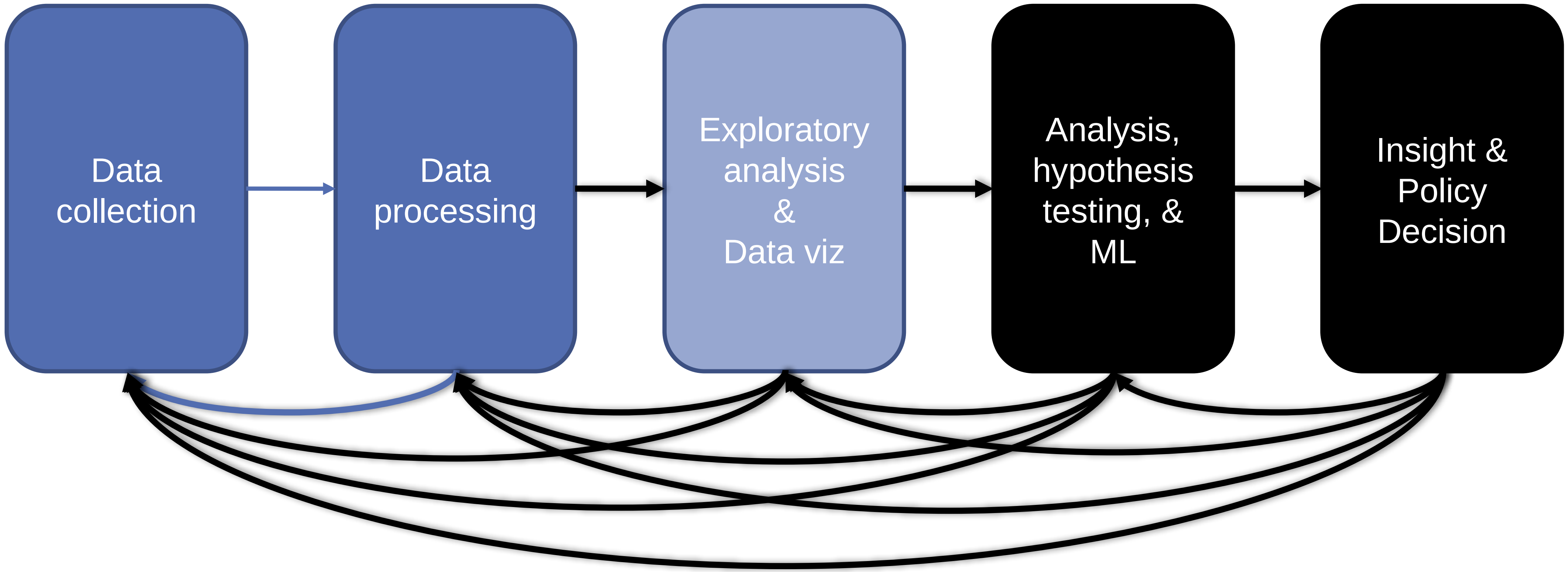


MORE ADVANCED METHODS

Interested? Further reading:

- Regression-based MI methods
- Multiple Imputation Chained Equations (MICE) or Fully Conditional Specification (FCS)
  - Readable summary from JHU School of Public Health: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3074241/
- Markov Chain Monte Carlo (MCMC)
  - We’ll cover this a bit, but also check out CMSC422!

REST OF TODAY'S LECTURE


Data collection
Data processing
Exploratory analysis & Data viz
Analysis, hypothesis testing, & ML
Insight & Policy Decision

Continue with the general topic of data wrangling and cleaning & EDA intersection

Many slides from Amol Deshpande (UMD)

OVERVIEW

Goal: get data into a structured form suitable for analysis

- Variously called: data preparation, data munging, data curation
- Also often called ETL (Extract-Transform-Load) process

Often the step where majority of time (80-90%) is spent

Key steps:

- Scraping: extracting information from sources, e.g., webpages, spreadsheets
- Data transformation: to get it into the right structure
- Data integration: combine information from multiple sources
- Information extraction: extracting structured information from unstructured/text sources
- Data cleaning: remove inconsistencies/errors

OVERVIEW

Goal: get data into a structured form suitable for analysis

- Variously called: data preparation, data munging, data curation
- Also often called ETL (Extract-Transform-Load) process

Often the step where majority of time (80-90%) is spent

Key steps:

- Scraping: extracting information from sources, e.g., webpages, spreadsheets
- Data transformation: to get it into the right structure
- Information extraction: extracting structured information from unstructured/text sources
- **Data integration: combine information from multiple sources**
- **Data cleaning: remove inconsistencies/errors**

Already covered

In a few classes

OVERVIEW

Many of the problems are not easy to formalize, and have seen little work

- E.g., Cleaning
- Others aspects of integration, e.g., schema mapping, have been studied in depth

A mish-mash of tools typically used

- Visual (e.g., Trifacta), or not (UNIX grep/sed/awk, Pandas)
- Ad hoc programs for cleaning data, depending on the exact type of errors
- Different types of transformation tools
- Visualization and exploratory data analysis to understand and remove outliers/noise
- Several tools for setting up the actual pipelines, assuming the individual steps are setup (e.g., Talend, AWS Glue)

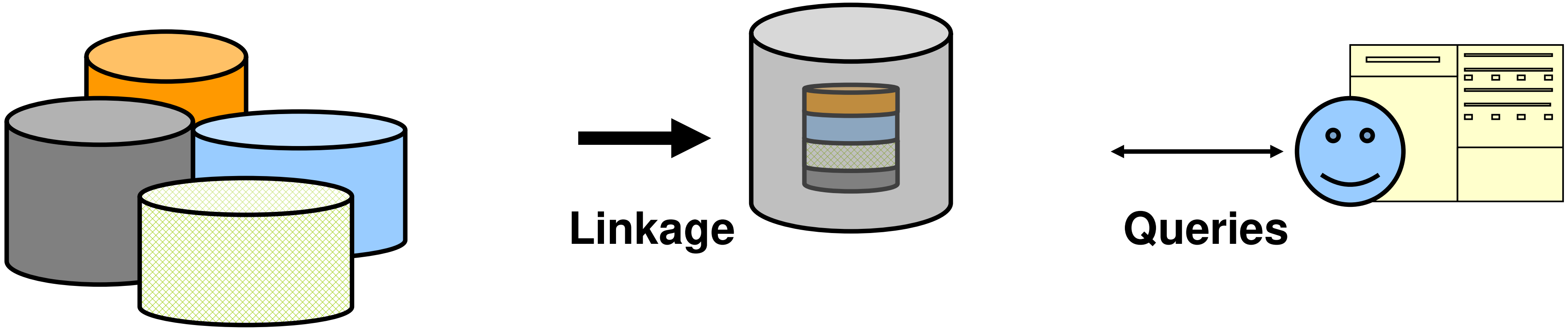
OUTLINE

- **Data Integration**
- **Data Quality Issues**
- **Data Cleaning**
- **Entity Resolution**

OUTLINE

- **Data Integration**
- **Data Quality Issues**
- **Data Cleaning**
- **Entity Resolution**

DATA INTEGRATION



Linkage

Queries

- **Discovering** information sources (e.g. deep web modeling, schema learning, …)
- **Gathering** data (e.g., wrapper learning & information extraction, federated search, …)

- **Cleaning** data (e.g., de-duping and **linking records**) to form a single [virtual] database

- **Querying** integrated information sources (e.g. queries to views, execution of web-based queries, …)
- **Data mining & analyzing** integrated information (e.g., collaborative filtering/classification learning using extracted data, …)

DATA INTEGRATION

Goal: Combine data residing in different sources and provide users with a unified view of these data for querying or analysis

- Each data source has its own schema called **local schemas** (much work assumes relational schemas, but some work on XML as well)
- The unified schema is often called **mediated schema** or **global schema**

Two different setups:

1. Bring the data together into a single repository (often called data warehousing)
2. Keep the data where it is, and send queries back and forth

1. DATA WAREHOUSING

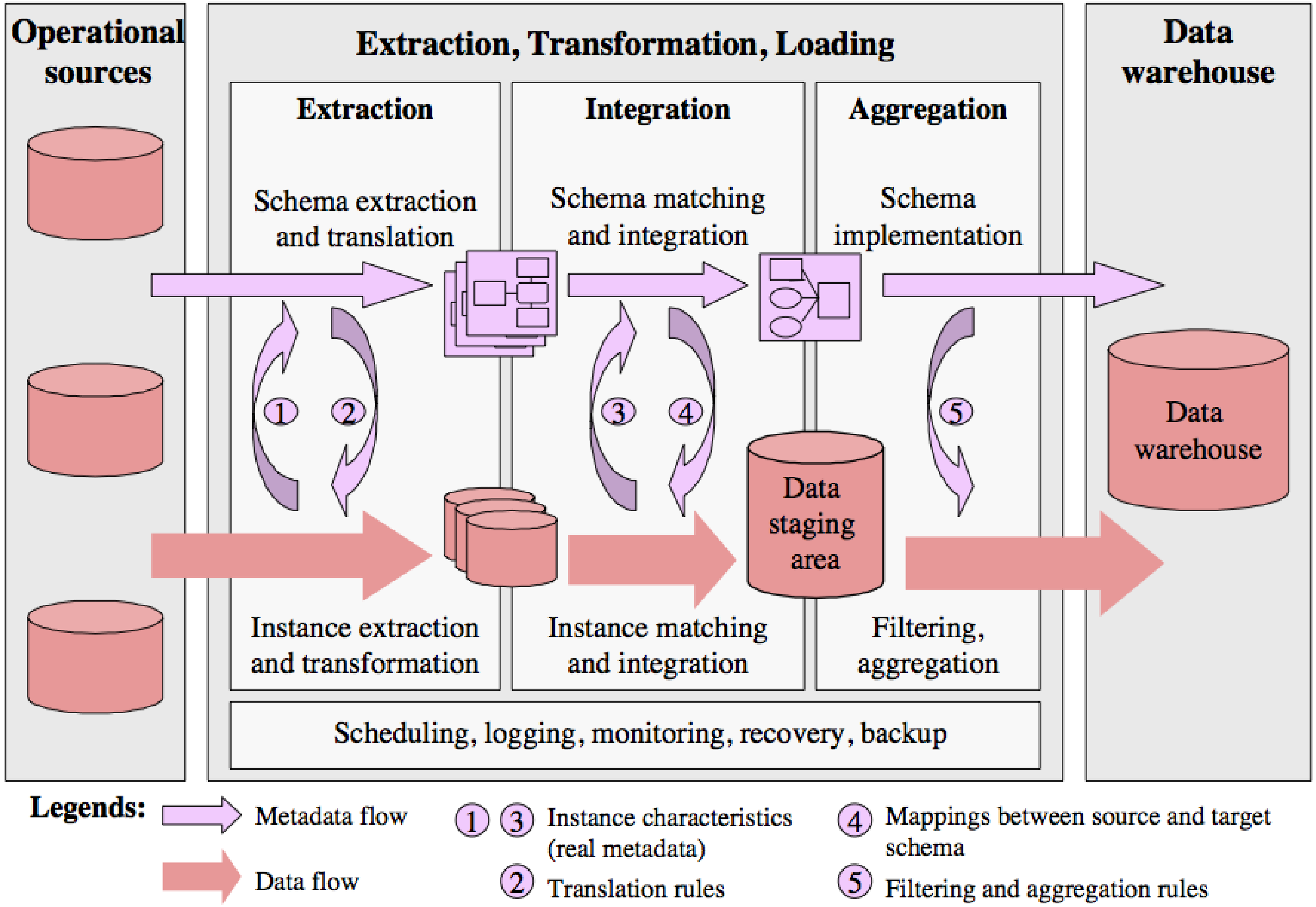
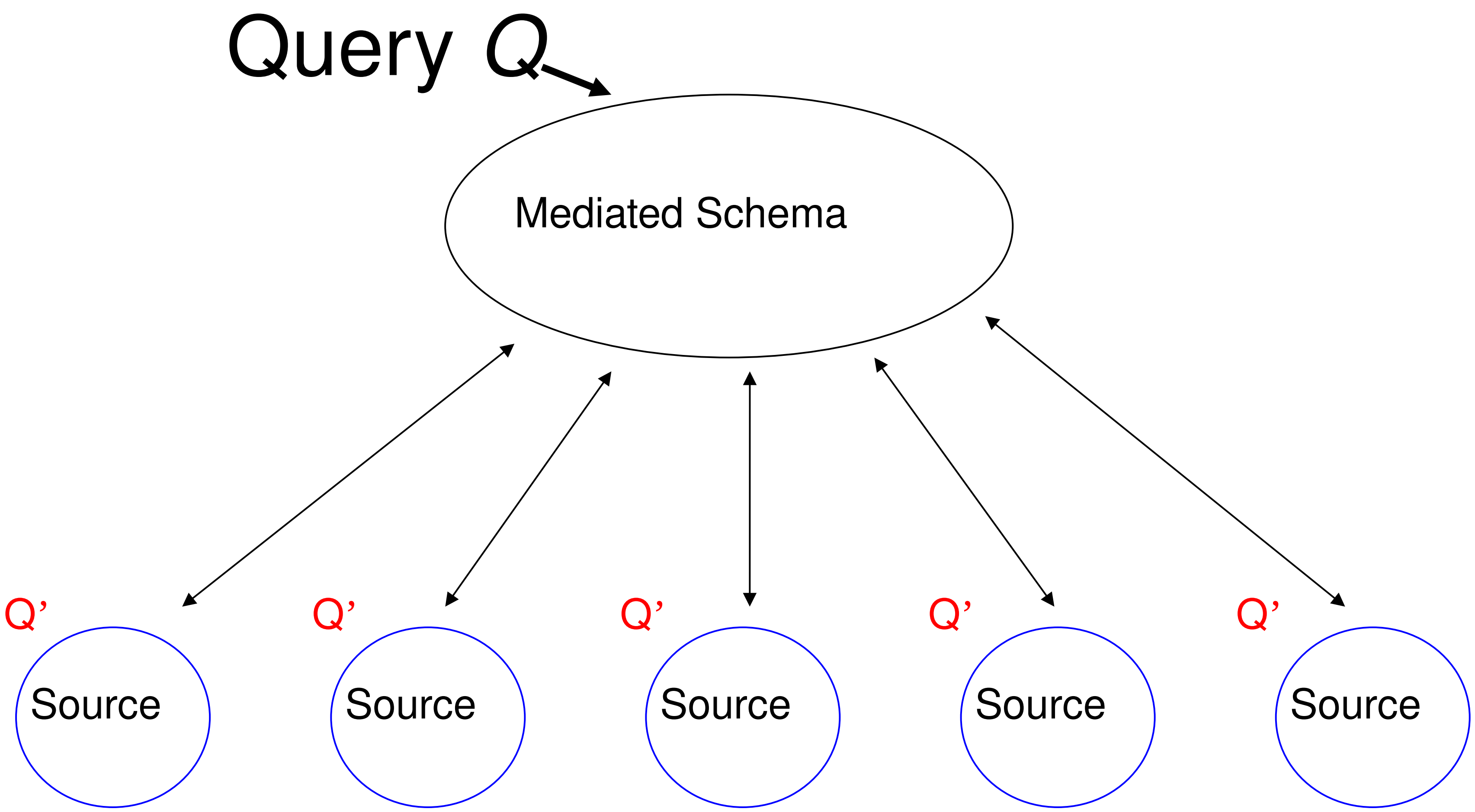

Operational sources
Extraction, Transformation, Loading
Data warehouse
Extraction
Integration
Aggregation
Schema extraction and translation
Schema matching and integration
Schema implementation
1
2
3
4
5
Data staging area
Data warehouse
Instance extraction and transformation
Instance matching and integration
Filtering, aggregation
Scheduling, logging, monitoring, recovery, backup
Legends:
Metadata flow
Data flow
1 3 Instance characteristics (real metadata)
2 Translation rules
4 Mappings between source and target schema
5 Filtering and aggregation rules

Figure 1. Steps of building a data warehouse: the ETL process

From
Data Cleaning: Problems
d Current Approaches

2. IN-PLACE INTEGRATION



DATA INTEGRATION

Two different setups:

1. Bring the data together into a single repository (often called data warehousing)
   - Relatively easier problem - only need one-way-mappings
   - Query performance predictable and under your control

2. Keep the data where it is, and send queries back and forth
   - Need two-way mappings -- a query on the mediated schema needs to be translated into queries over data source schemas
   - Not as efficient and clean as data warehousing, but a better fit for dynamic data
   - Or when data warehousing is not feasible

DATA INTEGRATION: KEY CHALLENGES

Data extraction, reconciliation, and cleaning

- Get the data from each source in a structured form
- Often need to use wrappers to extract data from web sources
- May need to define a schema

Schema alignment and mapping

- Decide on the best mediated schema
- Figure out mappings and matchings between the local schemas and the global schema

Answer queries over the global schema

- In the second scenario, need to figure out how to map a query on global schema onto queries over local schemas
- Also need to decide which sources contain relevant data

Limitations in mechanisms for accessing sources

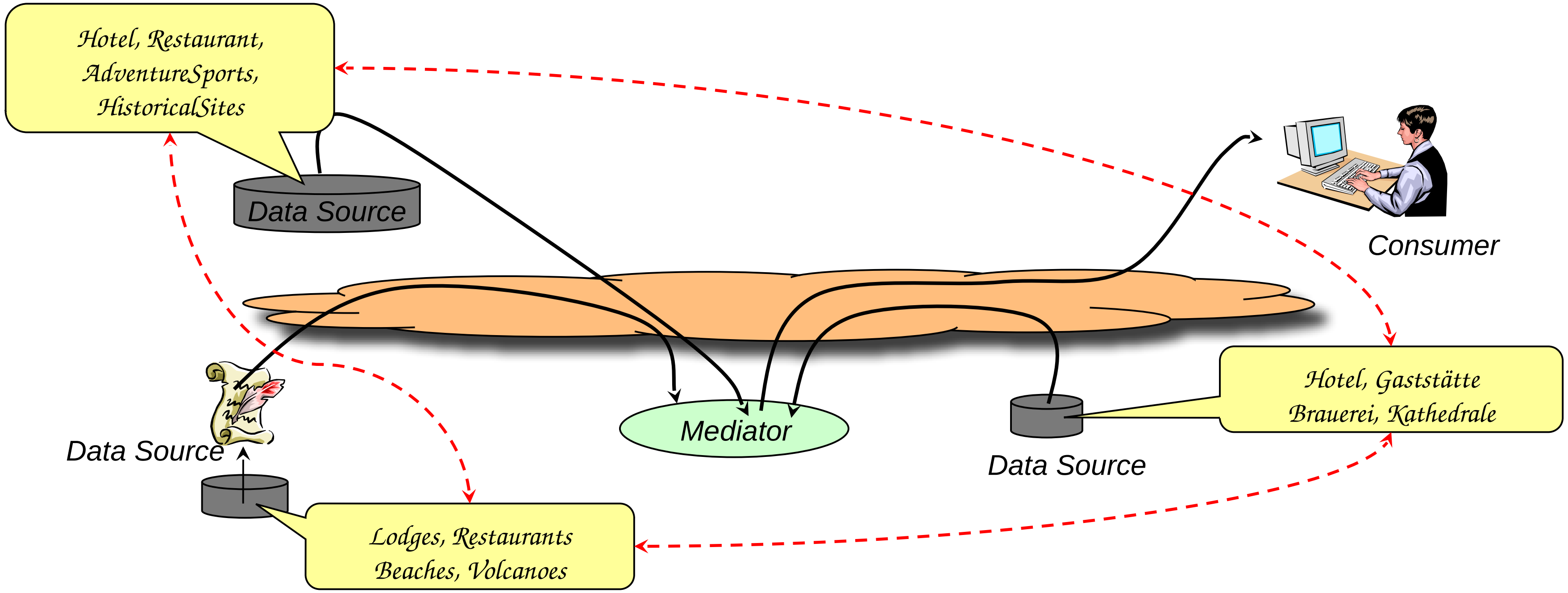
- Many sources have limits on how you can access them
- Limits on the number of queries you can issues (say 100 per min)
- Limits on the types of queries (e.g., must enter a zipcode to get information from a web source)

Flashbacks to Project 1 …

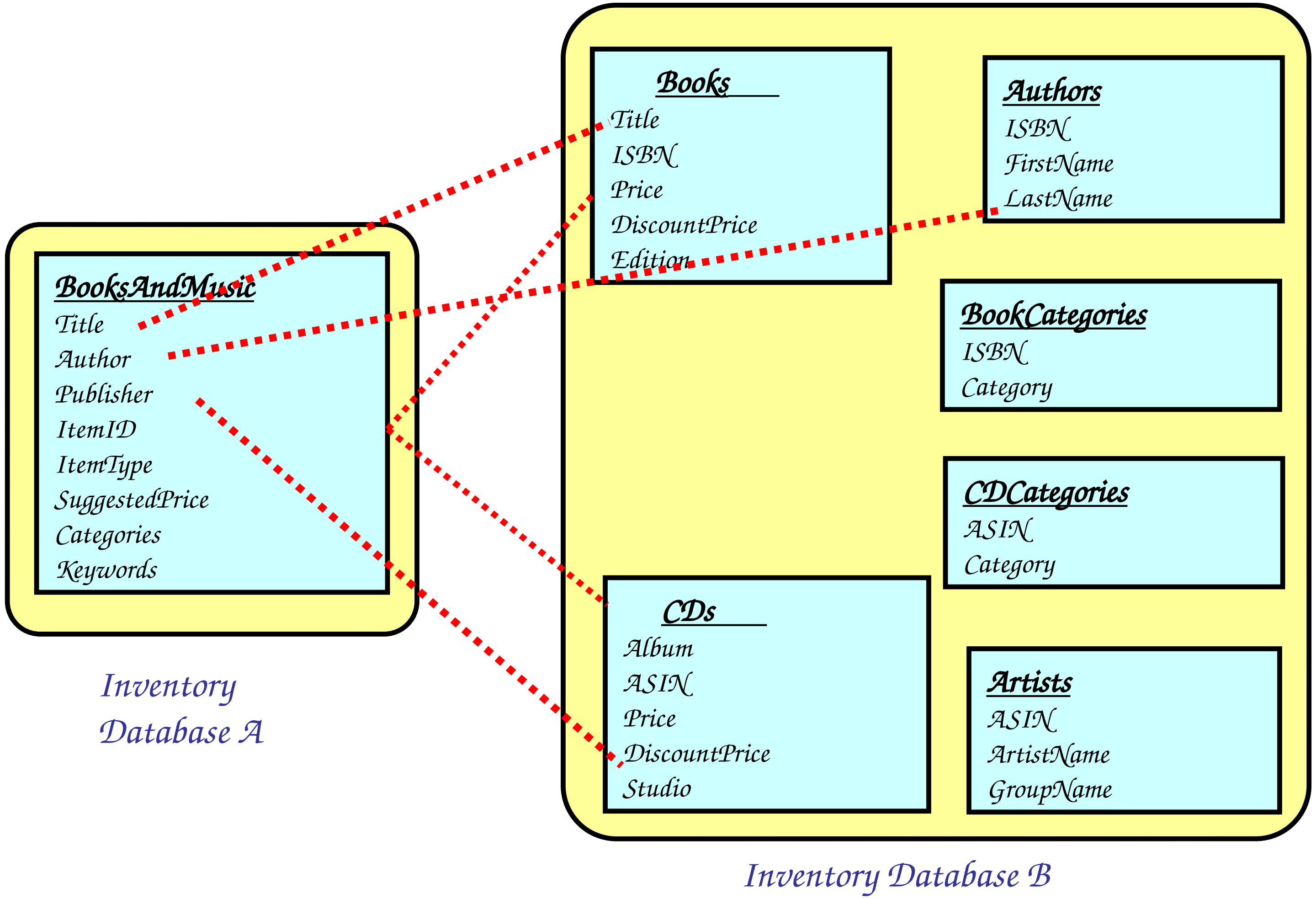
SCHEMA MATCHING OR ALIGNMENT

Goal: Identify corresponding elements in two schemas

- As a first step toward constructing a global schema
- Schema heterogeneity is a key roadblock
    - Different data sources speak their own schema


Hotel, Restaurant, AdventureSports, HistoricalSites
Data Source
Consumer
Data Source
Mediator
Data Source
Hotel, Gaststätte Brauerei, Kathedrale
Lodges, Restaurants Beaches, Volcanoes

SCHEMA MATCHING OR ALIGNMENT



SUMMARY

- **Data integration continues to be a very active area in research and increasingly industry**
- **Solutions still somewhat ad hoc and manual, although tools beginning to emerge**
- **Need to minimize the time needed to integrate a new data source**
    - Crucial opportunities may be lost otherwise
    - Can take weeks to do it properly
- **Dealing with changes to the data sources a major headache**
    - Especially for data sources not under your control

OUTLINE

- **Data Integration**
- **Data Quality Issues**
- **Data Cleaning**
- **Entity Resolution**

DATA QUALITY PROBLEMS

Different sources are developed separately, and maintained by different people

Issue 1: Mapping information across sources (schema mapping/transformation)

- Naming conflicts: same name used for different objects
- Structural conflicts: different representations across sources
- We will cover this later

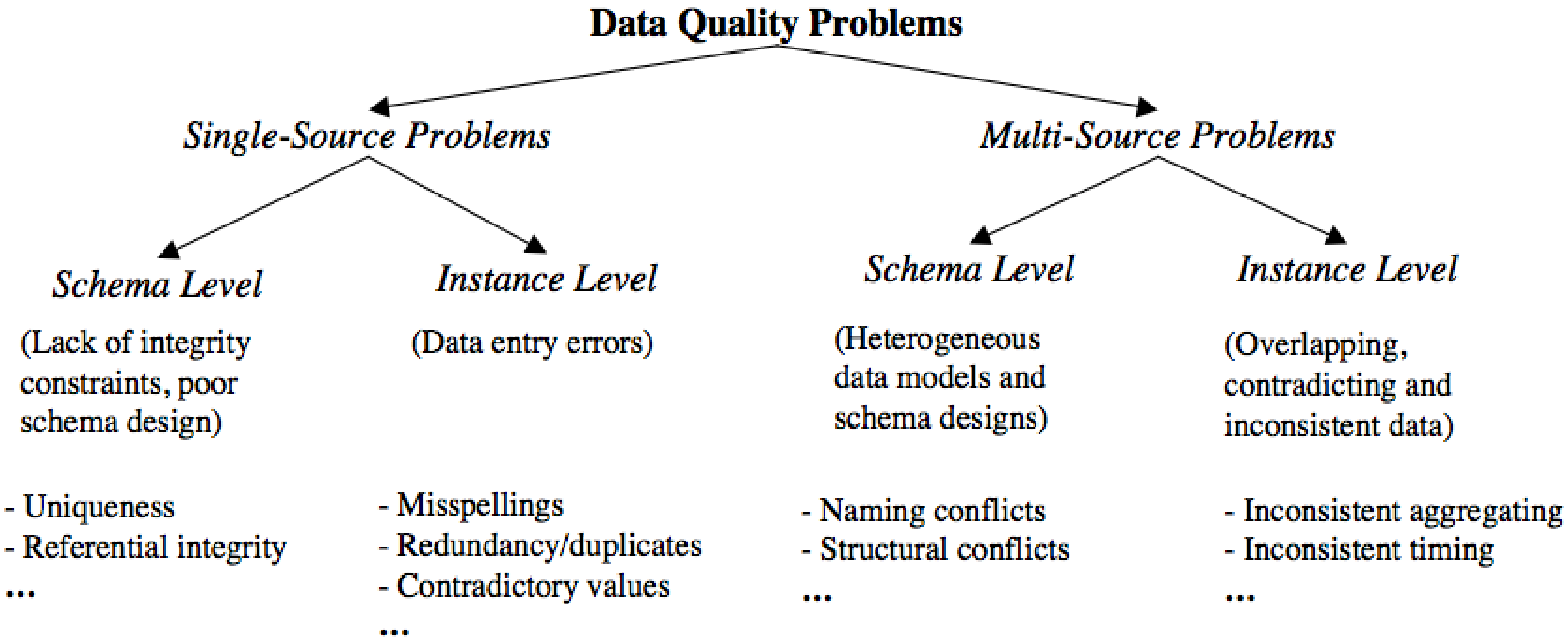
Issue 2: Entity Resolution: Matching entities across sources

Issue 3: Data quality issues

- Contradicting information, Mismatched information, etc.

From Data Cleaning: Problems and Current Approaches

A TAXONOMY OF DATA QUALITY PROBLEMS



Data Quality Problems

Single-Source Problems

Schema Level

(Lack of integrity constraints, poor schema design)

- Uniqueness
- Referential integrity

...

Instance Level

(Data entry errors)

- Misspellings
- Redundancy/duplicates
- Contradictory values

...

Multi-Source Problems

Schema Level

(Heterogeneous data models and schema designs)

- Naming conflicts
- Structural conflicts

...

Instance Level

(Overlapping, contradicting and inconsistent data)

- Inconsistent aggregating
- Inconsistent timing

...

Figure 2. Classification of data quality problems in data sources

SINGLE-SOURCE PROBLEMS

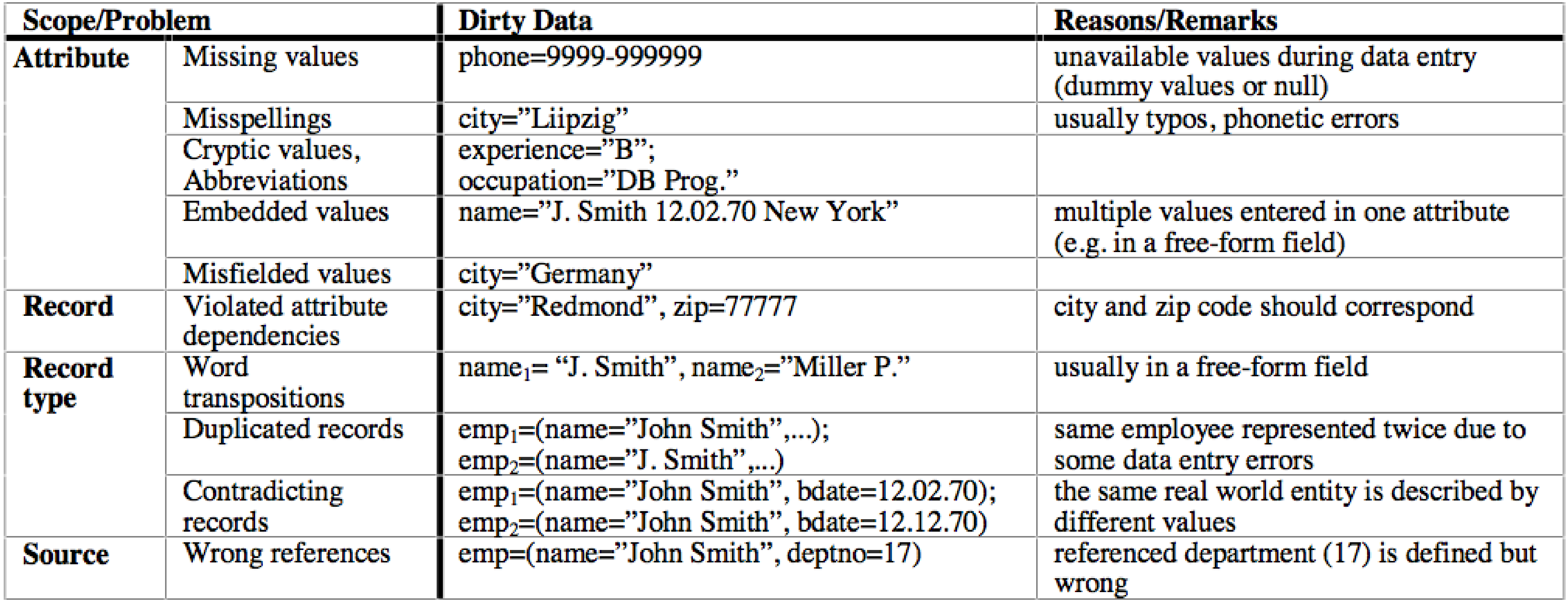
Depends largely on the source

Databases can enforce constraints, whereas data extracted from files or spreadsheets, or scraped from webpages is much more messy

Types of problems:

- Ill-formatted data, especially from webpages or files or spreadsheets
- Missing or illegal values, Misspellings, Use of wrong fields, Extraction issues (not easy to separate out different fields)
- Duplicated records, Contradicting Information, Referential Integrity Violations
- Unclear default values (e.g., data entry software needs something)
- Evolving schemas or classification schemes (for categorical attributes)
- Outliers

DATA QUALITY PROBLEMS



Scope/Problem		Dirty Data	Reasons/Remarks
Attribute	Missing values	phone=9999-999999	unavailable values during data entry (dummy values or null)
	Misspellings	city=”Liipzig”	usually typos, phonetic errors

	Cryptic values, Abbreviations	experience=”B”;
		occupation=”DB Prog.”	

	Embedded values	name=”J. Smith 12.02.70 New York”	multiple values entered in one attribute (e.g. in a free-form field)
	Misfielded values	city=”Germany”	
Record	Violated attribute dependencies	city=”Redmond”, zip=77777	city and zip code should correspond
Record type	Word transpositions	name$_1$= “J. Smith”, name$_2$=”Miller P.”	usually in a free-form field

	Duplicated records	emp$_1$=(name=”John Smith”,...);
		emp$_2$=(name=”J. Smith”,...)
			same employee represented twice due to some data entry errors

	Contradicting records	emp$_1$=(name=”John Smith”, bdate=12.02.70);
		emp$_2$=(name=”John Smith”, bdate=12.12.70)
			the same real world entity is described by different values

Source	Wrong references	emp=(name=”John Smith”, deptno=17)	referenced department (17) is defined but wrong

Table 2. Examples for single-source problems at instance level

MULTI-SOURCE PROBLEMS

Different sources are developed separately, and maintained by different people

Issue 1: Mapping information across sources (schema mapping/transformation)

- Naming conflicts: same name used for different objects
- Structural conflicts: different representations across sources
- We will cover this later

Issue 2: Entity Resolution: Matching entities across sources

Issue 3: Data quality issues

- Contradicting information, Mismatched information, etc.

OUTLINE

- **Data Integration**
- **Data Quality Issues**
- **Data Cleaning**
  - Outlier Detection
  - Constraint-based Cleaning
  - Entity Resolution

UNIVARIATE OUTLIERS

A set of values can be characterized by metrics such as center (e.g., mean), dispersion (e.g., standard deviation), and skew

Can be used to identify outliers

- Must watch out for "masking": one extreme outlier may alter the metrics sufficiently to mask other outliers
- Should use **robust statistics**: considers effect of corrupted data values on distributions (recall median vs mean, …)
- Robust center metrics: median, k% trimmed mean (discard lowest and highest k% values)
- Robust dispersion:
    - Median Absolute Deviation (MAD): median distance of all the values from the median value

A reasonable approach to find outliers: any data points 1.4826x MAD away from median

- The above assumes that data follows a **normal** distribution
- May need to eyeball the data (e.g., plot a histogram) to decide if this is true

UNIVARIATE OUTLIERS

Wikipedia Article on Outliers lists several other normality-based tests for outliers

If data appears to be not normally distributed:

- Distance-based methods: look for data points that do not have many neighbors
- Density-based methods:
    - Define *density* to be average distance to *k* nearest neighbors
    - *Relative density* = density of node/average density of its neighbors
    - Use relative density to decide if a node is an outlier

Most of these techniques start breaking down as the dimensionality of the data increases

- *Curse of dimensionality*
- Can project data into lower-dimensional space and look for outliers there
    - Not as straightforward

OTHER OUTLIERS

Timeseries outliers

- Often the data is in the form of a timeseries
- Can use the historical values/patterns in the data to flag outliers
- Rich literature on *forecasting* in timeseries data

Frequency-based outliers

- An item is considered a "heavy hitter" if it is much more frequent than other items
- In relational tables, can be found using a simple *groupby-count*
- Often the volume of data may be too much (e.g., internet routers)
    - Approximation techniques often used
    - To be discussed sometime later in the class

Things generally not as straightforward with other types of data

- Outlier detection continues to be a major research area

WRAP-UP

Data wrangling/cleaning are a key component of data science pipeline

Still largely ad hoc although much tooling in recent years

Specifically, we covered:

- Schema mapping and matching
- Outliers

Next up:

- Constraint-based Cleaning
- Entity Resolution/Record Linkage/Data Matching

DATA CLEANING: OUTLIER RESOLUTION

From: **Entity Resolution Tutorial**

Identify different manifestations of the same real world object

- Also called: identity reconciliation, record linkage, deduplication, fuzzy matching, Object consolidation, Coreference resolution, and several others

Motivating examples: ?????????????

- Postal addresses
- Entity recognition in NLP/Information Extraction
- Identifying companies in financial records
- Comparison shopping
- Author disambiguation in citation data
- Connecting up accounts on online networks
- Crime/Fraud Detection
- Census
- ...

DATA CLEANING: OUTLIER RESOLUTION

Important to correctly identify references

- Often actions taken based on extracted data
- Cleaning up data by entity resolution can show structure that may not be apparent before

Challenges

- Such data is naturally ambiguous (e.g., names, postal addresses)
- Abbreviations/data truncation
- Data entry errors, Missing values, Data formatting issues complicate the problem
- Heterogeneous data from many diverse sources

No magic bullet here !!

- Approaches fairly domain-specific
- Be prepared to do a fair amount of manual work

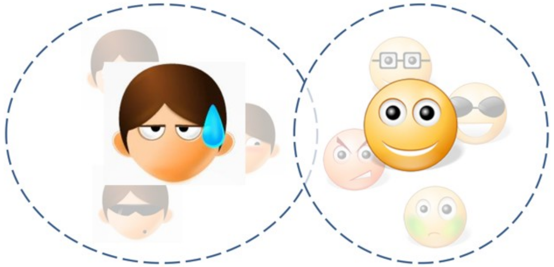
ENTITY RESOLUTION: THREE SLIGHTLY DIFFERENT PROBLEMS

Setup:

- Real world: there are entities (people, addresses, businesses)
- We have a large collection of noisy, ambiguous "references" to those entities (also called "mentions")
- Somewhat different techniques, but a lot of similarities

Deduplication

- Cluster records/mentions that correspond to the same entity
- Choose/construct a cluster representative
  - This is in itself a non-trivial task (e.g., averaging may work for numerical attributes, but what about string attributes?)



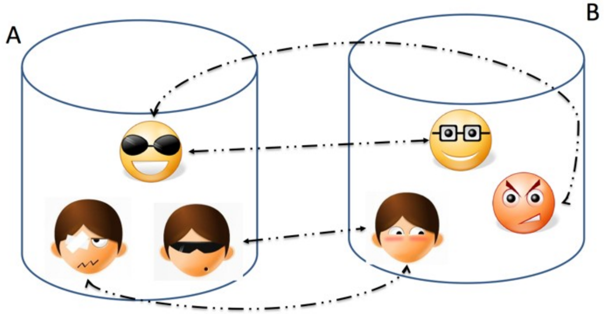
ENTITY RESOLUTION: THREE SLIGHTLY DIFFERENT PROBLEMS

Setup:

- Real world: there are entities (people, addresses, businesses)
- We have a large collection of noisy, ambiguous "references" to those entities (also called "mentions")
- Somewhat different techniques, but a lot of similarities

Record Linkage

- Match records across two different databases (e.g., two social networks, or financial records w/ campaign donations)
- Typically assume that the two databases are fairly clean



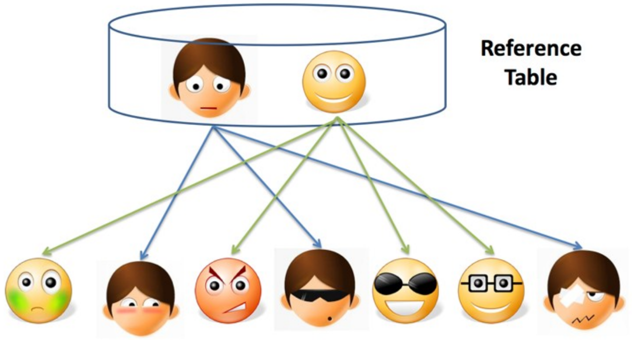
ENTITY RESOLUTION: THREE SLIGHTLY DIFFERENT PROBLEMS

Setup:

- Real world: there are entities (people, addresses, businesses)
- We have a large collection of noisy, ambiguous "references" to those entities (also called "mentions")
- Somewhat different techniques, but a lot of similarities

Reference Matching

- Match "references" to clean records in a reference table
- Commonly comes up in "entity recognition" (e.g., matching newspaper article mentions to names of people)



ENTITY RESOLUTION: DATA MATCHING

Comprehensive treatment: Data Matching; P. Christen; 2012 (Springer Books -- not available for free)

One of the key issues is finding similarities between two references

- What similarity function to use?

Edit Distance Functions

- Levenstein: min number of changes to go from one reference to another
  - A change is defined to be: a single character insertion or deletion or substitution
  - May add transposition
- Many adjustments to the basic idea proposed (e.g., higher weights to changes at the start)
- Not cheap to compute, especially for millions of pairs

Set Similarity

- Some function of intersection size and union size
- E.g., Jaccard distance = size of intersection/size of union
- Much faster to compute

Vector Similarity

- Cosine similarity – we'll talk about this much more in NLP lectures

ENTITY RESOLUTION: DATA MATCHING

Q-Grams

- Find all length-q substrings in each string
- Use set/vector similarity on the resulting set

Several approaches that combine the above (especially q-grams and edit distance, e.g., Jaro-Winkler)

<u>Soundex</u>: Phonetic Similarity Metric

- Homophones should be encoded to the same representation so spelling errors can be handled
- Robert and Rupert get assigned the same code (R163), but Rubin yields R150

May need to use Translation Tables

- To handle abbreviations, nicknames, other synonyms

Different types of data requires more domain-specific functions

- E.g., geographical locations, postal addresses
- Also much work on computing distances between XML documents etc.

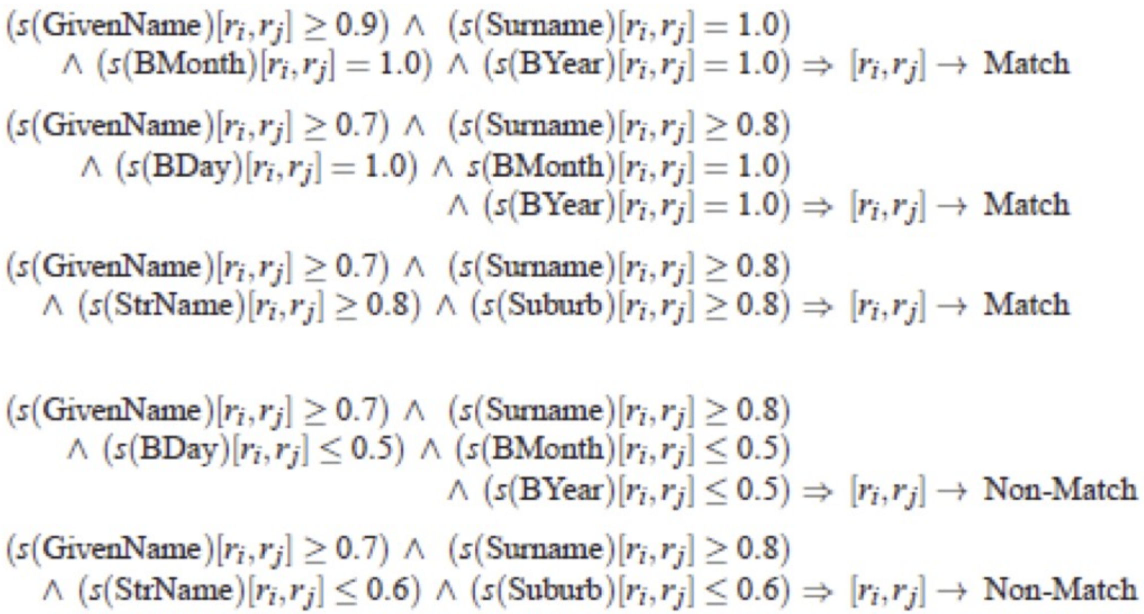
ENTITY RESOLUTION: ALGORITHMS

Simple threshold method

- If the distance below some number, the two references are assumed to be equal
- May review borderline matches manually

Can be generalized to rule-based:

- Example from Christen, 2012



$$(s(\text{GivenName})[r_i,r_j] \geq 0.9) \wedge (s(\text{Surname})[r_i,r_j] = 1.0) \wedge (s(\text{BMonth})[r_i,r_j] = 1.0) \wedge (s(\text{BYear})[r_i,r_j] = 1.0) \Rightarrow [r_i,r_j] \rightarrow \text{Match}$$

$$(s(\text{GivenName})[r_i,r_j] \geq 0.7) \wedge (s(\text{Surname})[r_i,r_j] \geq 0.8) \wedge (s(\text{BDay})[r_i,r_j] = 1.0) \wedge s(\text{BMonth})[r_i,r_j] = 1.0) \wedge (s(\text{BYear})[r_i,r_j] = 1.0) \Rightarrow [r_i,r_j] \rightarrow \text{Match}$$

$$(s(\text{GivenName})[r_i,r_j] \geq 0.7) \wedge (s(\text{Surname})[r_i,r_j] \geq 0.8) \wedge (s(\text{StrName})[r_i,r_j] \geq 0.8) \wedge (s(\text{Suburb})[r_i,r_j] \geq 0.8) \Rightarrow [r_i,r_j] \rightarrow \text{Match}$$

$$(s(\text{GivenName})[r_i,r_j] \geq 0.7) \wedge (s(\text{Surname})[r_i,r_j] \geq 0.8) \wedge (s(\text{BDay})[r_i,r_j] \leq 0.5) \wedge (s(\text{BMonth})[r_i,r_j] \leq 0.5) \wedge (s(\text{BYear})[r_i,r_j] \leq 0.5) \Rightarrow [r_i,r_j] \rightarrow \text{Non-Match}$$

$$(s(\text{GivenName})[r_i,r_j] \geq 0.7) \wedge (s(\text{Surname})[r_i,r_j] \geq 0.8) \wedge (s(\text{StrName})[r_i,r_j] \leq 0.6) \wedge (s(\text{Suburb})[r_i,r_j] \leq 0.6) \Rightarrow [r_i,r_j] \rightarrow \text{Non-Match}$$

ENTITY RESOLUTION: ALGORITHMS

May want to give more weight to matches involving rarer words

- More naturally applicable to record linkage problem
- If two records match on a rare name like "Machanavajjhala", they are likely to be a match
- Can formalize this as "probabilistic record linkage”

Constraints: May need to be satisfied, but can also be used to find matches

- Often have constraints on the matching possibilities
- Transitivity: M1 and M2 match, and M2 and M3 match, and M1 and M3 must match
- Exclusivity: M1 and M2 match --> M3 cannot match with M2
- Other types of constraints:
    - E.g., if two papers match, their venues must match

ENTITY RESOLUTION: ALGORITHMS

Clustering-based ER Techniques:

- Deduplication is basically a clustering problem
- Can use clustering algorithms for this purpose
- But most clusters are very small (in fact of size = 1)
- Some clustering algorithms are better suited for this, especially Agglomerative Clustering
    - Unlikely K-Means would work here

ENTITY RESOLUTION: ALGORITHMS

Crowdsourcing

- Humans are often better at this task
- Can use one of the crowdsourcing mechanisms (e.g., Mechanical Turk) for getting human input on the difficult pairs
- Quite heavily used commercially (e.g., to disambiguate products, restaurants, etc.)

ENTITY RESOLUTION: SCALING TO BIG DATA

One immediate problem

- There are $O(N^2)$ possible matches
- Must reduce the search space

Use some easy-to-evaluate criterion to restrict the pairs considered further

- May lead to false negative (i.e., missed matches) depending on how noisy the data is

Much work on this problem as well, but domain-specific knowledge likely to be more useful in practice

One useful technique to know: min-hash signatures

- Can quickly find potentially overlapping sets
- Turns up to be very useful in many domains (beyond ER)