

MSiA-413 Introduction to Databases and Information Retrieval

Lecture 19 Semi-structured data, Cleaning messy data

Instructor: Nikos Hardavellas

Slides adapted from Steve Tarzia, John Canny, Michael Franklin,
Dan Bruckner, Evan Sparks, Shivaram Venkataraman

Last Lecture

- Partitioning

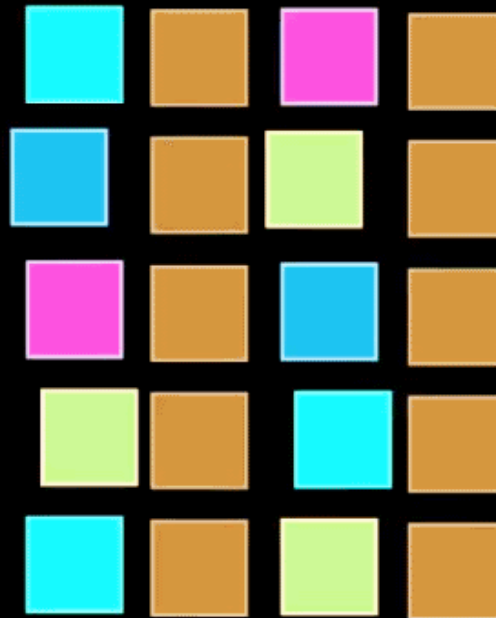
- Vertical vs. horizontal partitioning
- Round-robin, hash and range partitioning
- Declarative partitioning in PostgreSQL and tuple routing
- Attaching/detaching external partitions
- Foreign partitions
- Query optimizations: partition pruning, partition-wise joins and aggregates

- NoSQL / Big Data

- Distributed databases
- Key/value stores
- Map/Reduce
- Replicas, fault tolerance, concurrency and coherence

Part I

Semi-structured data in JSON and XML files



Data files

- A computer *file* is a container for data, and files have:
 - A *path* (sequence of folders and a filename):
`C:/Users/Steve/My Documents/my_data.csv`
 - A sequence of data bits “in” the file:
`00010101101101011101010111010010101000110101010111110`
 - Other metadata like *permissions*, *time of last modification*, depending on the filesystem type
- Files are:
 - *Persistent*, meaning that they remain in the computer after it is rebooted
 - *Sharable* by other programs running on the computer
- Thus, files allow programs to
 - Save their own data
 - Share data with other programs on the same computer
 - Transfer data between computers
- Databases are a more powerful alternative to plain files (“flat files”), but they are not as *portable*
 - we still use flat files to exchange bulk data

Standard data file formats

- The filename *extension* conventionally determines the *file format*
 - Tells us how to interpret the sequence of bits in the file
- Some file formats use *human-readable* ASCII or UTF-8 text
 - `txt`, `csv`, `json`, `xml`
- More efficient file formats represent data directly in *binary* form
 - `mat` (matlab), `RData`, `sqlite`, `jpg`, `zip`
- Some files use both formats in two stages:
 - human-readable files that have been compressed to a binary format:
 - `xlsx`, `docx`, `csv.gz`, `txt.gz`

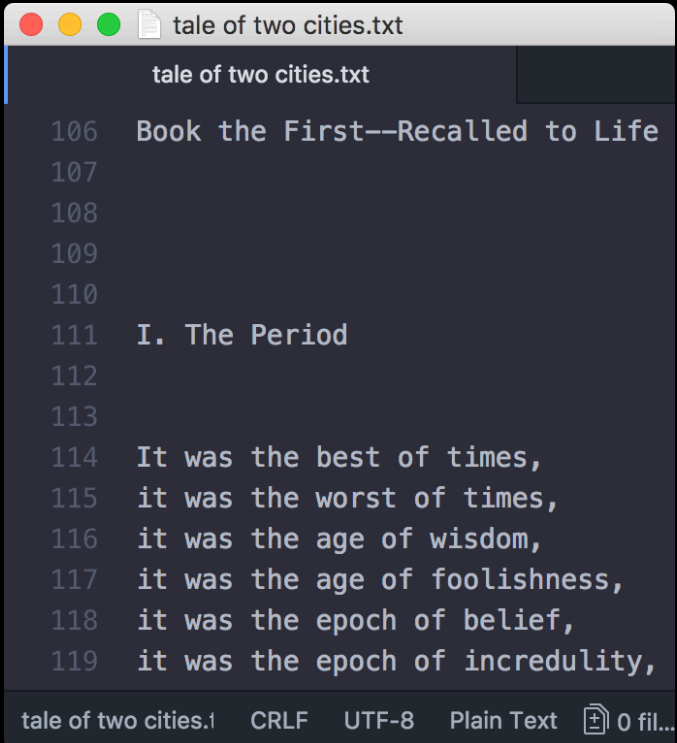
Encoding a text file (from Lecture 3)

C:\Users\Steve\My Documents\tale of two cities.txt

```
42 6f 6f 6b 20 74 68 65 20 46 69 72 73 74 2d 2d |Book the First--|
52 65 63 61 6c 6c 65 64 20 74 6f 20 4c 69 66 65 |Recalled to Life|
0d 0a 0d 0a 0d 0a 0d 0a 0d 0a 49 2e 20 54 68 65 |.....I. The|
20 50 65 72 69 6f 64 0d 0a 0d 0a 0d 0a 49 74 20 | Period.....It |
77 61 73 20 74 68 65 20 62 65 73 74 20 6f 66 20 |was the best of |
74 69 6d 65 73 2c 0d 0a 69 74 20 77 61 73 20 74 |times,..it was t|
68 65 20 77 6f 72 73 74 20 6f 66 20 74 69 6d 65 |he worst of time|
73 2c 0d 0a 69 74 20 77 61 73 20 74 68 65 20 61 |s,..it was the a|
67 65 20 6f 66 20 77 69 73 64 6f 6d 2c 0d 0a 69 |ge of wisdom,..i|
74 20 77 61 73 20 74 68 65 20 61 67 65 20 6f 66 |t was the age of|
20 66 6f 6f 6c 69 73 68 6e 65 73 73 2c 0d 0a 69 | foolishness,..i|
74 20 77 61 73 20 74 68 65 20 65 70 6f 63 68 20 |t was the epoch |
6f 66 20 62 65 6c 69 65 66 2c 0d 0a 69 74 20 77 |of belief,..it w|
61 73 20 74 68 65 20 65 70 6f 63 68 20 6f 66 20 |as the epoch of |
69 6e 63 72 65 64 75 6c 69 74 79 2c 0d 0a 69 74 |incredulity,..it|
```

Data bits in the the file, shown
in hex notation for brevity
(from “hexdump -C” command)

ASCII or UTF-8 encoding translates each
byte (or up to 4 bytes) to a character



```
106 Book the First--Recalled to Life
107
108
109
110
111 I. The Period
112
113
114 It was the best of times,
115 it was the worst of times,
116 it was the age of wisdom,
117 it was the age of foolishness,
118 it was the epoch of belief,
119 it was the epoch of incredulity,
```

Appearance in
text editor

Comma Separated Values (CSV)

- CSV is a simple text format for storing tabular data (spreadsheets)
- Each row is represented on one line of text
- Columns are separated by commas
- Values can be enclosed in double quotes ("...") if necessary
 - For example, if value includes comma or newline characters
 - Double quotes within a text value must be “escaped” by using two double quotes
 - e.g., for string Samsung Monitor 24” should write “Samsung Monitor 24”””
- Cells (values) can be empty by having nothing between the commas

titanic_passenger_list.csv viewed in Excel

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	pclass	survived	name	sex	age	sibsp	parch	ticket	fare	cabin	embarked	boat	body	home.dest
2	1	1	Allen, Miss. Elisabeth Walton	female	29	0	0	24160	211.338	B5	S	2		St Louis, MO
3	1	1	Allison, Master. Hudson Trevor	male	0.92	1	2	113781	151.55	C22 C26	S	11		Montreal, PQ / Chesterville, ON
4	1	0	Allison, Miss. Helen Loraine	female	2	1	2	113781	151.55	C22 C26	S			Montreal, PQ / Chesterville, ON
5	1	0	Allison, Mr. Hudson Joshua Creighton	male	30	1	2	113781	151.55	C22 C26	S		135	Montreal, PQ / Chesterville, ON
6	1	0	Allison, Mrs. Hudson J C (Bessie Waldo Daniels)	female	25	1	2	113781	151.55	C22 C26	S			Montreal, PQ / Chesterville, ON
7	1	1	Anderson, Mr. Harry	male	48	0	0	19952	26.55	E12	S	3		New York, NY
8	1	1	Andrews, Miss. Kornelia Theodosia	female	63	1	0	13502	77.9583	D7	S	10		Hudson, NY
9	1	0	Andrews, Mr. Thomas Jr	male	39	0	0	112050	0	A36	S			Belfast, NI
10	1	1	Appleton, Mrs. Edward Dale (Charlotte Lamson)	female	53	2	0	11769	51.4792	C101	S	D		Bayside, Queens, NY
11	1	0	Artagaveytia, Mr. Ramon	male	71	0	0	PC 17609	49.5042		C		22	Montevideo, Uruguay
12	1	0	Astor, Col. John Jacob	male	47	1	0	PC 17757	227.525	C62 C64	C		124	New York, NY
13	1	1	Astor, Mrs. John Jacob (Madeleine Talmadge Force)	female	18	1	0	PC 17757	227.525	C62 C64	C	4		New York, NY
14	1	1	Aubart, Mme. Leontine Pauline	female	24	0	0	PC 17477	69.3	B35	C	9		Paris, France
15	1	1	Barber, Miss. Ellen "Nellie"	female	26	0	0	19877	78.85		S	6		
16	1	1	Barkworth, Mr. Algernon Henry Wilson	male	80	0	0	27042	30	A23	S	B		Hessle, Yorks
17	1	0	Baumann, Mr. John D	male		0	0	PC 17318	25.925		S			New York, NY
18	1	0	Baxter, Mr. Quigg Edmond	male	24	0	1	PC 17558	247.521	B58 B60	C			Montreal, PQ
19	1	1	Baxter, Mrs. James (Helene DeLaunierie Chaput)	female	50	0	1	PC 17558	247.521	B58 B60	C	6		Montreal, PQ
20	1	1	Bazzani, Miss. Albina	female	32	0	0	11813	76.2917	D15	C	8		
21	1	0	Beattie, Mr. Thomson	male	36	0	0	13050	75.2417	C6	C	A		Winnipeg, MN
22	1	1	Beckwith, Mr. Richard Leonard	male	37	1	1	11751	52.5542	D35	S	5		New York, NY
23	1	1	Beckwith, Mrs. Richard Leonard (Sallie Monypeny)	female	47	1	1	11751	52.5542	D35	S	5		New York, NY
24	1	1	Behr, Mr. Karl Howell	male	26	0	0	111369	30	C148	C	5		New York, NY
25	1	1	Bidois, Miss. Rosalie	female	42	0	0	PC 17757	227.525		C	4		
26	1	1	Bird, Miss. Ellen	female	29	0	0	PC 17483	221.779	C97	S	8		
27	1	0	Birnbaum, Mr. Jakob	male	25	0	0	13905	26		C		148	San Francisco, CA
28	1	1	Bishop, Mr. Dickinson H	male	25	1	0	11967	91.0792	B49	C	7		Dowagiac, MI
29	1	1	Bishop, Mrs. Dickinson H (Helen Walton)	female	19	1	0	11967	91.0792	B49	C	7		Dowagiac, MI
30	1	1	Bissette, Miss. Amelia	female	35	0	0	PC 17760	135.633	C99	S	8		
31	1	1	Bjornstrom-Steffansson, Mr. Mauritz Hakan	male	28	0	0	110564	26.55	C52	S	D		Stockholm, Sweden / Washington, DC
32	1	0	Blackwell, Mr. Stephen Weart	male	45	0	0	113784	35.5	T	S			Trenton, NJ
33	1	1	Blank, Mr. Henry	male	40	0	0	112277	31	A31	C	7		Glen Ridge, NJ
34	1	1	Bonnell, Miss. Caroline	female	30	0	0	36928	164.867	C7	S	8		Youngstown, OH

titanic_passenger_list.csv viewed as text

```
"pclass","survived","name","sex","age","sibsp","parch","ticket","fare","cabin","embarked","boat","body","home.dest"
1,1,"Allen, Miss. Elisabeth Walton","female",29,0,0,"24160",211.3375,"B5","S",2,,,"St Louis, MO"
1,1,"Allison, Master. Hudson Trevor","male",0.92,1,2,"113781",151.5500,"C22 C26","S",11,,,"Montreal, PQ / Chesterville, ON"
1,0,"Allison, Miss. Helen Loraine","female",2,1,2,"113781",151.5500,"C22 C26","S",,,,"Montreal, PQ / Chesterville, ON"
1,0,"Allison, Mr. Hudson Joshua Creighton","male",30,1,2,"113781",151.5500,"C22 C26","S",,"135",,"Montreal, PQ / Chesterville, ON"
1,0,"Allison, Mrs. Hudson J C (Bessie Waldo Daniels)","female",25,1,2,"113781",151.5500,"C22 C26","S",,,,"Montreal, PQ / Chesterville, ON"
1,1,"Anderson, Mr. Harry","male",48,0,0,"19952",26.5500,"E12","S",3,,,"New York, NY"
1,1,"Andrews, Miss. Kornelia Theodosia","female",63,1,0,"13502",77.9583,"D7","S",10,,,"Hudson, NY"
1,0,"Andrews, Mr. Thomas Jr","male",39,0,0,"112050",0.0000,"A36","S",,,,"Belfast, NI"
1,1,"Appleton, Mrs. Edward Dale (Charlotte Lamson)","female",53,2,0,"11769",51.4792,"C101","S",D,,,"Bayside, Queens, NY"
1,0,"Artagaveytia, Mr. Ramon","male",71,0,0,"PC 17609",49.5042,,,"C",,"22",,"Montevideo, Uruguay"
1,0,"Astor, Col. John Jacob","male",47,1,0,"PC 17757",227.5250,"C62 C64","C",,"124",,"New York, NY"
1,1,"Astor, Mrs. John Jacob (Madeleine Talmadge Force)","female",18,1,0,"PC 17757",227.5250,"C62 C64","C",4,,,"New York, NY"
1,1,"Aubart, Mme. Leontine Pauline","female",24,0,0,"PC 17477",69.3000,"B35","C",9,,,"Paris, France"
1,1,"Barber, Miss. Ellen ""Nellie""","female",26,0,0,"19877",78.8500,,,"S",6,,,"
1,1,"Barkworth, Mr. Algernon Henry Wilson","male",80,0,0,"27042",30.0000,"A23","S",B,,,"Hessle, Yorks"
1,0,"Baumann, Mr. John D","male",,0,0,"PC 17318",25.9250,,,"S",,,,"New York, NY"
1,0,"Baxter, Mr. Quigg Edmond","male",24,0,1,"PC 17558",247.5208,"B58 B60","C",,,,"Montreal, PQ"
1,1,"Baxter, Mrs. James (Helene DeLaudeniére Chaput)","female",50,0,1,"PC 17558",247.5208,"B58 B60","C",6,,,"Montreal, PQ"
1,1,"Bazzani, Miss. Albina","female",32,0,0,"11813",76.2917,"D15","C",8,,,"
1,0,"Beattie, Mr. Thomson","male",36,0,0,"13050",75.2417,"C6","C",A,,,"Winnipeg, MN"
1,1,"Beckwith, Mr. Richard Leonard","male",37,1,1,"11751",52.5542,"D35","S",5,,,"New York, NY"
1,1,"Beckwith, Mrs. Richard Leonard (Sallie Monypeny)","female",47,1,1,"11751",52.5542,"D35","S",5,,,"New York, NY"
1,1,"Behr, Mr. Karl Howell","male",26,0,0,"111369",30.0000,"C148","C",5,,,"New York, NY"
1,1,"Bidois, Miss. Rosalie","female",42,0,0,"PC 17757",227.5250,,,"C",4,,,"
1,1,"Bird, Miss. Ellen","female",29,0,0,"PC 17483",221.7792,"C97","S",8,,,"
1,0,"Birnbaum, Mr. Jakob","male",25,0,0,"13905",26.0000,,,"C",,"148",,"San Francisco, CA"
1,1,"Bishop, Mr. Dickinson H","male",25,1,0,"11967",91.0792,"B49","C",7,,,"Dowagiac, MI"
1,1,"Bishop, Mrs. Dickinson H (Helen Walton)","female",19,1,0,"11967",91.0792,"B49","C",7,,,"Dowagiac, MI"
1,1,"Bishop, Miss. Amelia","female",25,0,0,"PC 17760",125.6222,"C60","C",8,,,"
```

CSV files represent a single table

- Relational models for complex data involve several tables, so you need several CSV files to represent complex data
- Groups of CSV files are often used for data exchange
- The CSV file can have *column names* in the first row
- However, other important schema information is not stored in CSV:
 - Data types
 - Primary keys
 - Unique key constraints
 - Foreign key relationships
 - Indexes
- The above *metadata* can be included in an SQL script that accompanies the CSV files, or in a human-readable document
- Each DBMS also has its own proprietary format for exchanging databases, including both the data and metadata
- SQLite is the simplest. Its just the `*.sqlite` file

Semi-structured data

- Often we want to represent complex data in a single file and in a standard way
 - JSON and XML files store semi-structured data
 - Not limited to two dimensions like CSV files
 - Data is represented in a tree format, where any node can have more details below it
 - However, unlike a relational database, there is no clear pre-defined structure for the data
 - The data defines its own structure
-
- Compared to CSV, it is more difficult to read and is more prone to errors because data elements can be missing

JSON – JavaScript Object Notation

- Used in many web applications and data APIs
- Allows an arbitrary amount of nesting
- Spaces are ignored, except within quotes
- Basic components are:
 - `[]` for ordered lists
 - Items are separated by commas
 - Items can be any JSON
 - `{ }` for unordered dictionaries/objects
 - Key/value pairs are separated by commas
 - Keys must be strings (text)
 - Values can be any JSON
 - Numbers, `true`, `false`, `null`
 - Strings (text) in double quotes

```
[  
  {  
    "name": "John",  
    "age": 30,  
    "cars": [  
      "Ford", "BMW", "Fiat"]  
    },  
  {  
    "name": "Alicia",  
    "age": 32,  
    "hometown": "Seattle"  
  }  
]
```

XML – eXtensible Markup Language

- Older than JSON
- Used in HTML for web page design
- Basic components are:
 - Text
 - Tags
 - `<tagname>...</tagname>`
 - Have a name, and have XML inside
 - Each start tag has a corresponding end tag
 - Attributes
 - `<tag attr="value" ...>`
 - Appear within tags
 - Attribute name and value must be text
 - Tag can have multiple attributes, but each must have a unique name

```
<people>
  <person name="John"
          age="30">
    <cars>
      <car>Ford</car>
      <car>BMW</car>
      <car>Fiat</car>
    </cars>
  </person>
  <person name="Alicia"
          age="32">
    <hometown city="Seattle">
      </hometown>
    </person>
</people>
```

Part II

Cleaning Messy Data



Real world data is often *messy*

The Database view

- Data entered manually can have **typos**
- Buggy computer programs can produce bad results
 - Program may not behave well when data is **missing**
 - Data overflow and type conversion can cause problems
- Temporary sensor malfunction can lead to a **bogus** measurement
- Different data sources can use different **naming conventions**
- Data can be missing due to an interrupted data import
- Some data may have been **corrupted**

Real world data is often *messy*

The Statistics view

- There is a process that produces data
- We want to model ideal samples of that process, but in practice we have non-ideal samples:
 - **Distortion** – some samples are corrupted by a process
 - **Selection Bias** - likelihood of a sample depends on its value
 - **Left and right censorship** - users come and go from our scrutiny
 - **Dependence** – samples are supposed to be independent, but are not (e.g. social networks)
- You can add new models for each type of imperfection, but you can't model everything
- What's the best trade-off between accuracy and simplicity?

Real world data is often *messy*

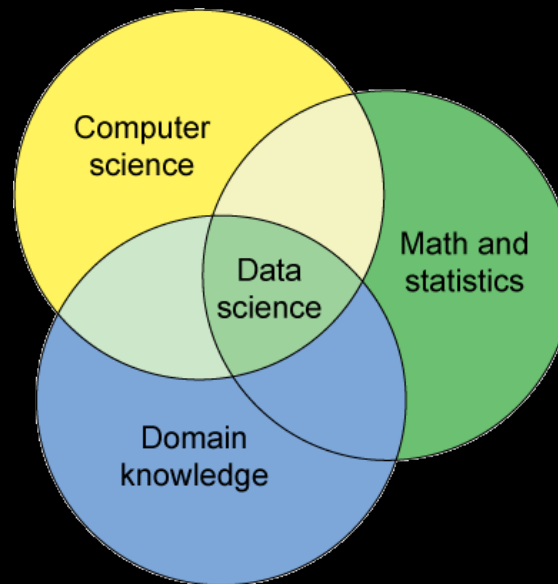
The Domain Expert's view

- These data don't look right
 - This answer doesn't look right
 - This process doesn't pass my sniff test
 - Hmmm... That's funny!
-
- Domain experts have an implicit model of the data that they can test against...

Real world data is often *messy*

The Data Scientist's view

- Some combination of the above...



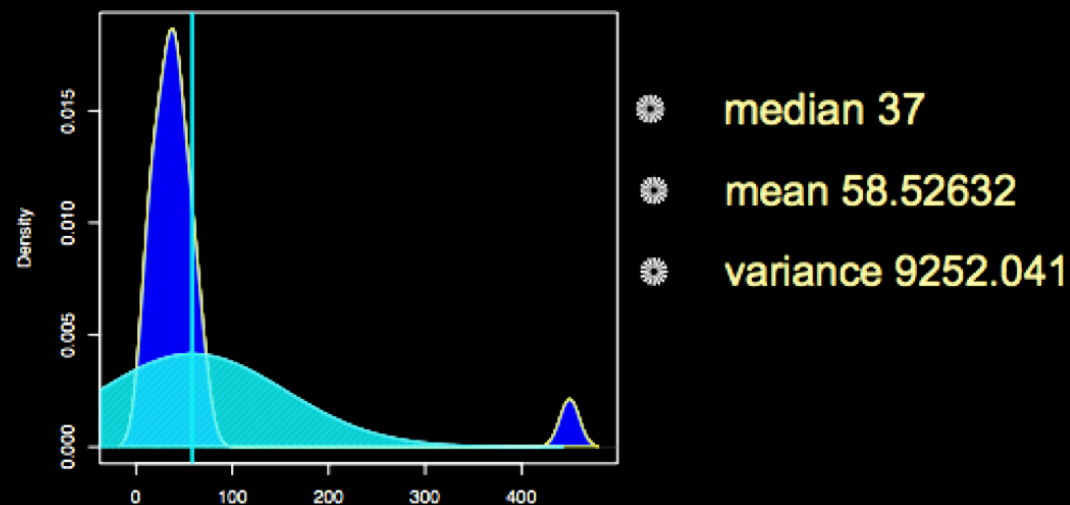
Data quality problems

- (Source) Data are dirty on their own
- Transformations corrupt the data (complexity of software pipelines)
- Data sets are clean but integration (i.e., combining them) screws them up
- “Rare” errors can become frequent after transformation or integration
- Data sets are clean but suffer “bit rot”
 - Old data loses its value/accuracy over time
- Any combination of the above

Numeric outliers

12	13	14	21	22	26	33	35	36	37	39	42	45	47	54	57	61	68	450
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	-----

ages of employees (US)

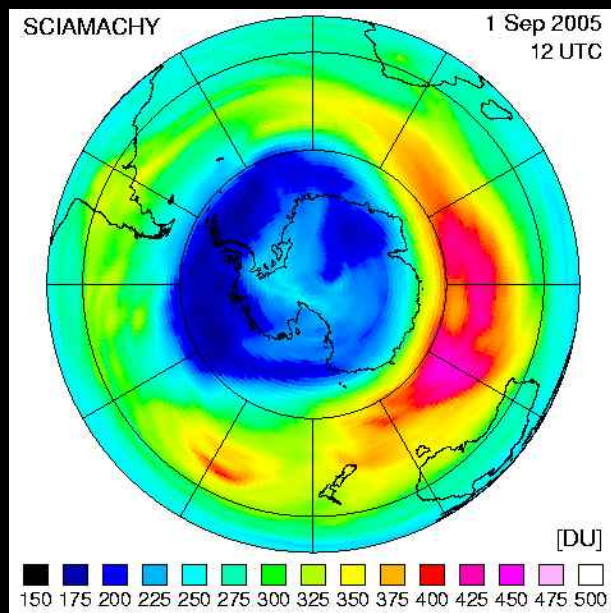


Adapted from Joe Hellerstein's 2012 CS 194 Guest Lecture

Data cleaning makes everything okay?

The appearance of a hole in the earth's ozone layer over Antarctica, first detected in 1976, was so unexpected that scientists didn't pay attention to what their instruments were telling them; they thought their instruments were malfunctioning.

National Center for Atmospheric Research



In fact, the data were rejected as unreasonable by data quality control algorithms

How to recognize bad data?

No simple or easy answer

- Start with good documentation
- Know what each column means
- Define a very strict schema and look for warnings when importing
 - Define columns as `NOT NULL`, when appropriate, to prevent incomplete data
 - Define columns with numeric types rather than text if you expect numbers
 - Define *foreign keys* if you expect columns to match between tables
 - Define triggers to implement integrity constraints
- Look at summary statistics after data is imported:
`SELECT MIN(col), MAX(col), AVG(col)...`
 - If min and max values are unexpected, then look for outliers by sorting according to that column
 - In R, use the `summary(...)` command on a data frame

Conventional definition of data quality

- Accuracy
 - The data was recorded correctly
- Completeness
 - All relevant data was recorded
- Uniqueness
 - Entities are recorded once
- Timeliness
 - The data is kept up to date
 - Special problems in federated data: time consistency
- Consistency
 - The data agrees with itself

Problems...

- Unmeasurable
 - Accuracy and completeness are extremely difficult, perhaps impossible to measure
- Context independent
 - No accounting for what is important. E.g., if you are computing aggregates, you can tolerate a lot of inaccuracy
- Incomplete
 - What about interpretability, accessibility, metadata, analysis, etc
- Vague
 - The conventional definitions provide no guidance towards practical improvements of the data

Finding a modern definition

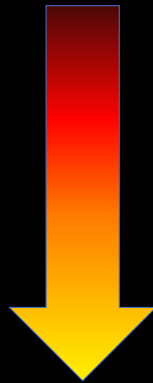
- We need a definition of data quality which
 - Reflects the **use** of the data
 - Leads to **improvements in processes**
 - Is **measurable** (we can define metrics)
- First, we need a better understanding of how and where data quality problems occur
 - The **data quality continuum**

Meaning of Data Quality

- There are many uses of data
 - Operations
 - Aggregate analysis
 - Customer relations ...
- Data Interpretation : the data is useless if we don't know all of the rules behind the data
- Data Suitability : Can you get the answer from the available data
 - Use of proxy data
 - Relevant data is missing

The data quality continuum

- Data and information is not static
- It flows through a collection and usage process
 - Data gathering
 - Data delivery
 - Data storage
 - Data integration
 - Data retrieval
 - Data mining/analysis



Data gathering

- How does the data enter the system?
- Sources of problems:
 - Manual entry
 - No uniform standards for content and formats
 - Parallel data entry (duplicates)
 - Approximations, surrogates – SW/HW constraints
 - Measurement or sensor errors

Data gathering - solutions

- Potential Solutions:
 - Preemptive:
 - Process architecture (build in integrity checks)
 - Process management (reward accurate data entry, data sharing, data stewards)
 - Retrospective:
 - Cleaning focus (duplicate removal, merge/purge, name & address matching, field value standardization)
 - Diagnostic focus (automated detection of glitches)

Data delivery

- Destroying or mutilating information by inappropriate pre-processing
 - Inappropriate aggregation
 - Nulls converted to default values
- Loss of data:
 - Buffer overflows
 - Transmission problems
 - No checks

Data delivery - solutions

- Build reliable transmission protocols
 - Use a relay server
- Verification
 - Checksums, verification parser
 - Do the uploaded files fit an expected pattern?
- Relationships
 - Are there dependencies between data streams and processing steps
- Interface agreements
 - Data quality commitment from the data stream supplier

Data storage

- You get a data set. What do you do with it?
- Problems in physical storage
 - Can be an issue, but terabytes are cheap.
- Problems in logical storage
 - Poor metadata.
 - Data feeds are often derived from application programs or legacy data sources. What does it mean?
 - Inappropriate data models.
 - Missing timestamps, incorrect normalization, etc.
 - Ad-hoc modifications.
 - Structure the data to fit the GUI.
 - Hardware / software constraints.
 - Data transmission via Excel spreadsheets, Y2K, Y2038

Adapted from Ted Johnson's SIGMOD 2003 Tutorial

Data storage - solutions

- Metadata
 - Document and publish data specifications
- Planning
 - Assume that everything bad will happen
 - Can be very difficult
- Data exploration
 - Use data browsing and data mining tools to examine the data
 - Does it meet the specifications you assumed?
 - Has something changed?

Debugging a data import

- If data fails to import completely, try loading it into a *temporary text table*
 - Drop keys and use large text types for every column
- Query the text table to look for unexpected values in the source data

This table has strict constraints on what kind of data can be inserted:

```
CREATE TABLE person (  
  SSN int NOT NULL,  
  firstName varchar(30) NOT NULL,  
  lastName varchar(30) NOT NULL,  
  birthDate datetime NOT NULL,  
  PRIMARY KEY (SSN)  
);
```

This temporary table relaxes those constraints:

```
CREATE TABLE _import_person (  
  SSN varchar(1000) NOT NULL,  
  firstName varchar(1000) NOT NULL,  
  lastName varchar(1000) NOT NULL,  
  birthDate varchar(1000) NOT NULL,  
);
```

Data integration

- Combine data sets (acquisitions, across departments)
- Common source of problems
 - Heterogenous data: no common key, different field formats
 - **Approximate matching**
 - Different definitions
 - What is a customer: an account, an individual, a family, ...
 - Time synchronization
 - Does the data relate to the same time periods? Are the time windows compatible?
 - Legacy data
 - IMS, spreadsheets, ad-hoc structures

Named entity matching

- In real-world data, people, companies, products, etc., all can be represented with variations of their name:
 - Eleanor Roosevelt
 - E. Roosevelt
 - Roosevelt, Eleanor
 - Mrs. Roosevelt
 - Northwestern Univ.
 - NU
 - Northwestern
 - Northwestern University
 - Apple iPhone 6S
 - iPhone 6S 32 GB Space Gray
 - A1633
- When combining data from multiple sources, we need *approximate matching* to join according to text fields
 - Humans are good at this, but it's difficult to automate

Approximate matching

- Relate tuples whose fields are “close”
 - Approximate string matching
 - Generally, based on edit distance
- Approximate tree matching
 - For Nested Data Structures (or flattened ones)
 - Much more expensive than string matching
 - Recent research in fast approximations
- Feature vector matching
 - Similarity search
 - Many techniques discussed in the data mining literature
- Ad-hoc or Domain-focused matching
 - Use domain insights and/or clever tricks

SQL synonym table

- The simplest solution is to create a *synonym table* to list all variations of names
- Use the synonym table as a linking table in a four-way join
- For example, if *product* and *product_details* use different variations of the product name:

```
SELECT * FROM product
  INNER JOIN product_synonym AS n1
    ON product.name=n1.name
  INNER JOIN product_synonym AS n2
    ON n1.id=n2.id
  INNER JOIN product_details
    ON n2.name=product_details.name;
```

product_synonym	
<i>product_id</i>	<i>name</i>
1	Apple iPhone 6s
1	iPhone 6 S
1	iPhone 6S 32 GB
1	iPhone 6S Space Gray
1	iPhone 6S Gold
2	Google Nexus 6P
2	Nexus 6P
2	Nexus 6-P

Shortcomings of synonym table

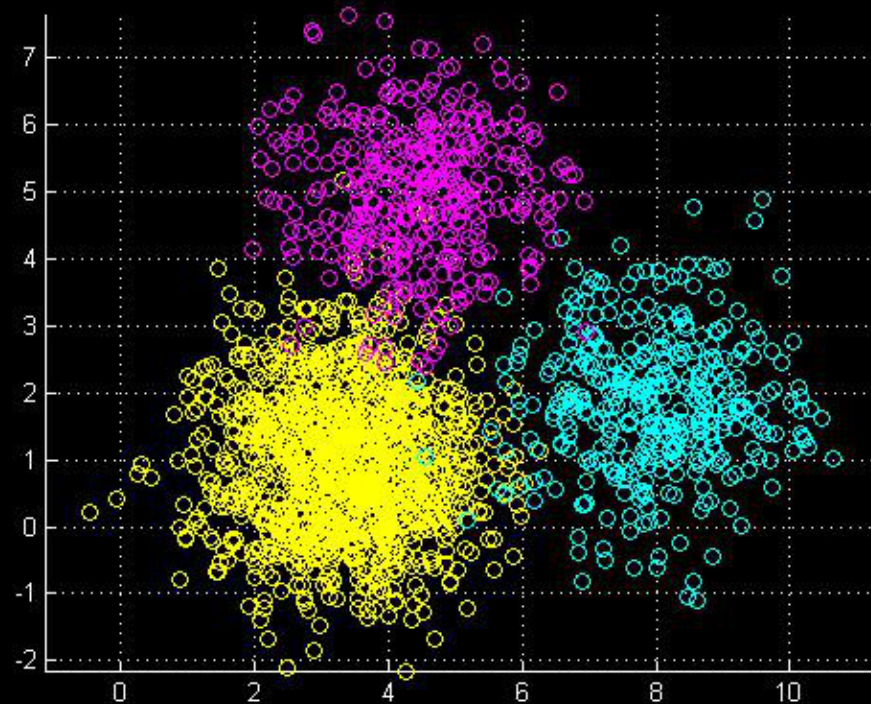
- Creating the synonym table is a slow, manual process
 - Cannot scale to many thousands of rows
- Synonym table must be updated every time new data arrives

Automatically generating synonyms

- This requires a general purpose programming language
 - Cannot do it in plain SQL
- We can think of named entity matching as a *clustering* or *classification* problem, both suitable for machine learning
- Problem definition:
 - Given:
 - A list of names (unknown strings)
 - Optionally, a list of entities
 - Optionally, a known map between a subset of the names and the entities (training data)
 - Produce:
 - A mapping between all the names and a list of entities

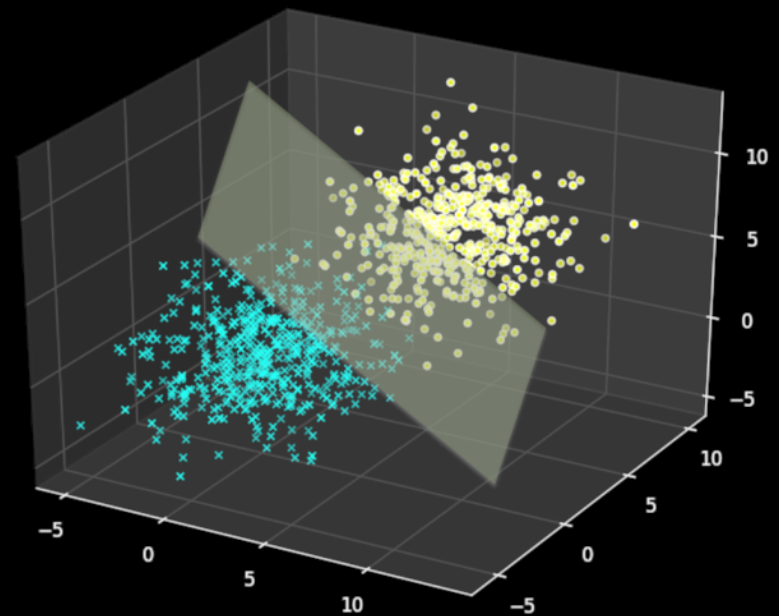
Clustering

- If we do not know what entities exist in the data, then try to find clusters of names and call each cluster an entity



Classification

- If we have a list of entities and some training samples for each, then we can train a classifier for each entity
- Each classifier would return yes/no for a name matching a particular entity, along with a confidence measure
- Choose the entity whose classifier has highest confidence



Some similarity measures

Handle
Typographical errors

- Equality on a boolean predicate
- Edit distance
 - Levenstein, Smith-Waterman, Affine

- Set similarity
 - Jaccard, Dice
- Vector Based
 - Cosine similarity, TFIDF

Good for Text like
reviews/ tweets

- Useful packages:

Good for Names

- Alignment-based or Two-tiered
 - Jaro-Winkler, Soft-TFIDF, Monge-Elkan
- Phonetic Similarity
 - Soundex
- Translation-based
- Numeric distance between values
- Domain-specific

Useful for
abbreviations,
alternate names.

Text similarity metrics

- An alternative to ML is a graph partitioning approach
- Use text similarity metrics to build a name similarity graph
- For example, the *edit distance* (or Levenshtein distance)
 - The minimum number of single-character changes needed to make one phrase equal to another
 - Edit distance between “school” and “college” is 7
 - Delete *s*, *h*, *o*, and append “*lege*” at the end
 - Edit distance between “iPhone 6S” and “iPhone 6-S” is just 1
 - Delete the hyphen

Soundex encoding

- A phonetic algorithm that indexes names by their sounds when pronounced in English
- Consists of the first letter of the name followed by three numbers
- The numbers encode similar sounding consonants
 - Remove all W, H
 - B, F, P, V encoded as 1
 - C, G, J, K, Q, S, X, Z encoded as 2
 - Remaining encodings: D, T = 3; L = 4; M, N = 5; R = 6
 - Remove vowels
 - Concatenate first letter of string with first 3 numerals
- Example: “great” and “grate” become G6EA3 and G6A3E, and then G63
- More recent: metaphone, double metaphone, etc.

Schema matching

- Use similarity measures and structural cues (e.g., column names, data types, etc.) to match data definitions
- Looking at data instances (or examples of them can help)
- Constraints in the schema (if you have them) can also help
- Auxiliary Information: dictionaries, documentation, usage... ditto

Data retrieval

- Exported data sets are often a view of the actual data. Problems occur because:
 - Source data not properly understood
 - Need for derived data not understood
 - Just plain mistakes
 - Inner join vs. outer join
 - Understanding NULL values
- Computational constraints
 - E.g., too expensive to give a full history, we'll supply a snapshot
- Incompatibility
 - ASCII? Ebcdic? Unicode?

Data mining and analysis

- What are you doing with all this data anyway?
- Problems in the analysis
 - Scale and performance
 - Confidence bounds?
 - Black boxes and dart boards
 - Attachment to models
 - Insufficient domain expertise
 - Casual empiricism

Retrieval and mining - solutions

- Data exploration
 - Determine which models and techniques are appropriate, find data bugs, develop domain expertise
- Continuous analysis
 - Are the results stable? How do they change?
- Accountability
 - Make the analysis part of the feedback loop

Data quality constraints

- Many data quality problems can be captured by static constraints based on the schema
 - Nulls not allowed, type domains, foreign key constraints, etc
- Many others are due to problems in workflow, and can be captured by dynamic constraints
 - E.g., orders above \$200 are processed by Biller 2
- The constraints follow an 80-20 rule
 - A few constraints capture most cases, thousands of constraints to capture the last few cases
- Constraints are measurable
 - **Data Quality Metrics?**

Data quality metrics

- We want a measurable quantity
 - Indicates what is wrong and how to improve
 - Realize that DQ is a messy problem, no set of numbers will be perfect
- Types of metrics
 - Static vs. dynamic constraints
 - Operational vs. diagnostic
- Metrics should be directionally correct with an improvement in use of the data
- A very large number metrics are possible
 - Choose the most important ones

Examples of data quality metrics

- Conformance to schema
 - Evaluate constraints on a snapshot
- Conformance to business rules
 - Evaluate constraints on changes in the database
- Accuracy
 - Perform inventory (expensive), or use proxy (track complaints). Audit samples?
- Accessibility
- Interpretability
- Glitches in analysis
- Successful completion of end-to-end process

Technical approaches

- We need a multi-disciplinary approach to attack data quality problems
- No one approach solves all problems
- **Process management**
 - Ensure proper procedures
- **Statistics**
 - Focus on analysis: find and repair anomalies in data
- **Database**
 - Focus on relationships: ensure consistency
- **Metadata / domain expertise**
 - What does it mean? Interpretation

Extract Transform Load (ETL)

- ETL programs import data from data files to databases
 - For example, *MS SQL Server Integration Services*
- ETL script can include:
 - List of validation rules to identify problematic data
 - List of behaviors to correct or discard problematic data
 - Transformations to apply to the input data before inserting into DB

Data cleaning is an active research field

- Identifying and cleaning messy data is a real world problem
- You cannot get correct results without correct data
- Very active research field today
- Intersection of machine learning and database systems

Several tools, open and proprietary

- Open Refine (Google)

<http://openrefine.org>

- Messy data identification and cleaning
- Data transformation
- Linking data to databases

- Data Wrangler (Stanford)

<http://vis.stanford.edu/wrangler/>

- Graduated to a startup (Trifacta)
- Interactive tool for data cleaning and transformation

ML for data cleaning

- Snorkel (Stanford)
<http://snorkel.stanford.edu>
 - Classification
 - Data cleaning and integration
 - Entity and relationship extraction
 - Deep learning → feature engineering and extraction
 - Requires massive training tests
 - How to label data for training ?

More to read

- Big Data's Dirty Problem
[Fortune]
<http://fortune.com/2014/06/30/big-data-dirty-problem/>
- A Taxonomy of Dirty Data
[Won Kim+]
<http://sci2s.ugr.es/docencia/m1/KimTaxonomy03.pdf>
(Very detailed, slightly outdated)
- For Big-Data Scientists, 'Janitor Work' Is Key Hurdle to Insights
[New York Times]
http://www.nytimes.com/2014/08/18/technology/for-big-data-scientists-hurdle-to-insights-isjanitor-work.html?_r=0