NYU

# Data Profiling & Data Cleaning

**Heiko Mueller**
*Research Engineer – Center for Data Science*
heiko.mueller @ nyu.edu

https://na.rdcpix.com/53259694/b57cb3fe060b4e365f4756e99b2b4287w-c265838xd-w685_h860_q80.jpg

MAR 23, 2016 @ 09:33 AM 15,078 VIEWS

# Cleaning Big Data: Most Time-Consuming, Least Enjoyable Data Science Task, Survey Says

**Gil Press,** CONTRIBUTOR

*I write about technology, entrepreneurs and innovation.* FULL BIO

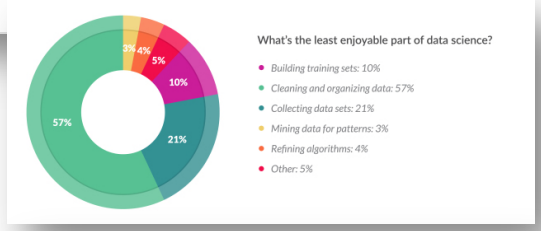
Opinions expressed by Forbes Contributors are their own.

**TWEET THIS**

data scientists found that they spend most of their time massaging rather than mining or modeling data.

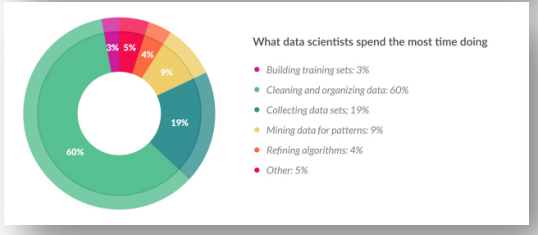
76% of data scientists view data preparation as the least enjoyable part of their work

A new survey of data scientists found that they spend most of their time massaging rather than mining or modeling data. Still, most are happy with having the sexiest job of the 21st century. The survey of about 80 data scientists was conducted for the second year in a row by CrowdFlower, provider of a "data enrichment" platform for data scientists. Here are the highlights:




***Least enjoyable part of Data Science?***

Collecting data (**21%**)
Cleaning and organizing data (**57%**)




***Spend most time doing***

Collecting data (**19%**)
Cleaning and organizing data (**60%**)

https://www.forbes.com/sites/gilpress/2016/03/23/data-preparation-most-time-consuming-least-enjoyable-data-science-task-survey-says

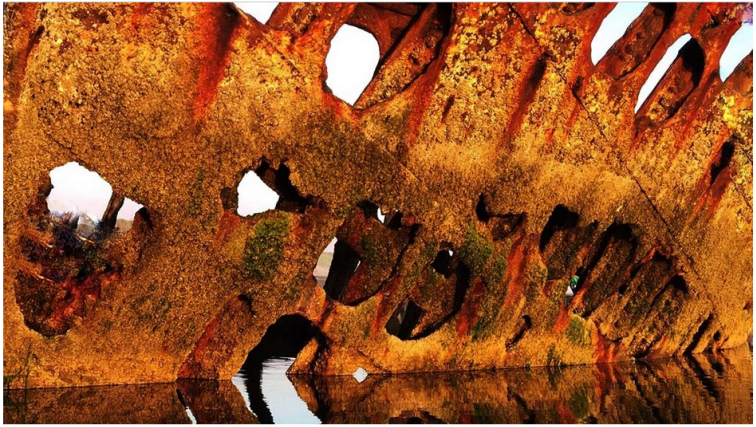
DATA

# Bad Data Costs the U.S. $3 Trillion Per Year

by Thomas C. Redman

SEPTEMBER 22, 2016





Consider this figure: $136 billion per year. That's the research firm IDC's estimate of the size of the big data market, worldwide, in 2016. This figure should surprise no one with an interest in big data.

But here's another number: $3.1 *trillion*, IBM's estimate of the yearly cost of poor quality data, in the US alone, in 2016. While most people who deal in data every day know that bad data is costly, this figure stuns.

While the numbers are not really comparable, and there is considerable variation around each, one can only conclude that right now, improving data quality represents the far larger data opportunity. Leaders are well-advised to develop a deeper appreciation for the opportunities improving data quality present and take fuller advantage than they do today.

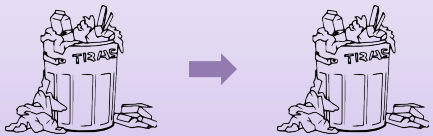
The reason bad data costs so much is that decision makers, managers, knowledge workers, data scientists, and others must accommodate it in their everyday work. And doing so is both time-consuming and expensive. The data they need has plenty

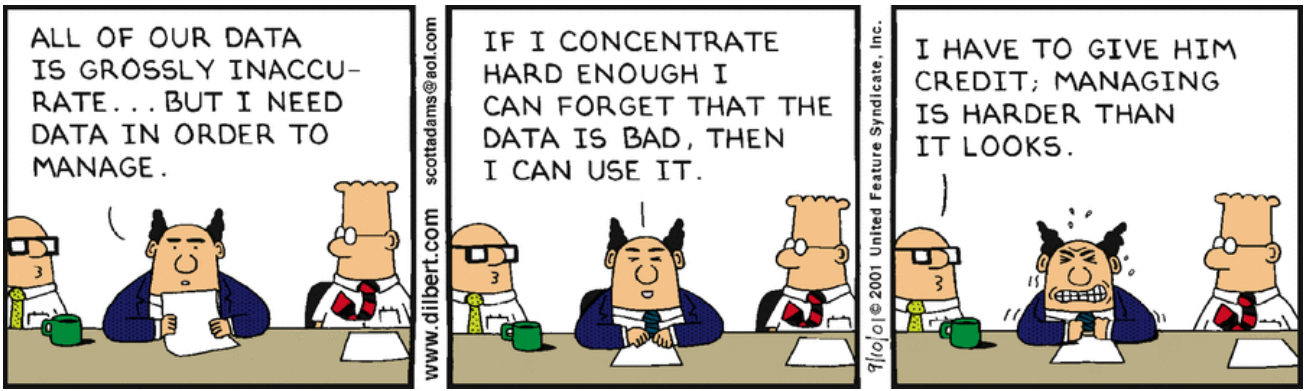
https://hbr.org/2016/09/bad-data-costs-the-u-s-3-trillion-per-year

# The Impact of Poor Data Quality

Organizational **data is a critical resource** that supports business processes and managerial decision making.

As data volumes increase, so does the complexity of managing it and the **risks of poor data quality**.



# What is Data Cleaning?

… **data** is generally considered high **quality** if it is "***fit for [its] intended uses*** *in operations, decision making and planning*"

Thomas C. Redman, Data Driven: Profiting from Your Most Important Business Asset. 2013

Even though quality cannot be defined, you know what it is.

Robert M. Prisig, Zen and the Art of Motorcycle Maintenance, 1975

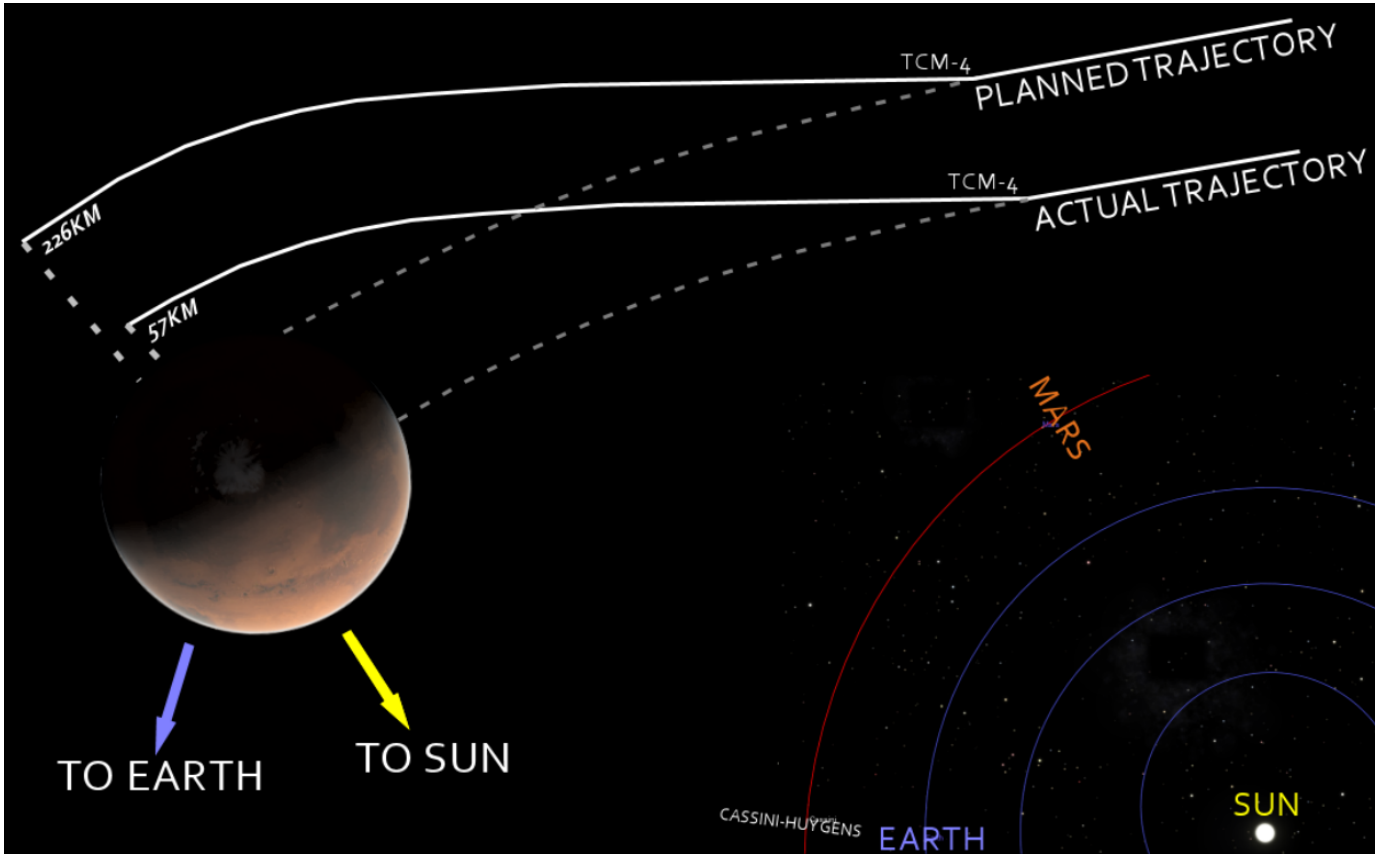
Data of poor quality is lacking rich metadata.

Divesh Srivastava, AT&T Research

**Data cleansing** or **data cleaning** is the process of detecting and repairing corrupt or inaccurate records from a data set in order to improve the quality of data.

https://en.wikipedia.org/wiki/Data_cleansing & Erhard Rahm, Hong Hai Do: Data Cleaning: Problems and Current Approaches, IEEE Data Engineering Bulletin, 2000.

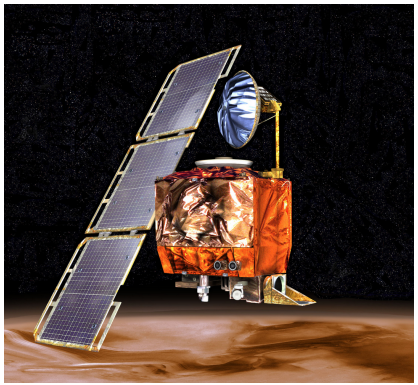
# Fate of the NASA Mars Climate Orbiter (1999)




https://en.wikipedia.org/wiki/Mars_Climate_Orbiter

# Fate of the NASA Mars Climate Orbiter (1999)





## Mars Probe Lost Due to Simple Math Error

**October 01, 1999** | ROBERT LEE HOTZ | TIMES SCIENCE WRITER



NASA lost its $125-million Mars Climate Orbiter because spacecraft engineers failed to convert from English to metric measurements when exchanging vital data before the craft was launched, space agency officials said Thursday.

A navigation team at the Jet Propulsion Laboratory used the metric system of millimeters and meters in its calculations, while Lockheed Martin Astronautics in Denver, which designed and built the spacecraft, provided crucial acceleration data in the English system of inches, feet and pounds.

As a result, JPL engineers mistook acceleration readings measured in English units of pound-seconds for a metric measure of force called newton-seconds.

In a sense, the spacecraft was lost in translation.

"That is so dumb," said John Logsdon, director of George Washington University's space policy institute. "There seems to have emerged over the past couple of years a systematic problem in the space community of insufficient attention to detail."

**FROM THE ARCHIVES**

Math Error Exaggerated TriZetto Loss Estimate
*March 7, 2001*

Math Error Inflated Ventura Blvd. Cost : Traffic: Mistake...

http://articles.latimes.com/1999/oct/01/news/mn-17288

*Why is Data Cleaning so Difficult and Time Consuming?*

# Why is Data Cleaning Time Consuming?

***1) One of the most 'annoying' parts of data analysis is the variety of data formats one must deal with.***

## Different Serializations

- JSON, YAML, XML, CSV, Text

## Different Data Models and System

- Relational Databases (e.g., PostgreSQL, MySQL, Oracle, IBM DB2, MS SQL Server)
- Document Stores (e.g., CouchDB, MongoDB)
- Key-Value Stores (e.g., LevelDB, Oracle NoSQL Database)
- Column Stores (e.g., MariaDB)
- Triplestores & Graph Databases (e.g., Jena, Neo4j)

## Legacy Formats

- dBase
- MS Access
- Lotus Notes

# Why is Data Cleaning Time Consuming? (cont.)

***2) Many different quality issues (databases try to fit a complex world into a simple abstraction).***

***3) Cleaning the data requires domain knowledge***

| FirstName | LastName | BDate | Salary | Gender |
|---|---|---|---|---|
| Alice | Smith | 5/4/1978 | 45.000 | M |
| Bob | Smith | 4.5.1978 | 45k | - |
| Jonez | Claire | May 4, 1978 | $45000 | F |
| Dave C | Adams | | 32 | M |

Image source: https://www.lowdermilk-associates.com/wp-content/uploads/sites/2231/2018/07/Screen-Shot-2018-07-26-at-2.12.42-PM-e1532695190753.png

# Data Quality Issues

***2) Many different quality issues (databases try to fit a complex world into a simple abstraction).***

***3) Cleaning the data requires domain knowledge***

| FirstName | LastName | BDate | Salary | Gender |
|---|---|---|---|---|
| Alice | Smith | 5/4/1978 | 45.000 | M |
| Bob | Smith | 4.5.1978 | 45k | - |
| Jonez | Claire | May 4, 1978 | $45000 | F |
| Dave C | Adams | | 32 | M |

**Timeliness**: Is the data up-do-date?

Image source: https://www.lowdermilk-associates.com/wp-content/uploads/sites/2231/2018/07/Screen-Shot-2018-07-26-at-2.12.42-PM-e1532695190753.png

# Data Quality Issues (cont.)

***2) Many different quality issues (databases try to fit a complex world into a simple abstraction).***

***3) Cleaning the data requires domain knowledge***

| FirstName | LastName | BDate | Salary | Gender |
|---|---|---|---|---|
| Alice | Smith | 5/4/1978 | 45.000 | M |
| Bob | Smith | 4.5.1978 | 45k | - |
| Jonez | Claire | May 4, 1978 | $45000 | F |
| Dave C | Adams | | 32 | M |

**Syntactical Consistency**: Are all values (in a column) represented in the same format?

Image source: https://www.lowdermilk-associates.com/wp-content/uploads/sites/2231/2018/07/Screen-Shot-2018-07-26-at-2.12.42-PM-e1532695190753.png

# Data Quality Issues (cont.)

***2) Many different quality issues (databases try to fit a complex world into a simple abstraction).***

***3) Cleaning the data requires domain knowledge***

| FirstName | LastName | BDate | Salary | Gender |
|---|---|---|---|---|
| Alice | Smith | 5/4/1978 | 45.000 | - |
| Bob | Smith | 4.5.1978 | 45k | M |
| Jonez | Claire | May 4, 1978 | $45000 | F |
| Dave C | Adams | | 32 | M |

**Uniqueness**: Is each entity represented only once?

Image source: https://www.lowdermilk-associates.com/wp-content/uploads/sites/2231/2018/07/Screen-Shot-2018-07-26-at-2.12.42-PM-e1532695190753.png

# Data Quality Issues (cont.)

***2) Many different quality issues (databases try to fit a complex world into a simple abstraction).***

***3) Cleaning the data requires domain knowledge***

| FirstName | LastName | BDate | Salary | Gender |
|---|---|---|---|---|
| Alice | Smith | 5/4/1978 | 45.000 | M |
| Bob | Smith | 4.5.1978 | 45k | - |
| Jonez | Claire | May 4, 2012 | $45000 | F |
| Dave C | Adams | | 32 | M |
| Emily | Brown | 11/5/1971 | 55000 | F |

**Completeness**: Is the data set a compete representation of the universe of discourse?

Image source: https://www.lowdermilk-associates.com/wp-content/uploads/sites/2231/2018/07/Screen-Shot-2018-07-26-at-2.12.42-PM-e1532695190753.png

# Data Quality Issues (cont.)

***2) Many different quality issues (databases try to fit a complex world into a simple abstraction).***

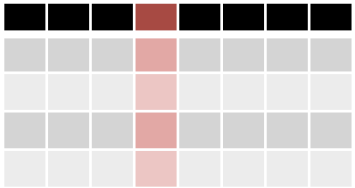
***3) Cleaning the data requires domain knowledge***

| FirstName | LastName | BDate | Salary | Gender |
|---|---|---|---|---|
| Alice | Smith | 5/4/1978 | 45.000 | M |
| Bob | Smith | 4.5.1978 | 45k | - |
| Jonez | Claire | May 4, 1978 | $45000 | F |
| Dave C | Adams | | 32 | M |

**Consistency**: Are all business rules (constraints) satisfied?

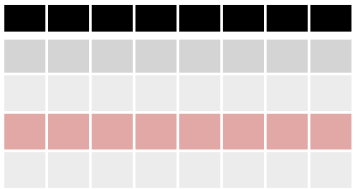
Image source: https://www.lowdermilk-associates.com/wp-content/uploads/sites/2231/2018/07/Screen-Shot-2018-07-26-at-2.12.42-PM-e1532695190753.png

# Classification of Data Quality Issues



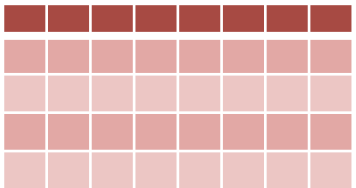
## Column

*Illegal Values, Missing Values (encodings of NULL), Value Representation*



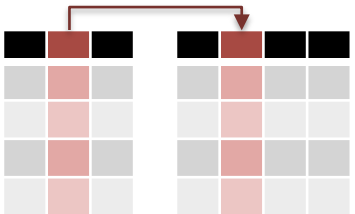
## Record

*Violation of attribute dependencies.*



## Record Type

*Uniqueness Violations (Functional Dependency Violations), Conflicting Values, Missing Records.*



## Source

*Referential Integrity Violation.*

See for example [Rahm, Do. 2000]

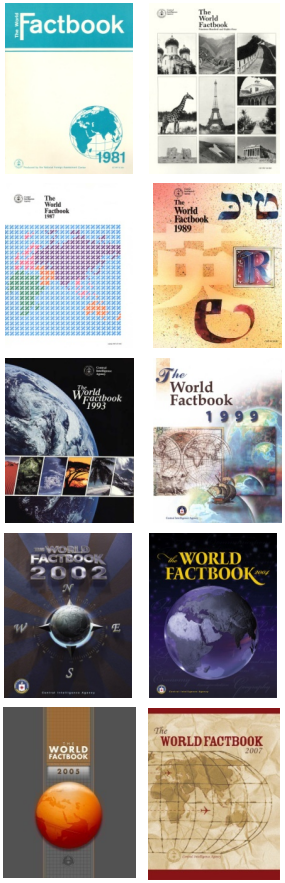
# Why is Data Cleaning Time Consuming? *(cont.)*

***1) One of the most 'annoying' parts of data analysis is the variety of data formats one must deal with.***

# CIA World Factbook

```
------------------------------------------------------------------------
                        THE WORLD FACTBOOK 1990
------------------------------------------------------------------------

Country:  Afghanistan
- Geography
Total area: 647,500 km2; land area: 647,500 km2

Comparative area: slightly smaller than Texas

Land boundaries: 5,826 km total; China 76 km, Iran 936 km,
Pakistan 2,430 km, USSR 2,384 km

Coastline: none--landlocked

Maritime claims: none--landlocked

Disputes: Pashtun question with Pakistan; Baloch question with Iran
and Pakistan; periodic disputes with Iran over Helmand water rights;
insurgency with Iranian and Pakistani involvement; traditional tribal
rivalries

Climate: arid to semiarid; cold winters and hot summers

Terrain: mostly rugged mountains; plains in north and southwest

Natural resources: natural gas, crude oil, coal, copper, talc, barites,
sulphur, lead, zinc, iron ore, salt, precious and semiprecious stones

Land use: 12% arable land; NEGL% permanent crops; 46% meadows and
pastures; 3% forest and woodland; 39% other; includes NEGL% irrigated

Environment: damaging earthquakes occur in Hindu Kush mountains;
soil degradation, desertification, overgrazing, deforestation, pollution

Note: landlocked
```

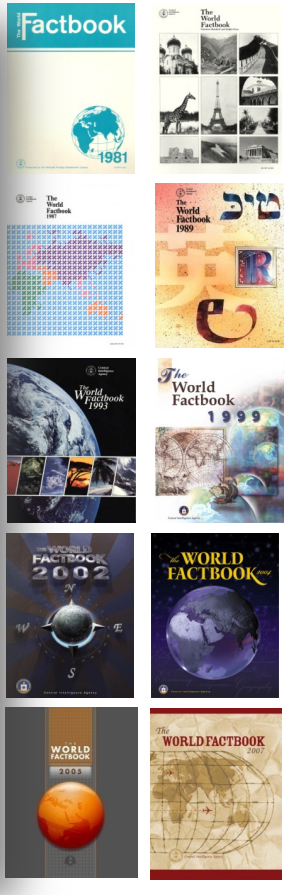
1990



# CIA World Factbook

```
Title          :Afghanistan
Text           :
                                    Afghanistan

                                     Geography

Location:
    Southern Asia, north of Pakistan
Map references:
    Asia
Area:
  total area:
    647,500 sq km
  land area:
    647,500 sq km
  comparative area:
    slightly smaller than Texas
Land boundaries:
    total 5,529 km, China 76 km, Iran 936 km, Pakistan 2,430 km, Tajikistan
    1,206 km, Turkmenistan 744 km, Uzbekistan 137 km
Coastline:
    0 km (landlocked)
Maritime claims:
    none; landlocked
International disputes:
    periodic disputes with Iran over Helmand water rights; Iran supports clients

    in country, private Pakistani and Saudi sources also are active; power
    struggles among various groups for control of Kabul, regional rivalries
    among emerging warlords, traditional tribal disputes continue; support to
    Islamic fighters in Tajikistan's civil war; border dispute with Pakistan
    (Durand Line); support to Islamic militants worldwide by some factions
Climate:
    arid to semiarid; cold winters and hot summers
```

1995



# CIA World Factbook

## Geography — Afghanistan

Top of Page

**Location:**
Southern Asia, north and west of Pakistan, east of Iran

**Geographic coordinates:**
33 00 N, 65 00 E

**Map references:**
Asia

**Area:**
*total:* 647,500 sq km
*water:* 0 sq km
*land:* 647,500 sq km

**Area - comparative:**
slightly smaller than Texas

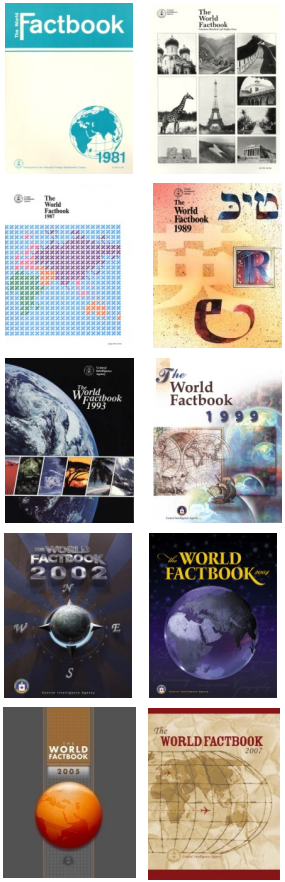
**Land boundaries:**
*total:* 5,529 km
*border countries:* China 76 km, Iran 936 km, Pakistan 2,430 km, Tajikistan 1,206 km, Turkmenistan 744 km, Uzbekistan 137 km

**Coastline:**
0 km (landlocked)

**Maritime claims:**
none (landlocked)

**Climate:**
arid to semiarid; cold winters and hot summers

2005




See [https://www.cia.gov/library/publications/the-world-factbook/]

# Why is Data Cleaning Time Consuming? *(cont.)*

***1) One of the most 'annoying' parts of data analysis is the variety of data formats one must deal with.***

***2) Many different quality issues (databases try to fit a complex world into a simple abstraction).***

***3) Cleaning the data requires domain knowledge***

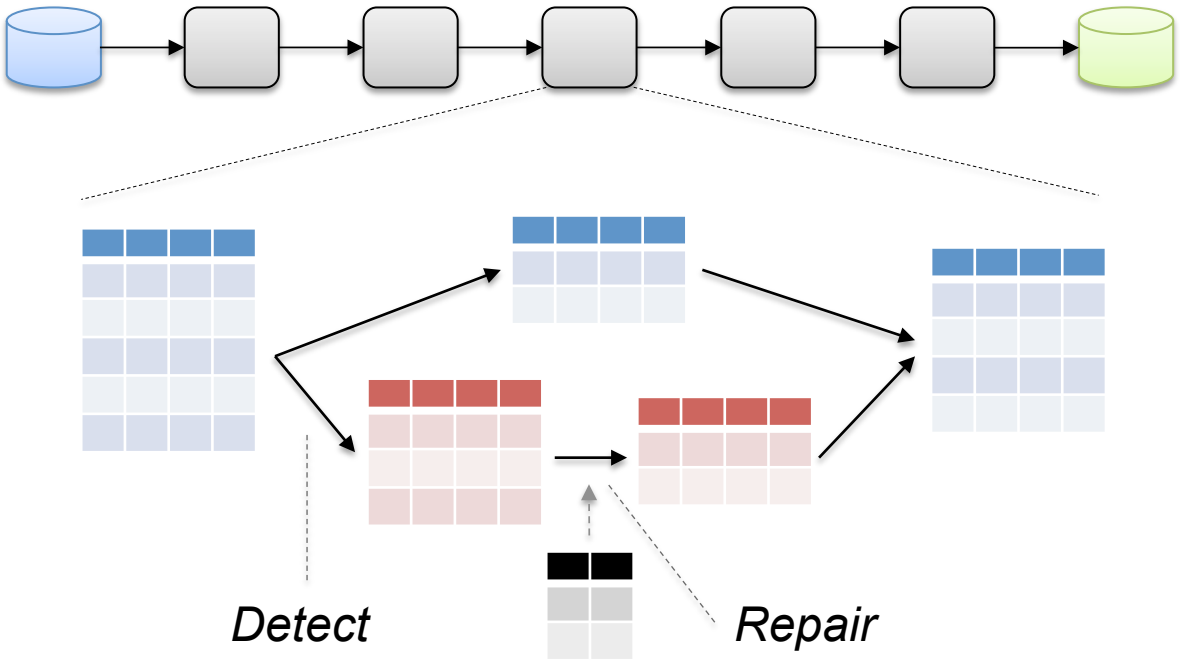
***4) Big Data***

**Volume** of data requires automated methods that encode what high quality data looks like.

**Variety** and heterogeneity of data sources requires specific scripts for individual sources.

# Data Cleaning Workflow

*The data cleaning workflow is a sequence of steps. Each step addresses one quality issue. Steps are divided into **detect** and **repair**.*



# Things to Keep in Mind

## *Purpose*

What is the intended use of the data?

## *Domain Knowledge*

*Learn as much as possible about data (generation process)*

## *Attention*

*Be mindful and pay attention to details*

**If you don't need it, don't clean it!**

NYU

# Cleaning Tabular Data

***Data Quality Problems and Quality Assessment***

***Detect and Repair***

***Examples from NYC Open Data***

# *Data Exploration & Data Profiling*

Get to know your data

NYC OpenData

# 311 Service Requests for 2005

*Data profiling refers to the activity of creating small but informative summaries of a database.*

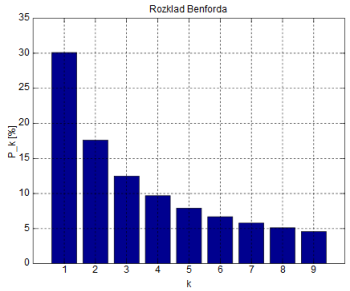
Ted Johnson, Encyclopedia of Database Systems

| Category | Task | Description |
|---|---|---|
| **Cardinalities** | num-rows | Number of rows |
| | value length | Measurements of value lengths (min, max, median, and average) |
| | null values | Number or percentage of null values |
| | distinct | Number of distinct values; aka "cardinality" |
| | uniqueness | Number of distinct values divided by number of rows |
| **Value distributions** | histogram | Frequency histograms (equi-width, equi-depth, etc.) |
| | constancy | Frequency of most frequent value divided by number of rows |
| | quartiles | Three points that divide the (numeric) values into four equal groups |
| | soundex | Distribution of soundex codes |
| | first digit | Distribution of first digit in numeric values (Benford's law) |
| **Patterns, data types, and domains** | basic type | Generic data type: numeric, alphabetic, date, time |
| | data type | Concrete DBMS-specific data type: varchar, timestamp, etc. |
| | decimals | Maximum number of decimal places in numeric values |
| | precision | Maximum number of digits in numeric values |
| | patterns | Histogram of value patterns (Aa9…) |
| | data class | Semantic, generic data type: code, indicator, text, date/time, quantity, identifier, etc. |
| | domain | Classification of semantic domain: credit card, first name, city, phenotype, etc. |

# Benford's Law

**Benford's law**, also called the law of anomalous numbers, is an observation about the frequency distribution of leading digits in many real-life sets of numerical data.

The law states that the leading significant digit is likely to be small. That is, the number 1 appears as the leading significant digit about 30% of the time, while 9 appears as the leading significant digit less than 5% of the time.

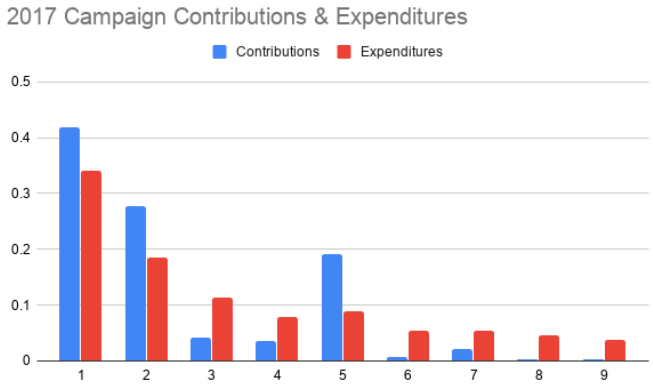



https://en.wikipedia.org/wiki/Benford%27s_law

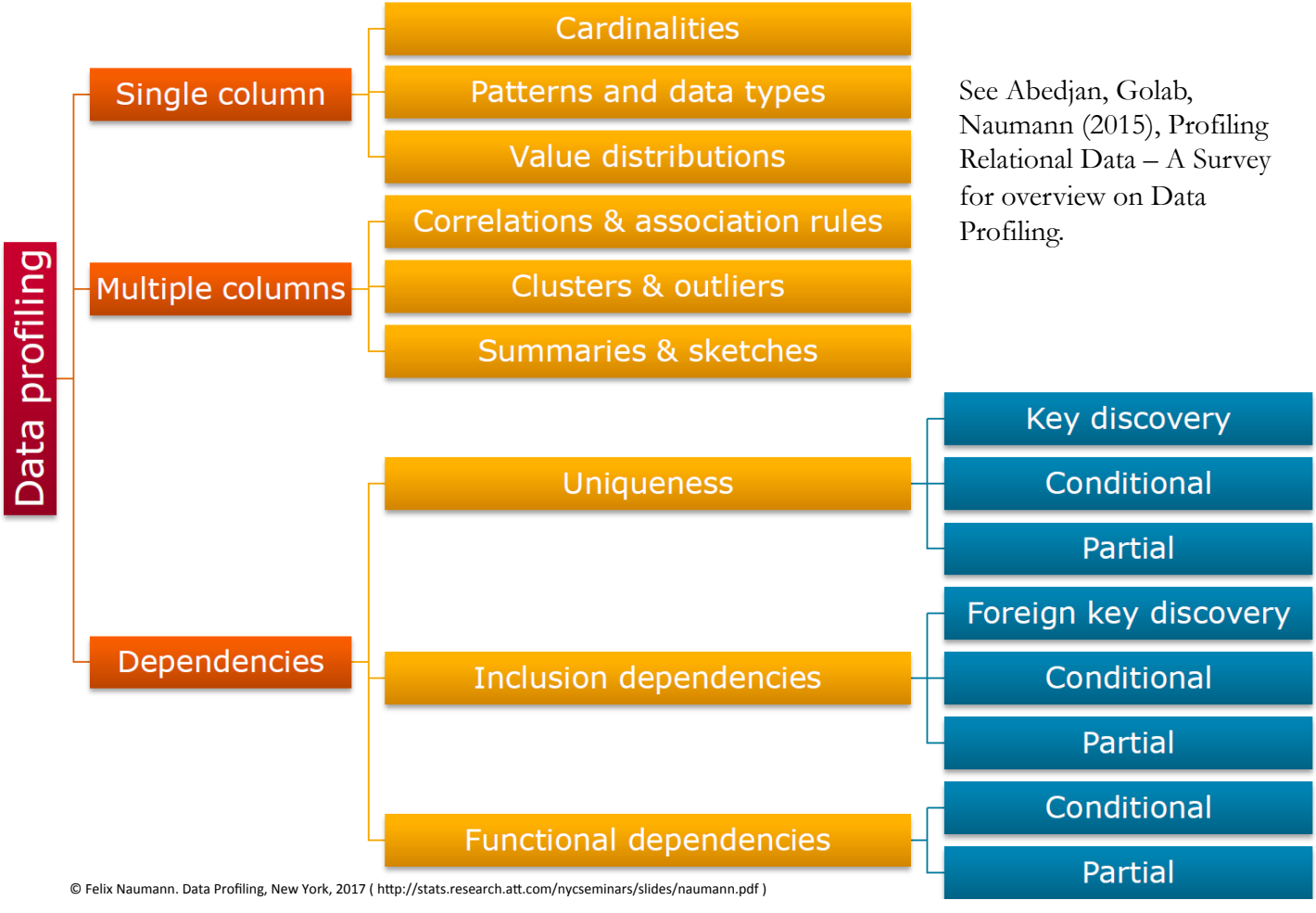
## 2019 Campaign Contributions
## 2019 Campaign Expenditures

Count frequency of first digit for values in attribute *amnt* (amount) in the 2017 Campaign Contributions and Expenditures tables.

```
SELECT fd, COUNT(*)
FROM (
  SELECT LEFT(CAST(amnt AS VARCHAR), 1) AS fd
  FROM ds_s9d3_x4fz WHERE amnt >= 0
) q GROUP BY fd ORDER BY fd;
```



# Classification of Data Profiling Tasks




See Abedjan, Golab, Naumann (2015), Profiling Relational Data – A Survey for overview on Data Profiling.

# Use Cases for Data Profiling

## *Query optimization*
Counts and histograms, functional dependencies, ...

## *Data cleaning*
Patterns, rules, and violations

## *Data integration*
Cross-DB inclusion dependencies

## *Data analytics and mining*
Profiling as preparation to decide on models and questions
Discover datasets that meet certain requirements

## *Database reverse engineering*

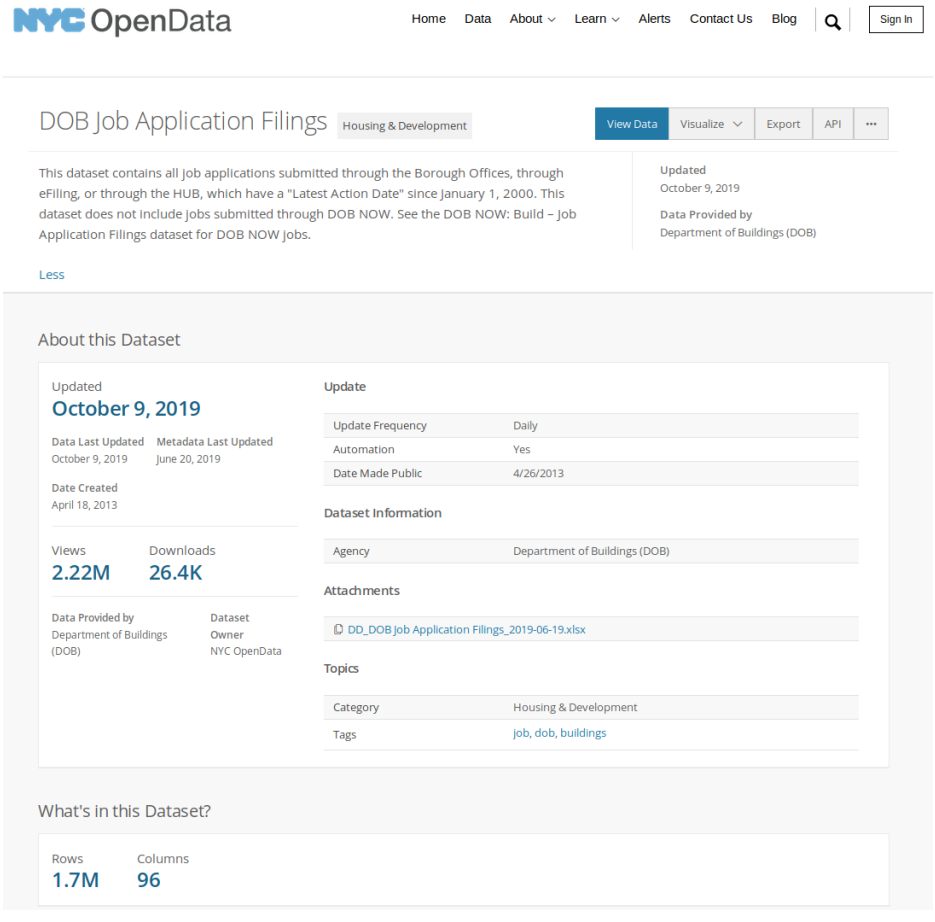
See [Abedjan, Golab, Naumann. 2015] for overview on Data Profiling.

# *Simple Steps*

## Sorting and Cardinalities

# DOB Job Application Filings (ic3t-wcy2)

https://data.cityofnewyork.us/Housing-Development/DOB-Job-Application-Filings/ic3t-wcy2



NYC OpenData

Home Data About Learn Alerts Contact Us Blog Sign In

## DOB Job Application Filings

Housing & Development

View Data | Visualize | Export | API | ...

This dataset contains all job applications submitted through the Borough Offices, through eFiling, or through the HUB, which have a "Latest Action Date" since January 1, 2000. This dataset does not include jobs submitted through DOB NOW. See the DOB NOW: Build – Job Application Filings dataset for DOB NOW jobs.

Less

**Updated**
October 9, 2019

**Data Provided by**
Department of Buildings (DOB)

### About this Dataset

Updated
**October 9, 2019**

Data Last Updated
October 9, 2019

Metadata Last Updated
June 20, 2019

Date Created
April 18, 2013

Views
**2.22M**

Downloads
**26.4K**

Data Provided by
Department of Buildings (DOB)

Dataset Owner
NYC OpenData

**Update**

| | |
|---|---|
| Update Frequency | Daily |
| Automation | Yes |
| Date Made Public | 4/26/2013 |

**Dataset Information**

| | |
|---|---|
| Agency | Department of Buildings (DOB) |

**Attachments**

DD_DOB Job Application Filings_2019-06-19.xlsx

**Topics**

| | |
|---|---|
| Category | Housing & Development |
| Tags | job, dob, buildings |

### What's in this Dataset?

Rows
**1.7M**

Columns
**96**

NYU

# DOB Job Ap

https://data.cityofnewyork.us/Housing-Development/DOB-Job-Application-Filings/i

NYC OpenData

Home Data About Learn Alerts Contact U

## DOB Job Application Filings

Housing & Development

View Data Visualize

This dataset contains all Job applications submitted through the Borough Offices, through eFiling, or through the HUB, which have a "Latest Action Date" since January 1, 2000. This dataset does not include jobs submitted through DOB NOW. See the DOB NOW: Build – Job Application Filings dataset for DOB NOW jobs.

Updated
October 9, 2019

Data Provided by
Department of Bu

Less

### About this Dataset

Updated
**October 9, 2019**

Data Last Updated
October 9, 2019

Metadata Last Updated
June 20, 2019

Date Created
April 18, 2013

Views
**2.22M**

Downloads
**26.4K**

Data Provided by
Department of Buildings (DOB)

Dataset Owner
NYC OpenData

**Update**

| | |
|---|---|
| Update Frequency | Daily |
| Automation | Yes |
| Date Made Public | 4/26/2013 |

**Dataset Information**

| | |
|---|---|
| Agency | Department of Buildings (DOB) |

**Attachments**

DD_DOB Job Application Filings_2019-06-19.xlsx

**Topics**

| | |
|---|---|
| Category | Housing & Development |
| Tags | job, dob, buildings |

### What's in this Dataset?

Rows
**1.7M**

Columns
**96**

| | |
|---|---|
| Owner's First Name | First Name of property owner |
| Owner's Last Name | Last Name of property owner |
| Owner's Business Name | Business Name of Property Owner |
| Owner's House Number | House Number of Property Owner |
| Owner'sHouse Street Name | House Street Name of Property Owner |
| City | City |
| State | State |
| Zip | Zip |
| Owner'sPhone # | Owner's Phone # |
| Job Description | Job Description |
| DOBRunDate | Date when query is run and pushed to Open Data. |
| JOB_S1_NO | JOB_S1_NO |
| TOTAL_CONSTRUCTION_FLOOR_AREA | Total Construction Floor Area |
| WITHDRAWAL_FLAG | Withdrawal Indicator |
| SIGNOFF_DATE | Sign-off Date |
| SPECIAL_ACTION_STATUS | Special Action Status |
| SPECIAL_ACTION_DATE | Special Action Date |
| BUILDING_CLASS | Building Class |
| JOB_NO_GOOD_COUNT | Job No Good Count |
| GIS_LATITUDE | Latitude |
| GIS_LONGITUDE | Longitude |

# DOB Job Application Fillings

**Distinct values** for attribute *business name (of property owner)* sorted in **ascending** order.

```
SELECT DISTINCT(owner_s_business_name) FROM ds_ic3t_wcy2
ORDER BY owner_s_business_name LIMIT 50;
```

```
      owner_s_business_name
---------------------------------
 `
 =
 ========================
 ============================
 ==============================
 ================================

 _
 __________
 -
 --
 ---
 ----
 -----
 ------
 -------
 ----------
 --------------
 --------------------
 ------------------------
 --------------------------------
 -.
 ,
 ??

(27 more)
```

# DOB Job Application Fillings

**Distinct values** for attribute *business name (of property owner)* sorted in **ascending** order.

```
SELECT DISTINCT(owner_s_business_name) FROM ds_ic3t_wcy2
ORDER BY owner_s_business_name LIMIT 50;
```

```
 /
 /////////
 //////////
 //////////
 ///////////////
 .
 ..
 ...
 ....
 *
 *********
 *****************
 **********************
 ***********************
 *************************
 *************************
 ******************************
 *******************************
 ********************************
 ++++++++++++++++++++++++++
 0
 000
 0000
 000000
 000 ENTERPRISE
 0010 WHITE STREET CORP
 0059 OWNERS CORP C/O HALSTEAD
(50 rows)
```

# DOB Job Application Fillings

**Distinct values** for attribute *business name (of property owner)* sorted in **descending** order.

```
SELECT DISTINCT(owner_s_business_name) FROM ds_ic3t_wcy2
ORDER BY owner_s_business_name DESC LIMIT 25;
```

```
 owner_s_business_name
--------------------------------

 ZZZ HOME HOLDING LLC
 ZZZ CARPENTRY INC.
 Z & Z Spectacular Creation inc.
 ZZSI CORP.
 Z.Z.''S FOOD COURT
 Z. ZINDEL INC
 Z. Zindel Inc.
 Z&Z GROUP INC.
 Z & Z EXPRESS INC
 ZZELLENT, LLC
 Z & Z Development LLC
 Z&Z888 PROPERTY LLC
 Z & Y REALTY MANAGEMENT LLC
 ZYQ LLC
 ZYP, INC.
 ZYP, INC
 ZYP INC
 ZYP Inc.
 ZYP Inc
 Z.Y.P.
 ZYP
 Z YOUNG INC
 Z&Y NYC INC.
 ZY MANAGEMENT LLC
(25 rows)
```

# DOB Job Application Fillings

**Distinct values** for attribute *city* sorted in **ascending** order.

```
SELECT DISTINCT(city) FROM ds_ic3t_wcy2 ORDER BY city LIMIT 25;
```

```
 city
----------------
 ---
 ///////
 .
 ...
 0000
 00000
 0ZONE PARK
 1
 100
 10010
 10012
 10013
 10016
 10017
 10021
 10022
 10023New York
 10025
 10036
 10036NY
 10153
 10314
 1041 Third Ave
 10452
 10462
(25 rows)
```

# DOB Job Application Fillings

**Distinct values** for attribute *city* sorted in **descending** order.

```
SELECT DISTINCT(city) FROM ds_ic3t_wcy2 ORDER BY city DESC LIMIT 25;
```

```
 city
-----------------

 ZURICH
 ZONE PARK
 ZONE 1LONG ISLA
 ZIONSVILLE
 ZCOLLEGE POINT
 YSTATEN ISLAND
 YPNKERS
 YOURBA LINDA
 YOUNKERS
 YOUNGSTOWN
 YORNKERS
 YORKTOWN HTS
 Yorktown Hts.
 Yorktown Hts
 YORKTOWN HT
 YORK TOWN HIGHT
 YORKTOWN HGTS.
 YORKTOWN HGHTS
 YORKTOWN HEIGTS
 YORKTOWN HEIGHT
 Yorktown Height
 yorktown height
 YORK TOWN HEIGH
 YORKTOWN
(25 rows)
```

# DOB Job Application Fillings

**Distinct values** for attribute *US state* sorted in ascending order.

```
SELECT DISTINCT(state) FROM ds_ic3t_wcy2 ORDER BY state;
```

```
 state
-------
 AK
 AL
 AR
 AZ
 CA
 CN
 CO
 CT
 DC
 DE
 FL
 FQ
 GA
 HI
 IA
 ID
 IL
 IN
 KS
 KY
 LA
 MA
 MD
 ME
 MI
 MN
 MO
 MS
 MT
 NC
 ND
 NE
 NH
 NJ
 NM
 NV
 NY
 OH
 OK
 ON
 OR
 PA
 PR
 RI
 SC
 SD
 sw
 TN
 TX
 UT
 VA
 VT
 WA
 WI
 WV
 WY

(57 rows)
```

# DOB Job Application Fillings

**Frequency** of distinct values for attribute *US state* sorted in ascending order.

```
SELECT DISTINCT(state), COUNT(*) FROM ds_ic3t_wcy2 GROUP BY state ORDER BY state;
```

| state | count |
|---|---|
| AK | 13 |
| AL | 12 |
| AR | 42 |
| AZ | 201 |
| CA | 2704 |
| CN | 9 |
| CO | 241 |
| CT | 2722 |
| DC | 320 |
| DE | 76 |
| FL | 2509 |
| FQ | 1 |
| GA | 509 |
| HI | 29 |
| IA | 45 |
| ID | 3 |
| IL | 1932 |
| IN | 61 |
| KS | 109 |
| KY | 72 |
| LA | 23 |
| MA | 1141 |
| MD | 892 |
| ME | 31 |
| MI | 269 |
| MN | 259 |
| MO | 96 |
| MS | 3 |
| MT | 2 |
| NC | 1086 |
| ND | 10 |
| NE | 24 |
| NH | 118 |
| NJ | 26603 |
| NM | 72 |
| NV | 304 |
| NY | 1649491 |
| OH | 1076 |
| OK | 18 |
| ON | 7 |
| OR | 15 |
| PA | 1649 |
| PR | 5 |
| RI | 407 |
| SC | 147 |
| SD | 13 |
| sw | 2 |
| TN | 245 |
| TX | 729 |
| UT | 158 |
| VA | 1016 |
| VT | 62 |
| WA | 224 |
| WI | 96 |
| WV | 3 |
| WY | 13 |
| | 60 |

(57 rows)

# *Use master data as reference for correct values*

https://simple.wikipedia.org/wiki/List_of_U.S._states

List [ change | change source ]

States of the United States of America

| Name | postal abbreviation[1] | Cities | | Established[upper-alpha 1] | Population [upper-alpha 2][3] | Total area[4] | |
|---|---|---|---|---|---|---|---|
| | | Capital | Largest[5] | | | mi² | km² |
| Alabama | AL | Montgomery | Birmingham | Dec 14, 1819 | 4,874,747 | 52,420 | 135,767 |
| Alaska | AK | Juneau | Anchorage | Jan 3, 1959 | 739,795 | 665,384 | 1,723,337 |
| Arizona | AZ | Phoenix | | Feb 14, 1912 | 7,016,270 | 113,990 | 295,234 |
| Arkansas | AR | Little Rock | | Jun 15, 1836 | 3,004,279 | 53,179 | 137,732 |
| California | CA | Sacramento | Los Angeles | Sep 9, 1850 | 39,536,653 | 163,695 | 423,967 |
| Colorado | CO | Denver | | Aug 1, 1876 | 5,607,154 | 104,094 | 269,601 |
| Connecticut | CT | Hartford | Bridgeport | Jan 9, 1788 | 3,588,184 | 5,543 | 14,357 |
| Delaware | DE | Dover | Wilmington | Dec 7, 1787 | 961,939 | 2,489 | 6,446 |
| Florida | FL | Tallahassee | Jacksonville | Mar 3, 1845 | 20,984,400 | 65,758 | 170,312 |
| Georgia | GA | Atlanta | | Jan 2, 1788 | 10,429,379 | 59,425 | 153,910 |
| Hawaii | HI | Honolulu | | Aug 21, 1959 | 1,427,538 | 10,932 | 28,313 |
| Idaho | ID | Boise | | Jul 3, 1890 | 1,716,943 | 83,569 | 216,443 |
| Illinois | IL | Springfield | Chicago | Dec 3, 1818 | 12,802,023 | 57,914 | 149,995 |
| Indiana | IN | Indianapolis | | Dec 11, 1816 | 6,666,818 | 36,420 | 94,326 |
| Iowa | IA | Des Moines | | Dec 28, 1846 | 3,145,711 | 56,273 | 145,746 |
| Kansas | KS | Topeka | Wichita | Jan 29, 1861 | 2,913,123 | 82,278 | 213,100 |
| Kentucky[upper-alpha 3] | KY | Frankfort | Louisville | Jun 1, 1792 | 4,454,189 | 40,408 | 104,656 |
| Louisiana | LA | Baton Rouge | New Orleans | Apr 30, 1812 | 4,684,333 | 52,378 | 135,659 |
| Maine | ME | Augusta | Portland | Mar 15, 1820 | 1,335,907 | 35,380 | 91,633 |
| Maryland | MD | Annapolis | Baltimore | Apr 28, 1788 | 6,052,177 | 12,406 | 32,131 |
| Massachusetts[upper-alpha 3] | MA | Boston | | Feb 6, 1788 | 6,859,819 | 10,554 | 27,336 |
| Michigan | MI | Lansing | Detroit | Jan 26, 1837 | 9,962,311 | 96,714 | 250,487 |
| Minnesota | MN | St. Paul | Minneapolis | May 11, 1858 | 5,576,606 | 86,936 | 225,163 |
| Mississippi | MS | Jackson | | Dec 10, 1817 | 2,984,100 | 48,432 | 125,438 |
| Missouri | MO | Jefferson City | Kansas City | Aug 10, 1821 | 6,113,532 | 69,707 | 180,540 |
| Montana | MT | Helena | Billings | Nov 8, 1889 | 1,050,493 | 147,040 | 380,831 |
| Nebraska | NE | Lincoln | Omaha | Mar 1, 1867 | 1,920,076 | 77,348 | 200,330 |
| Nevada | NV | Carson City | Las Vegas | Oct 31, 1864 | 2,998,039 | 110,572 | 286,380 |
| New Hampshire | NH | Concord | Manchester | Jun 21, 1788 | 1,342,795 | 9,349 | 24,214 |
| New Jersey | NJ | Trenton | Newark | Dec 18, 1787 | 9,005,644 | 8,723 | 22,591 |
| New Mexico | NM | Santa Fe | Albuquerque | Jan 6, 1912 | 2,088,070 | 121,590 | 314,917 |
| New York | NY | Albany | New York | Jul 26, 1788 | 19,849,399 | 54,555 | 141,297 |
| North Carolina | NC | Raleigh | Charlotte | Nov 21, 1789 | 10,273,419 | 53,819 | 139,391 |
| North Dakota | ND | Bismarck | Fargo | Nov 2, 1889 | 755,393 | 70,698 | 183,108 |

| us_states |
|---|
| abbrev |
| AL |
| AK |
| AZ |
| AR |
| CA |
| CO |
| … |

# DOB Job Application Fillings

**Distinct values** for attribute *US state* that are **not included in the master data**.

```
SELECT DISTINCT(state) FROM ds_ic3t_wcy2
EXCEPT
SELECT abbrev FROM us_states;
```

```
 state
-------

 PR
 CN
 DC
 FQ
 ON
 sw
(7 rows)
```

# *Data Repair*

Modify the Data to eliminate Quality Flaws

# DOB Job Application Fillings

City and ZIP code for records where *US state* is **FQ**.

```
SELECT city, zip FROM ds_ic3t_wcy2 WHERE state = 'FQ';
```

```
 city           |   zip
----------------+----------
 Kearny NXECMBF | b\ZZZZZ\
(1 row)
```

City and ZIP code for records where *US state* is **sw**.

```
SELECT city, zip FROM ds_ic3t_wcy2 WHERE state = 'sw';
```

```
 city      | zip
-----------+-----
 Stockholm |
 Stockholm |
(2 rows)
```

# DOB Job Application Fillings

City and ZIP code for records where *US state* is **ON**.

```
SELECT city, zip FROM ds_ic3t_wcy2 WHERE state = 'ON';
```

```
 city_     |  zip
-----------+-------
 THORNHILL | 11240
 THORNHILL | 11240
 QUEENS    | 11101
 VAUGHAN   | 09865
 VAUGHAN   | 09865
 VAUGHAN   | 04011
 VAUGHAN   | 11111
(7 rows)
```

City and ZIP code for records where *US state* is **CN**.

```
SELECT city, zip FROM ds_ic3t_wcy2 WHERE state = 'CN';
```

```
 city        |  zip
-------------+-------
 WILTON      | 06897
 NORWALK     | 06851
 HARTFORD    | 06103
 SHELTON     | 06484
 SHELTON     | 06484
 SHELTON     | 06484
 MT. ROYAL   | 00000
 RIDGEFIELD  | 06877
 RIDGEFIELD  | 06877
(9 rows)
```

https://www.zip-codes.com
https://m.usps.com/m/ZipLookupAction

## *Replace all incorrect spellings*

**Challenge**: Identify all incorrect spellings

How many different spellings of Brooklyn are there in the dataset?

- Use fuzzy matching (string similarity search)
  - Soundex
  - String Edit Distance (Levenshtein Distance)

# DOB Job Application Fillings

## City names that have **same '*sound index*' as BROOKLYN**.

```
SELECT DISTINCT(UPPER(city)) AS name
FROM ds_ic3t_wcy2
WHERE SOUNDEX(city) = SOUNDEX('BROOKLYN')
ORDER BY UPPER(city);
```

```
 name
-----------------
 B2ROOKLYN
 B4ROOKLYN
 BBBROOKLYN
 BBROOKLYN
 BERKELEY
 BERKELEY HEIGHT
 BERKELEY HTS
 BERKELEY HTS.
 BERKLEY
 BERKLEY HEIGHTS
 BERKLEY HTS.
 BEROOKLYN
 BFROOKLYN
 BOORKLY
 BORRKLYN
 BORSALOM
 BR00KLYN
 BR0OKLYN
 BR4OOKLYN
 BRAOOKLYN
 BREOOKLYN
 BRIACLIFF MANOR
 BRIIKLYN
 BRIOKLYN
 BRIOOKLYN
 BRKKLYN
 BRKLLYN
 BRKLN
 BRKLY
 BRKLYN
 …
 BROOKLYNTCHEN
 BROOKLYNY
 BROOKLYON
 BROOKLYTN
 BROOKLYU
 BROOKLYYN
 BROOKOLYN
 BROOKYL
 BROOKYLN
 BROOKYLYN
 BROOOKKLYN
 BROOOKLN
 BROOOKLYN
 BROOYKLN
 BRROKLY
 BRROKLYN
 BRROOKLYN
 BRRRKLYN
 BRUUKLYN
 BVROOKLYN
(172 rows)
```

Soundex is a phonetic algorithm for indexing names by sound, as pronounced in English, .. so that they can be matched despite minor differences in spelling. The Soundex code for a name consists of a letter (the first letter of the name) followed by three numerical digits that encode the remaining consonants.

https://en.wikipedia.org/wiki/Soundex

SELECT SOUNDEX('BROOKLYN') = B624
SELECT SOUNDEX('QUEENS') = Q520

**True Positive**: Misspelling that is identified as such.

**False Positive**: Not a misspelling that is falsely identified as one.

**True Negative**: Not a misspelling that is not identified as one.

**False Negative**: Misspelling that is not identified as one.

# DOB Job Application Fillings

City names that have **are similar spellings to BROOKLYN**.

```
SELECT DISTINCT(UPPER(city)) AS name FROM ds_ic3t_wcy2
WHERE LEVENSHTEIN (city, 'BROOKLYN', 1, 1, 1) < 3
ORDER BY UPPER(city);
```

```
 name
-----------------
 B4ROOKLYN
 BBBROOKLYN
 BBOOKLYN
 BBROOKLYN
 BDOOKLYN
 BEOKLYN
 BEOOKLYN
 BEOOKYN
 BEROOKLYN
 BFROOKLYN
 BKOOKLYN
 BKOOLYN
 BLOOKLYN
 BLOOKYN
 BNROOKLYN
 BOOKLYN
 BOOOKLYN
 BR00KLYN
 BR0OKLYN
 BR4OOKLYN
 BRAOOKLYN
 BREOOKLYN
 BRIIKLYN
 BRIOOKLYN
 …
```

```
 BROOKLEN
 BROOKLEY
 BROOKLIN
 BROOKLINE
 BROOKLING
 BROOKLIYN
 BROOKLKN
 BROOKLKY
 BROOKLKYN
 BROOKLLYN
 BROOKLN
 BROOKLNM
 BROOKLNY
 BROOKLNYN
 BROOKLOYN
 BROOKLRN
 BROOKLTN
 …
 BVROOKLYN
 CROOKLYN
 NBROOKLLYN
 NBROOKLYN
 NROOKLYN
 ROOKKYN
 ROOKLYN
 S BROOKLYN
(171 rows)
```

Edit Distance … the minimum number of single-character edits (insertions, deletions or substitutions) required to change one word into the other.

https://en.wikipedia.org/wiki/Levenshtein_distance

*levenshtein*('BROOKLYN', 'BROOYKLN') ?

```
BROOKLYN
||||    |
BROOYKLN
```

3 x Replace

```
BROO-KLYN
||||  ||  |
BROOYKL-N
```

1 x Insert
1 x Delete

## *Imputation is the process of replacing missing data with substituted values*

There have been many theories embraced by scientists to account for missing data but many of them introduce large amounts of bias.

**Delete rows**: "… *the most common means of dealing with missing data is deletion.*"

**Hot-deck**: "… *a missing value is imputed from a randomly selected similar record.*"

**Mean substitution**: *"…replace a missing value with the mean of that variable for all other cases.*"

**Regression**: "*A regression model is estimated to predict observed values of a variable based on other variables.*"

https://en.wikipedia.org/wiki/Imputation_(statistics)
F. Biesseman et al., "Deep" Learning for Missing Value Imputation in Tables with Non-Numerical Data, CIKM 2018.

# Semantics of NULL

***Sometimes different text representations of NULL encode different semantics***

**“unknown”**

There is a value, but I do not know it.

E.g.: Unknown date-of-birth

**“not applicable”**

There is no meaningful value.

E.g.: Spouse for singles

**“withheld”**

There is a value, but we are not authorized to see it.

E.g.: Private phone line

# *Frequency & Length Outliers*

Identify values with different frequency or value length

# DOB Job Application Fillings

Values in attribute phone number sorted by decreasing number of characters.

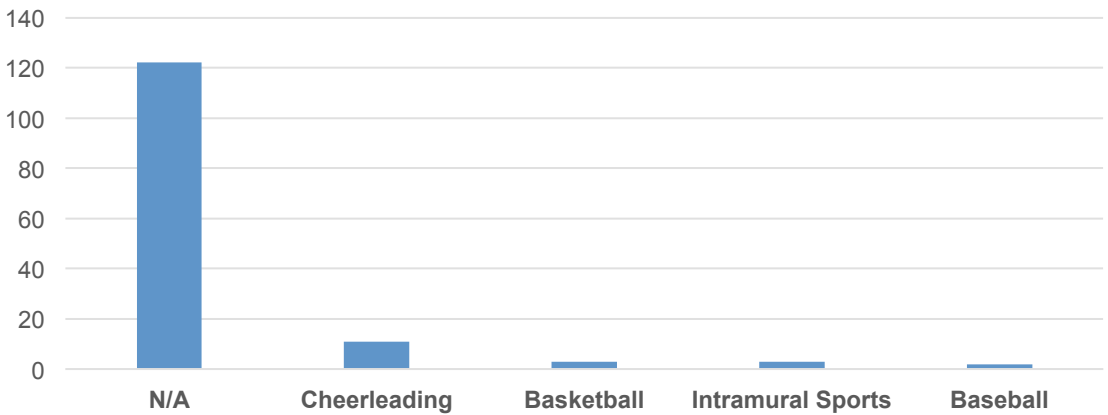
```
SELECT phone FROM ds_43nn_pn8j GROUP BY phone
ORDER BY LENGTH(phone) DESC LIMIT 20;
```

```
 phone
--------------

 914 255 1158
 718 322 8500
 607 227 5536
 718 898 1900
 19175617545
 90172017170
 7189392800
 7189497446
 2123589800
 7187888001
 7187889699
 9176985449
 2124756116
 5165224875
 3476491326
 7182720301
 2125875389
 3477036610
 7187227200
(20 rows)
```

# Frequency Outliers

## DOE High School Directory 2013-2014

### school_sports



# Frequency Outliers (cont.)

***Values that frequently occur as high frequency outliers***

Values that occur with frequency >50% in + 15,000 columns of NYC Open Data datasets (in Nov. 2016).

| | | |
|---|---|---|
| 0 | 262 | Columns |
| N/A | 71 | - ″ - |
| UNSPECIFIED | 67 | - ″ - |
| S | 57 | - ″ - |
| - | 50 | - ″ - |
| 0.00 | 47 | - ″ - |
| NY | 38 | - ″ - |
| 1 | 25 | - ″ - |
| 0.0 | 20 | - ″ - |
| IND | 12 | - ″ - |
| CLOSED | 10 | - ″ - |
| 100 | 8 | - ″ - |
| NOT AVAILABLE | 8 | - ″ - |
| 0 UNSPECIFIED | 6 | - ″ - |
| NONE | 5 | - ″ - |

# *Feature-based Outliers*

Use vector of value features to mine distance-based outliers

***Example features derived from column values***

**Value Frequency**

**Value length**

**Unique characters**

**Digits (%)**

**Letters (%)**

**Whitespaces (%)**

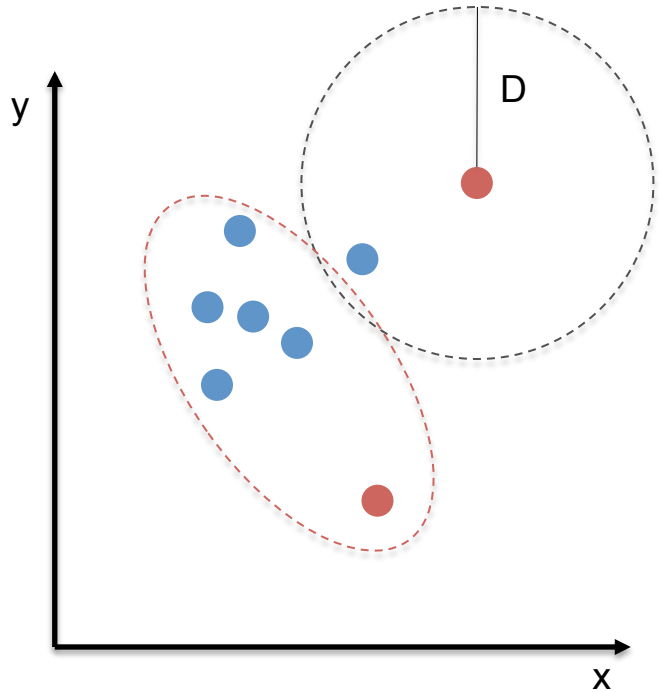
**Special Characters (%)**

**Character frequency**

Find outliers using the nested-loop algorithm described in [Knorr and Ng, VLDB 1998]. A value is considered an outlier if at least fraction **p** of the values in a column lies greater than distance **D** from the value.

## *Distance-based Outliers*

A value is considered an outlier if at least fraction **p** of the values in a column lies greater than distance **D** from the value.



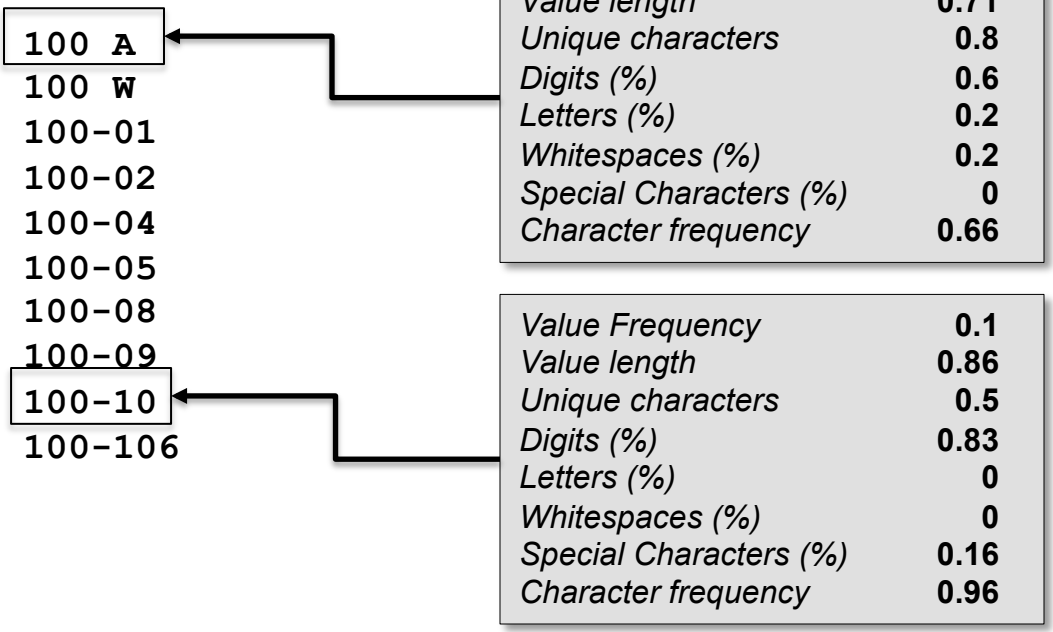
## DOB Permit Issuance

### owner_s_house_nr

```
100 A
100 W
100-01
100-02
100-04
100-05
100-08
100-09
100-10
100-106
```

## DOB Permit Issuance

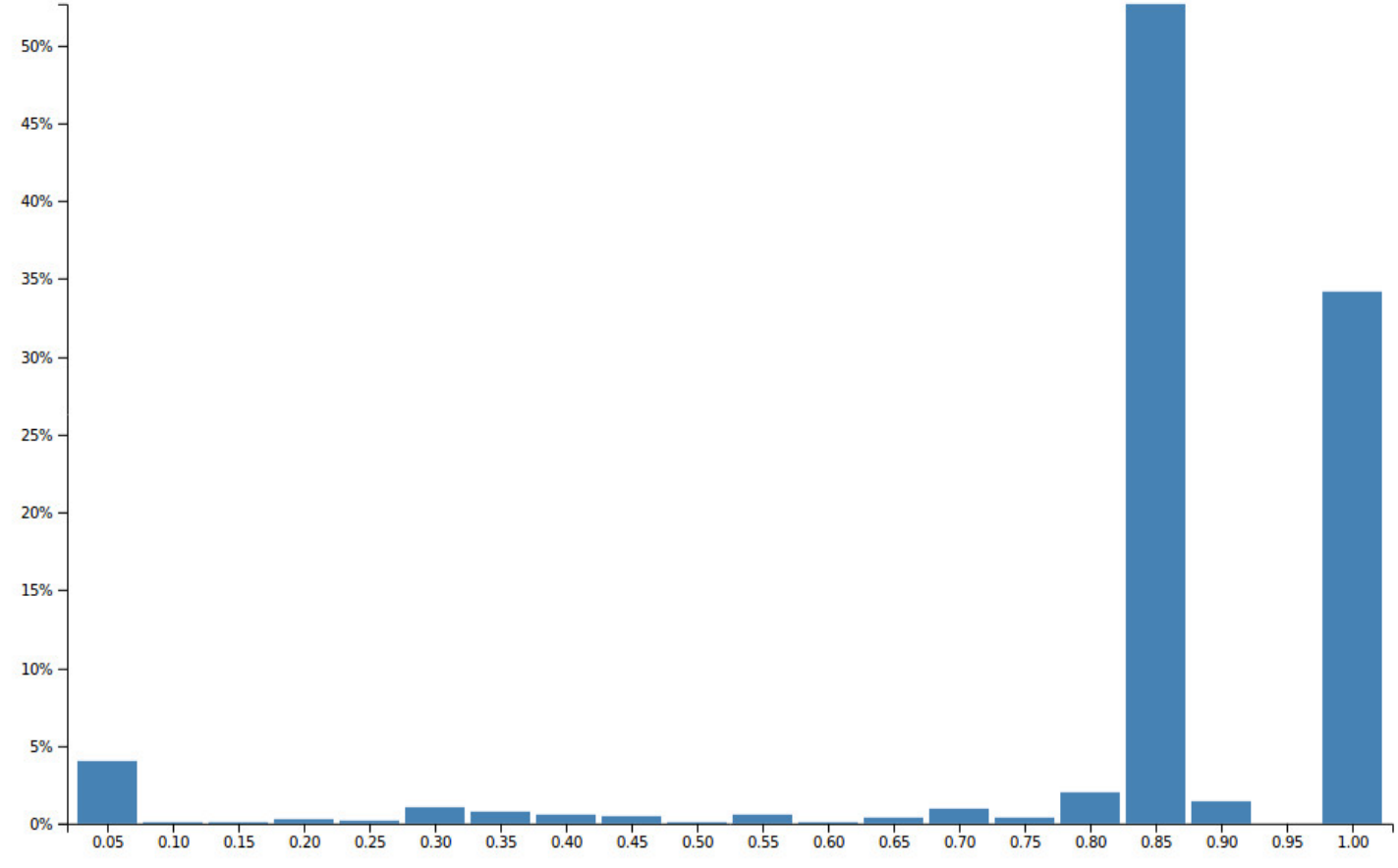
### owner_s_house_nr



# DOB Permit Issuance – owner_s_house_nr

Digits (%) 18374

Min: 0 731 Max: 1 6274



## *Distance-based Outliers for D = 0.4 and p = 0.75*

A value is considered an outlier if at least fraction **p** of the values in a column lies greater than distance **D** from the value.

| | |
|---|---|
| & SCHWAB | 4 |
| -C/O GOODMAN | 3 |
| 1 PEN PLAZA | 2 |
| 1 UNION SQ. | 4 |
| 11 NEW STREE | 2 |
| C/O | 1597 |
| C/O 1ST SVS | 1 |
| C/O A.S. | 2 |
| C/O ABC | 15 |
| C/O ABC PROP | 4 |
| N/A | 9 |
| NARROWS MGMT | 3 |
| NEWMARK | 1 |
| … | |

| | |
|---|---|
| NORTH MGMT. | 13 |
| NY REAL | 3 |
| OF JESUS | 1 |
| OF THE CHURC | 1 |
| One | 13 |
| P O BOX | 3 |
| P O BOX 313 | 7 |
| P,O. BOX | 1 |
| P. O. | 44 |
| WOODCREST | 2 |
| YELLOWSTONE | 2 |
| ZUCKERBROT | 6 |
| ___ | 1 |
| c/o Urban | 1 |

# *Different Representations*

Same set of entities represented using different sets of terms

**Frequency** of distinct values for attribute *borough*.

## Bureau of Fire Prevention - Certificates of Fitness

```
SELECT borough, COUNT(*) FROM ds_pdiy_9ae5
GROUP BY borough ORDER BY borough;
```

```
 borough | count
---------+-------
 BK      | 19699
 BX      | 14314
 MN      | 73381
 QN      | 16843
 SI      |  3726
         |   330
(6 rows)
```

## Historical DOB Permit Issuance

```
SELECT borough, COUNT(*) FROM ds_bty7_2jhb
GROUP BY borough ORDER BY borough;
```

```
 borough     |  count
---------------+---------
 BRONX         |  215035
 BROOKLYN      |  532384
 MANHATTAN     | 1008004
 QUEENS        |  517986
 STATEN ISLAND |  155117
(5 rows)
```

**Frequency** of distinct values for attribute *borough*.

## DOB Violations

```
 boro | count
------+--------
 `    |      3
 0    |      3
 1    | 808657
 2    | 239280
 3    | 610038
 4    | 365333
 5    |  57400
 B    |   1091
 M    |   1849
 Q    |    396
 S    |     44
      |      2
(12 rows)
```

## Bureau of Fire Prevention - Certificates of Fitness

```
 borough | count
---------+-------
 BK      | 19699
 BX      | 14314
 MN      | 73381
 QN      | 16843
 SI      |  3726
         |   330
(6 rows)
```

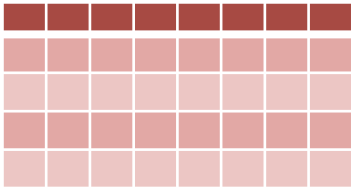
## Historical DOB Permit Issuance

```
 borough       |  count
---------------+---------
 BRONX         |  215035
 BROOKLYN      |  532384
 MANHATTAN     | 1008004
 QUEENS        |  517986
 STATEN ISLAND |  155117
(5 rows)
```

# *Integrity Constraints*

Constraint violations highlight quality issues

# Integrity Constraints



***Integrity constraints are a mechanism to define 'good' data***

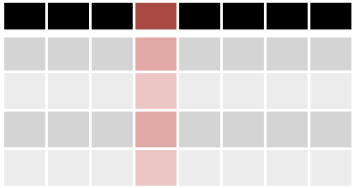
***A dataset that satisfies all constraints is of high quality***

Constraint violations are indicators of quality problems

**Challenge: Need a powerful language to express constraints**

# *Data Type Constraints*

Determine the type of each value in the attribute



## Column

*Illegal Values, Missing Values (encodings of NULL), Value Representation*

***We have an assumption about the valid values in a column, i.e., the domain***

**Data Type (INT, DECIMAL, TEXT, DATE)**

Defined at the schema level (if using a RDMBS)

***Simple test for INTEGER in Python***

```
def is_int(val):
  try:
    int(val)
    return True
  except ValueError:
    return False
```

# Data Type Outliers

FY03 – FY12 MMR Agency Performance Indicators

| agency |
| --- |
| 311 |
| ACS |
| BIC |
| BPL |
| CCHR |
| CCRB |
| CUNY |
| DCA |
| DCAS |
| DCLA |
| … |

Not all data type ‘outliers’ are quality flaws!

# Data Type Outliers


## Parking Events

| arrivaltime |
| --- |
| 2011-11-04T14:39:11 |
| 2011-11-04T14:39:13 |
| 2011-11-04T14:39:26 |
| 2012-03-11T02:00:11 |
| 2012-03-11T02:03:24 |
| 2012-03-11T02:05:04 |
| 2012-03-11T03:01:23 |
| 2012-17-45T34:99:82 |
| … |

```
public boolean isDate(String val) {
    String format = "yyyy-MM-dd'T'HH:mm:ss";
    SimpleDateFormat df;
    df = new SimpleDateFormat(format);
    try {
        df.parse(val);
        return true;
    } catch (java.text.ParseException ex) {
        return false;
    }
}
```

# Data Type Outliers

MELBOURNE DATA

## Parking Events

| arrivaltime |
| --- |
| 2011-11-04T14:39:11 |
| 2011-11-04T14:39:13 |
| 2011-11-04T14:39:26 |
| 2012-03-11T02:00:11 |
| 2012-03-11T02:03:24 |
| 2012-03-11T02:05:04 |
| 2012-03-11T03:01:23 |
| 2012-17-45T34:99:82 |
| … |

```
public boolean isDate(String val) {
    String format = "yyyy-MM-dd'T'HH:mm:ss";
    SimpleDateFormat df;
    df = new SimpleDateFormat(format);
    try {
        df.parse(val);
        return true;
    } catch (java.text.ParseException ex) {
        return false;
    }
}
```

2013-06-15T11:40:22

# Data Type Outliers

MELBOURNE DATA

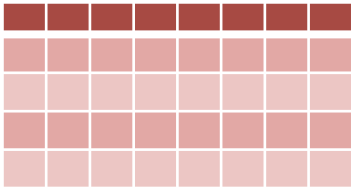
## Parking Events

| arrivaltime |
| --- |
| 2011-11-04T14:39:11 |
| 2011-11-04T14:39:13 |
| 2011-11-04T14:39:26 |
| **2012-03-11T02:00:11** |
| **2012-03-11T02:03:24** |
| **2012-03-11T02:05:04** |
| 2012-03-11T03:01:23 |
| **2012-17-45T34:99:82** |
| … |

```
public boolean isDate(String val) {
    String format = "yyyy-MM-dd'T'HH:mm:ss";
    SimpleDateFormat df;
    df = new SimpleDateFormat(format);
    try {
        Date date = df.parse(val);
        return df.format(date).equals(val);
    } catch (java.text.ParseException ex) {
        return false;
    }
}
```

# *Functional Dependencies*

Use Functional Dependencies for Data Cleaning



## *Use Functional Dependencies (FD) to detect and repair inconsistencies*

FD is a constraint that describes the relationship between sets of attributes in a relation

Detect pairs of records that violate FD

Repair FD violations by applying data modifications (delete record or modify values)

Use minimal number of modifications as measure for ‘good’ repair

## *Problem: FD’s are often unknown for a data set*

See [Papenbrock et al. 2016] for evaluation of FD discovery algorithms.

# Repairing Dependency Violations

## *Repair by deleting records*

Zip → City
Phone → Name, Street, Zip, City

| | Name | Street | Zip | City | Phone |
|---|---|---|---|---|---|
| $t_1$ | James Smith | 5th Ave | 10011 | Manhattan | 351-344-5671 |
| $t_2$ | Jane Smith | 6th Ave | 10012 | Manhattan | 351-244-4674 |
| $t_3$ | James L. Smith | 5th Ave | 10011 | NYC | +1 351.344.5671 |
| $t_4$ | Jane R. Smith | 6th Ave | 10012 | Manhattan | 351-244-4674 |
| $t_5$ | Rebecca Ryan | 2nd Ave | 10043 | Manhattan | 455-231-0872 |

See [Chomicki, Marcinkowski. 2005]

# Repairing Dependency Violations

## *Repair by deleting records*

**Zip → City**

Phone → Name, Street, Zip, City

| | Name | Street | Zip | City | Phone |
|---|---|---|---|---|---|
| $t_1$ | James Smith | 5th Ave | 10011 | **Manhattan** | 351-344-5671 |
| $t_2$ | Jane Smith | 6th Ave | 10012 | Manhattan | 351-244-4674 |
| $t_3$ | James L. Smith | 5th Ave | 10011 | **NYC** | +1 351.344.5671 |
| $t_4$ | Jane R. Smith | 6th Ave | 10012 | Manhattan | 351-244-4674 |
| $t_5$ | Rebecca Ryan | 2nd Ave | 10043 | Manhattan | 455-231-0872 |

See [Chomicki, Marcinkowski. 2005]

# Repairing Dependency Violations

## *Repair by deleting records*

Zip → City

**Phone → Name, Street, Zip, City**

| | Name | Street | Zip | City | Phone |
|---|---|---|---|---|---|
| $t_1$ | James Smith | 5th Ave | 10011 | Manhattan | 351-344-5671 |
| $t_2$ | **Jane Smith** | 6th Ave | 10012 | Manhattan | 351-244-4674 |
| $t_3$ | James L. Smith | 5th Ave | 10011 | NYC | +1 351.344.5671 |
| $t_4$ | **Jane R. Smith** | 6th Ave | 10012 | Manhattan | 351-244-4674 |
| $t_5$ | Rebecca Ryan | 2nd Ave | 10043 | Manhattan | 455-231-0872 |

See [Chomicki, Marcinkowski. 2005]

# Repairing Dependency Violations

## *Repair by deleting records*

Zip → City
Phone → Name, Street, Zip, City

| | Name | Street | Zip | City | Phone |
|---|---|---|---|---|---|
| $t_1$ | James Smith | 5th Ave | 10011 | Manhattan | 351-344-5671 |
| $t_2$ | Jane Smith | 6th Ave | 10012 | Manhattan | 351-244-4674 |
| $t_3$ | James L. Smith | 5th Ave | 10011 | NYC | +1 351.344.5671 |
| $t_4$ | Jane R. Smith | 6th Ave | 10012 | Manhattan | 351-244-4674 |
| $t_5$ | Rebecca Ryan | 2nd Ave | 10043 | Manhattan | 455-231-0872 |

See [Chomicki, Marcinkowski. 2005]

# Repairing Dependency Violations

## *Repair by deleting records*

Zip → City

Phone → Name, Street, Zip, City

| | Name | Street | Zip | City | Phone |
|---|---|---|---|---|---|
| $t_1$ | James Smith | 5th Ave | 10011 | Manhattan | 351-344-5671 |
| $t_2$ | Jane Smith | 6th Ave | 10012 | Manhattan | 351-244-4674 |
| $t_3$ | ~~James L. Smith~~ | ~~5th Ave~~ | ~~10011~~ | ~~NYC~~ | ~~+1 351.344.5671~~ |
| $t_4$ | ~~Jane R. Smith~~ | ~~6th Ave~~ | ~~10012~~ | ~~Manhattan~~ | ~~351-244-4674~~ |
| $t_5$ | Rebecca Ryan | 2nd Ave | 10043 | Manhattan | 455-231-0872 |

See [Chomicki, Marcinkowski. 2005]

# Repairing Dependency Violations

## *Repair by deleting records*

Zip → City

Phone → Name, Street, Zip, City

| | Name | Street | Zip | City | Phone |
|---|---|---|---|---|---|
| $t_1$ | ~~James Smith~~ | ~~5th Ave~~ | ~~10011~~ | ~~Manhattan~~ | ~~351-344-5671~~ |
| $t_2$ | Jane Smith | 6th Ave | 10012 | Manhattan | 351-244-4674 |
| $t_3$ | James L. Smith | 5th Ave | 10011 | NYC | +1 351.344.5671 |
| $t_4$ | ~~Jane R. Smith~~ | ~~6th Ave~~ | ~~10012~~ | ~~Manhattan~~ | ~~351-244-4674~~ |
| $t_5$ | Rebecca Ryan | 2nd Ave | 10043 | Manhattan | 455-231-0872 |

See [Chomicki, Marcinkowski. 2005]

# Repairing Dependency Violations (cont.)

## *Repair by value modification*

Zip → City
Phone → Name, Street, Zip, City

| | Name | Street | Zip | City | Phone |
|---|---|---|---|---|---|
| $t_1$ | James Smith | 5th Ave | 10011 | Manhattan | 351-344-5671 |
| $t_2$ | Jane Smith | 6th Ave | 10012 | Manhattan | 351-244-4674 |
| $t_3$ | James L. Smith | 5th Ave | 10011 | NYC | +1 351.344.5671 |
| $t_4$ | Jane R. Smith | 6th Ave | 10012 | Manhattan | 351-244-4674 |
| $t_5$ | Rebecca Ryan | 2nd Ave | 10043 | Manhattan | 455-231-0872 |

See [Bohannon et al. 2005]

# DOE High School Directory 2013-2014

## District Borough Number (DBN)

Every public school in New York City is assigned a unique “District Borough Number”, commonly referred to as its “DBN”.

http://teachnyc.net/assets/2017_Job_Search_Guide.pdf

DBN is a unique identifier for public schools in the dataset ?

**=> The number of unique values equals the number of rows.**

```
SELECT COUNT(*) AS rows, SELECT COUNT(DISTINCT dbn) AS dist_dbn
FROM ds_u553_m549;
```

```
 rows  | dist_dbn
-------+---------
   422 |      422
```

# Are there violations for FD **Postcode → City** ?

For each **post code** count the number of distinct **cities** they occur with in database records. If the number is greater than one the FD is violated.

```
SELECT postcode FROM ds_u553_m549
GROUP BY postcode HAVING COUNT(DISTINCT city) > 1;
```

```
 postcode
----------
    10468
    11106
```

Get the **post codes** that violate the FD together with the different city names that cause the violation.

```
SELECT DISTINCT t1.postcode, t1.city, t2.city
FROM ds_u553_m549 t1, ds_u553_m549 t2
WHERE t1.postcode = t2.postcode AND t1.city <> t2.city AND t1.city < t2.city;
```

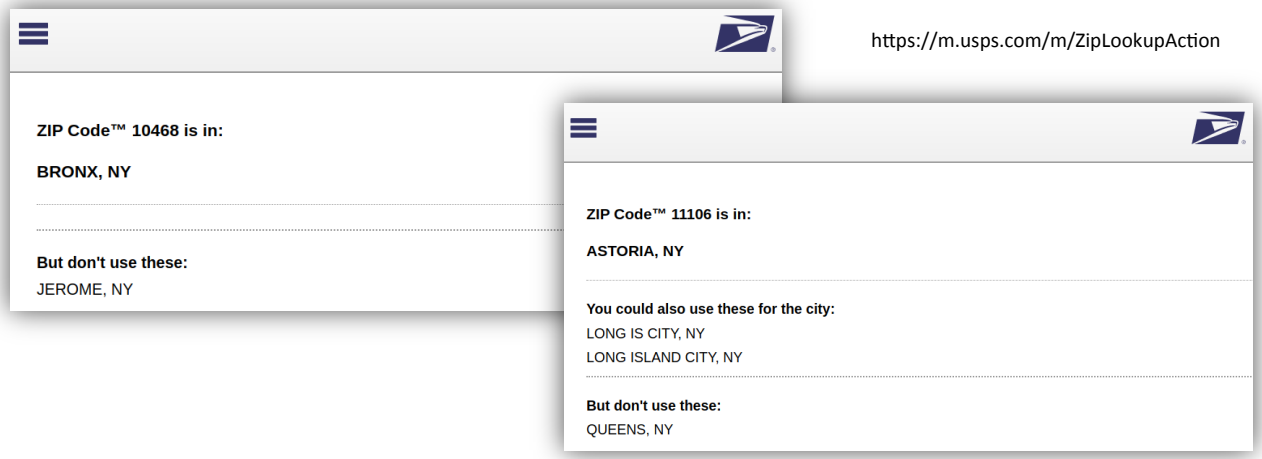
```
 postcode |  city   |       city
----------+---------+------------------
    11106 | Astoria | Long Island City
    10468 | Bronx   | South Bronx
(2 rows)
```

# DOE High School Directory 2013-2014

## Postcode → City ?

Use external knowledge bases to gather further information about the **post codes** and city names that violate the FD.

```
 postcode |  city    |       city
----------+---------+------------------
    11106 | Astoria | Long Island City
    10468 | Bronx   | South Bronx
(2 rows)
```

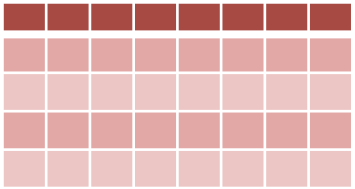



https://m.usps.com/m/ZipLookupAction

# *Duplicate Records*

Identify and Merge Different Representations of the same Entity

# Duplicate Detection



## Problem

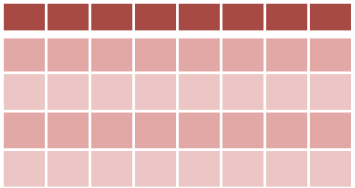
*Given one or more data sets, find all sets of records that represent the same real-world entity.*

## Main Difficulties

(1) Duplicates are not identical

(2) Large volume, cannot compare all pairs

# Duplicate Detection (cont.)



***Ironically "Duplicate Detection" has many Duplicates***

Record Linkage
Entity resolution
Object Identification
Doubles
Object Consolidation
Entity Clustering
Fuzzy Match
Approximate Match
Reference reconciliation
Reference matching
Merge/Purge

## *Decide if two records represent the same entity*

| Name | Street | Zip | City | Phone |
|---|---|---|---|---|
| James Smith | 5th Ave | 10011 | Manhattan | 351-344-5671 |
| Jane Smith | 6th Ave | 10011 | Manhattan | 351-244-4674 |
| Smith, J. | Fifth Avenue | 10011 | NYC | +1 351.344.5671 |

### Standardize data

### Apply different similarity measures

Similarity measures – Levenshtein, Soundex, Jaccard, etc.

Tokenize values

### Different weights for attributes

Two records with the same telephone number are more likely to be duplicates than records with the same (ZIP, City).

## *Need to compare all pairs of records*

**Sort table and use sliding window to reduce number of comparisons**

| Name | Street | Zip | City | Phone |
|---|---|---|---|---|
| James Smith | 5th Ave | 10011 | Manhattan | 351-344-5671 |
| Jane Smith | 6th Ave | 10011 | Manhattan | 351-244-4674 |
| Smith, J. | Fifth Avenue | 10011 | NYC | +1 351.344.5671 |
| L. Smith, James | 5th Ave | 10011 | Manhattan | 351-344-5671 |
| Rebecca Ryan | 2nd Avenue | 10043 | MN | 455-231-0872 |
| Peter Cox | Broadway | 10003 | Manhattan | 255-781-3280 |
| Jane R. Smith | 6th Ave | | Manhattan | 351.244.4674 |

See [Hernandez, Stolfo 1998]

## *Need to compare all pairs of records*

### Sort table and use sliding window to reduce number of comparisons

| Name | Street | Zip | City | Phone |
|---|---|---|---|---|
| James Smith | 5th Ave | 10011 | Manhattan | 351-344-5671 |
| Jane R. Smith | 6th Ave | | Manhattan | 351.244.4674 |
| Jane Smith | 6th Ave | 10011 | Manhattan | 351-244-4674 |
| L. Smith, James | 5th Ave | 10011 | Manhattan | 351-344-5671 |
| Peter Cox | Broadway | 10003 | Manhattan | 255-781-3280 |
| Rebecca Ryan | 2nd Avenue | 10043 | MN | 455-231-0872 |
| Smith, J. | Fifth Avenue | 10011 | NYC | 351.344.5671 |

See [Hernandez, Stolfo 1998]

## *Need to compare all pairs of records*

**Sort table and use sliding window to reduce number of comparisons**

| Name | Street | Zip | City | Phone |
|---|---|---|---|---|
| James Smith | 5th Ave | 10011 | Manhattan | 351-344-5671 |
| Jane R. Smith | 6th Ave | | Manhattan | 351.244.4674 |
| Jane Smith | 6th Ave | 10011 | Manhattan | 351-244-4674 |
| L. Smith, James | 5th Ave | 10011 | Manhattan | 351-344-5671 |
| Peter Cox | Broadway | 10003 | Manhattan | 255-781-3280 |
| Rebecca Ryan | 2nd Avenue | 10043 | MN | 455-231-0872 |
| Smith, J. | Fifth Avenue | 10011 | NYC | 351.344.5671 |

See [Hernandez, Stolfo 1998]

## *Need to compare all pairs of records*

**Sort table and use sliding window to reduce number of comparisons**

| Name | Street | Zip | City | Phone |
|---|---|---|---|---|
| James Smith | 5th Ave | 10011 | Manhattan | 351-344-5671 |
| Jane R. Smith | 6th Ave | | Manhattan | 351.244.4674 |
| Jane Smith | 6th Ave | 10011 | Manhattan | 351-244-4674 |
| L. Smith, James | 5th Ave | 10011 | Manhattan | 351-344-5671 |
| Peter Cox | Broadway | 10003 | Manhattan | 255-781-3280 |
| Rebecca Ryan | 2nd Avenue | 10043 | MN | 455-231-0872 |
| Smith, J. | Fifth Avenue | 10011 | NYC | 351.344.5671 |

See [Hernandez, Stolfo 1998]

## *Need to compare all pairs of records*

**Sort table and use sliding window to reduce number of comparisons**

| Name | Street | Zip | City | Phone |
|---|---|---|---|---|
| James Smith | 5th Ave | 10011 | Manhattan | 351-344-5671 |
| Jane R. Smith | 6th Ave | | Manhattan | 351.244.4674 |
| Jane Smith | 6th Ave | 10011 | Manhattan | 351-244-4674 |
| L. Smith, James | 5th Ave | 10011 | Manhattan | 351-344-5671 |
| Peter Cox | Broadway | 10003 | Manhattan | 255-781-3280 |
| Rebecca Ryan | 2nd Avenue | 10043 | MN | 455-231-0872 |
| Smith, J. | Fifth Avenue | 10011 | NYC | 351.344.5671 |

See [Hernandez, Stolfo 1998]

# Complexity (cont.)

***Sort Key is crucial for high recall***

Compose key (e.g., First 3 letters of Name + Seq. of digits in Phone)

Multiple passes with different keys

| Name | Street | Zip | City | Phone |
|---|---|---|---|---|
| James Smith | 5th Ave | 10011 | Manhattan | 351-344-5671 |
| Jane Smith | 6th Ave | 10011 | Manhattan | 351-244-4674 |
| Smith, J. | Fifth Avenue | 10011 | NYC | +1 351.344.5671 |
| L. Smith, James | 5th Ave | 10011 | Manhattan | 351-344-5671 |
| Rebecca Ryan | 2nd Avenue | 10043 | MN | 455-2312-0872 |
| Peter Cox | Broadway | 10003 | Manhattan | 255-781-3280 |
| Jane R. Smith | 6th Ave | | Manhattan | 351.244.4674 |

# Conflict Resolution Functions

| Function | Description | Examples |
|---|---|---|
| Min, Max, Sum, Count, Avg | Standard aggregation | NumChildren, Salary, Height |
| Random | Random choice | Shoe size |
| Longest, Shortest | Longest/shortest value | First_name |
| Choose(source) | Value from a particular source | DoB (DMV), CEO (SEC) |
| ChooseDepending(val, col) | Value depends on value chosen in other column | city & zip, e-mail & employer |
| Vote | Majority decision | Rating |
| Coalesce | First non-null value | First_name |
| Group, Concat | Group or concatenate all values | Book_reviews |
| MostRecent | Most recent (up-to-date) value | Address |
| MostAbstract, MostSpecific, CommonAncestor | Use a taxonomy / ontology | Location |
| Escalate | Export conflicting values | gender |

Copied from: Data Fusion - VLDB 2009 Tutorial - Luna Dong & Felix Naumann

# References


Z. Abedjan, X. Chu, D. Deng, R.C. Fernandez, I.F. Ilyas, M. Ouzzani, P. Papotti, M. Stonebraker, N. Tang. **Detecting data errors: where are we and what needs to be done?.** VLDB, 2016.

Z. Abedjan, L. Golab, F. Naumann. **Profiling relational data: a survey**. VLDB Journal, 2015.

P. Bohannon, W. Fan, M. Flaster, R. Rastogi. **A cost-based model and effective heuristic for repairing constraints by value modification**. SIGMOD, 2005.

J. Chomicki, J. Marcinkowski. **Minimal-change integrity maintenance using tuple deletions**. Information and Computation, 2005.

M.A. Hernández, S.J. Stolfo. **Real-world Data is Dirty: Data Cleansing and The Merge/Purge Problem**. Data Mining and Knowledge Discovery, 1998.

I.F. Ilyas, X. Chu. **Trends in Cleaning Relational Data: Consistency and Deduplication**. Foundations and Trends in databases, 2015.

E.M. Knorr, RR.T. Ng. **Algorithms for Mining Distance-Based Outliers in Large Datasets**. VLDB, 1998.

R. Marsh. **Drowning in dirty data? It's time to sink or swim: A four-stage methodology for total data quality management**. Database Marketing & Customer Strategy Management, 2005.

F. Naumann. **Quality-Driven Query Answering for Integrated Information Systems**. Springer 2002.

T. Papenbrock, J. Ehrlich, J. Marten, T. Neubert, J.-P. Rudolph, M. Schönberg, J. Zwiener, F. Naumann. **Functional dependency discovery: an experimental evaluation of seven algorithms**. VLDB, 2016.

E. Rahm, H.H. Do. **Data cleaning: Problems and current approaches**. IEEE Data Eng. Bull., 2000.

T.C. Redman. **Data Quality for the Information Age**. Artech House, 1996.

R.Y. Wang, D.M. Strong. **Beyond accuracy: What data quality means to data consumers**. J. Management of Information Systems, 1996.

NYU

# The End

*The Impact of Poor Data Quality*

# The Impact of Poor Data Quality (cont.)

***Because of poor data quality …***

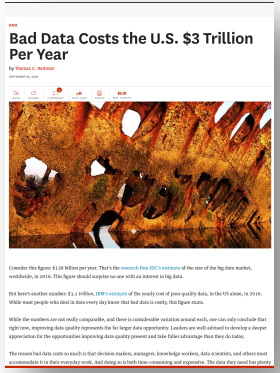
**88%** of data integration projects fail or significantly over-run budgets

**75%** of organizations have additional costs

**33%** of organizations delayed or cancelled new IT systems

**$611bn** per year is lost in the US

In [Marsh 2005] summarizing reports by **Gartner Group**, **PriceWaterhouseCoopers**, and **The Data Warehousing Institute**.



Bad Data Costs the U.S. $3 Trillion Per Year

by Thomas C. Redman

Consider this figure: $110 billion per year. That's the research firm IDC's estimate of the size of the big data market, worldwide, in 2016. This figure should surprise no one with an interest in big data.

But here's another number: $3.1 trillion, IBM's estimate of the yearly cost of poor quality data, in the US alone, in 2016. While most people who deal in data every day know that bad data is costly, this figure stuns.

While the numbers are not really comparable, and there is considerable variation around each, one can only conclude that right now, improving data quality represents the far larger data opportunity. Leaders are well-advised to develop a deeper appreciation for the opportunities improving data quality present and take fuller advantage than they do today.

The reason bad data costs so much is that decision makers, managers, knowledge workers, data scientists, and others must accommodate it in their everyday work. And doing so is both time-consuming and expensive. The data they need has plenty

*How to Measure the Quality of a Dataset?*

# How to Measure Quality of Data

## *Data Quality is Measured as Vector of Quality Criteria*

### Completeness

Quotient of number of missing values and records over all represented entities

### Uniqueness

Number of records that represent the same entity

### Timeliness

Number of records and values that are out of date

### Schema Conformance (Syntactic Integrity)

Number of values that violate format constraints

### Integrity (Semantic Integrity)

Number of records that violate integrity constraints

### Accuracy

Quotient of number correct values and the overall number of values.

See for example [Naumann. 2002][Redman. 1996][Wang, Strong. 1996]

# How to Measure Quality of Data (cont.)

## *Friday Afternoon Measurement*

**Step 1**. Assemble at least 100 data records your group used or created

**Step 2**. Ask two or three people with domain knowledge to join

**Step 3**. Mark obvious errors in a noticeable color.

**Step 4**. Summarize the results by counting the records that are perfect or not.

## *Rule of Ten*

Based on observation that '***it costs 10 times as much to complete a unit of work when the input data are defective as it does when they are perfect.***'

https://hbr.org/2016/07/assess-whether-you-have-a-data-quality-problem

# Some Friday Afternoon Fun

## DOHMH New York City Restaurant Inspection Results

| dba | street | zip | phone | PERFECT |
|---|---|---|---|---|
| O'HARA'S | CEDAR ST | 10006 | 2122673032 | YES |
| TAVOLA | 9 AVE | 10018 | 2122731181 | YES |
| CHEBURECHNAYA | 63 DRIVE | 11374 | 7188979080 | YES |
| SAM'S PIZZA | _WEST 231 ST | ONE | 7185489070 | NO |
| CAFE CON AMOR | ROOSEVELT AVE | 11377 | 7182055707 | YES |
| SHILLA UNION | UNION ST | 11354 | 0000000000 | NO |
| CAFE PRAGUE | WEST 19 ST | 10000 | 2129292602 | YES |
| TINY CUP CAFE | 4 AVE | 11232 | 6463829920 | YES |
| BUTTER & SCOTCH | JAMAICA AVE | 11225 | _____ | NO |
| DREAM PIZZA | JUNCT. BLVD | 11368 | 7189431599 | YES |

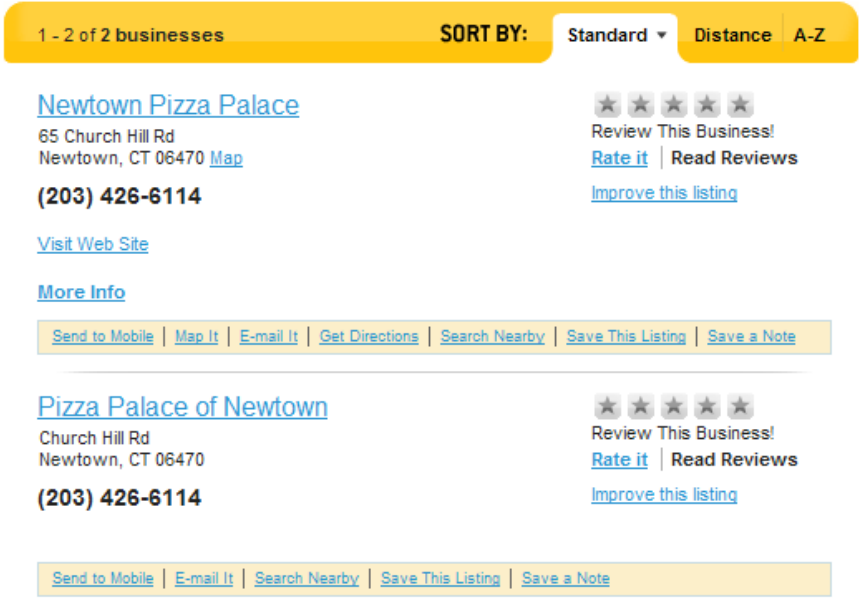
## *Total Cost*

Assume $1 cost to process one record

**7 * $1 + (3 * $1 * 10) = $37** (vs. $10 when all are perfect)

# *Duplicate Detection*

Quiz



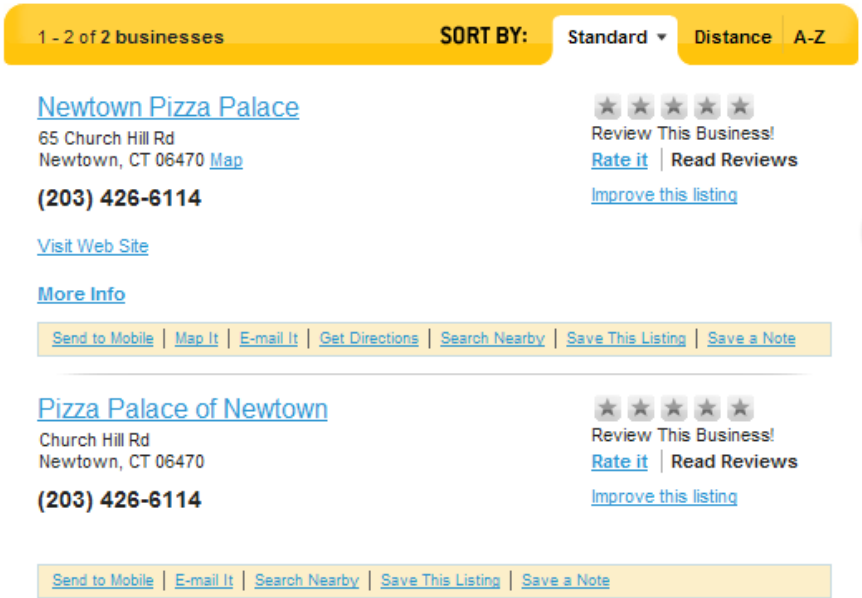

***Which type of listing are they?***

**A**: are the same business

**B**: are different businesses sharing the same phone#

**C**: are different businesses, only one with correct phone#

Copied from: Data Fusion - VLDB 2009 Tutorial - Luna Dong & Felix Naumann



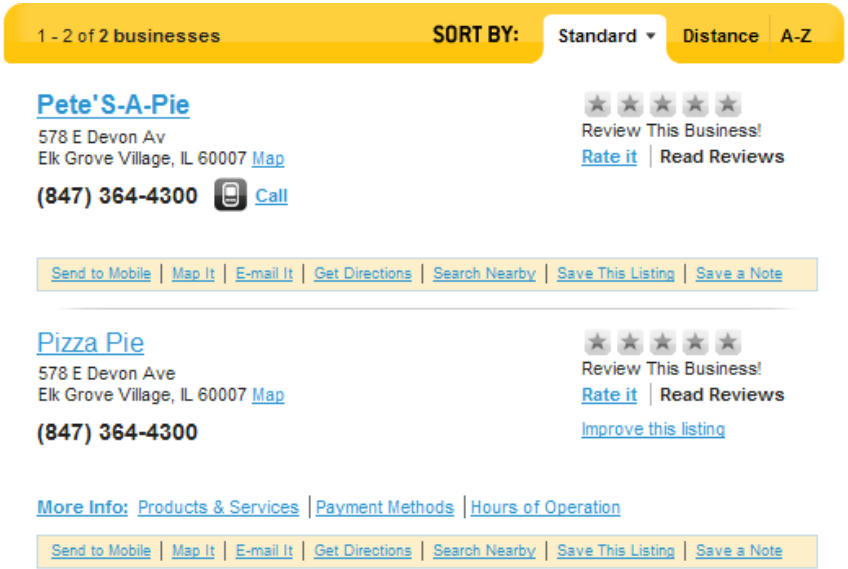

***Which type of listing are they?***

**A**: are the same business

**B**: are different businesses sharing the same phone#

**C**: are different businesses, only one with correct phone#

Copied from: Data Fusion - VLDB 2009 Tutorial - Luna Dong & Felix Naumann



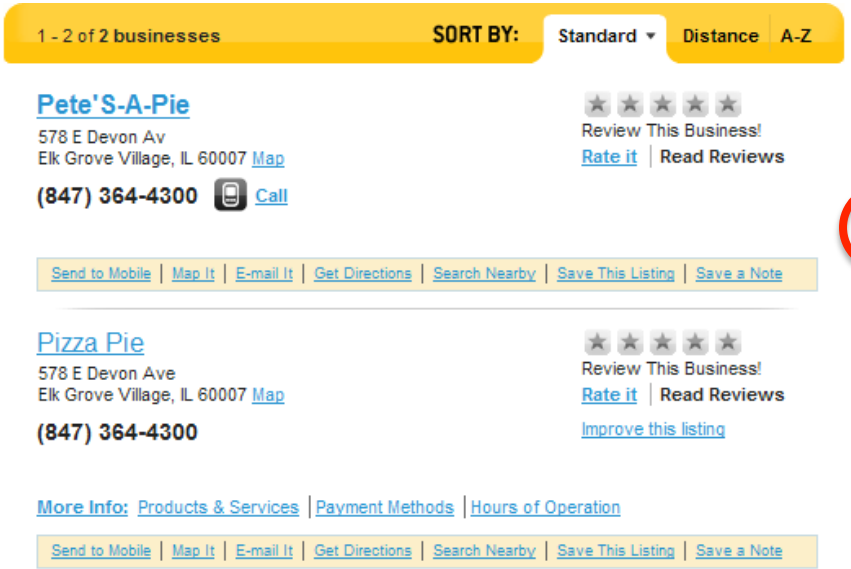

***Which type of listing are they?***

**A**: are the same business

**B**: are different businesses sharing the same phone#

**C**: are different businesses, only one with correct phone#

Copied from: Data Fusion - VLDB 2009 Tutorial - Luna Dong & Felix Naumann

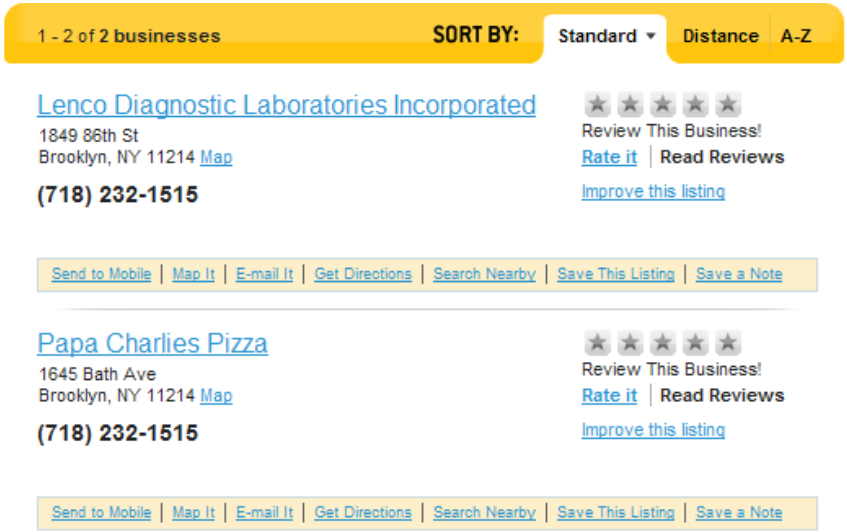



***Which type of listing are they?***

**A**: are the same business

**B**: are different businesses sharing the same phone#

**C**: are different businesses, only one with correct phone#

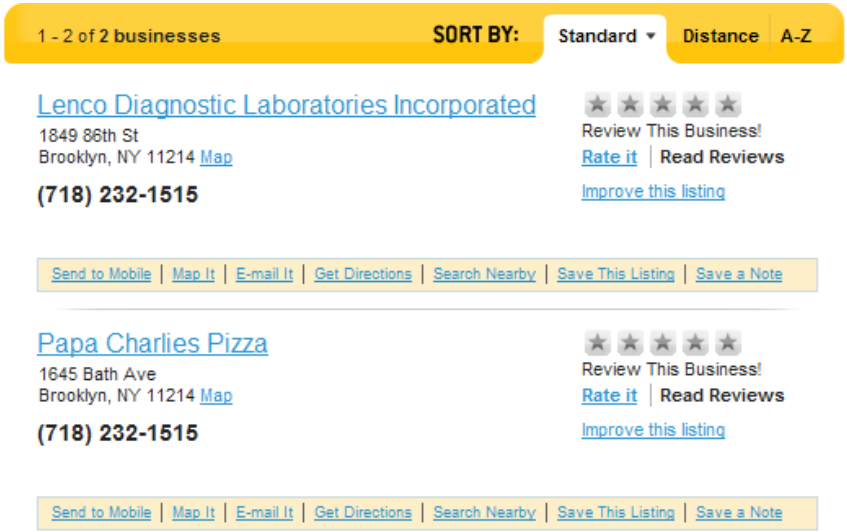



***Which type of listing are they?***

**A**: are the same business

**B**: are different businesses sharing the same phone#

**C**: are different businesses, only one with correct phone#



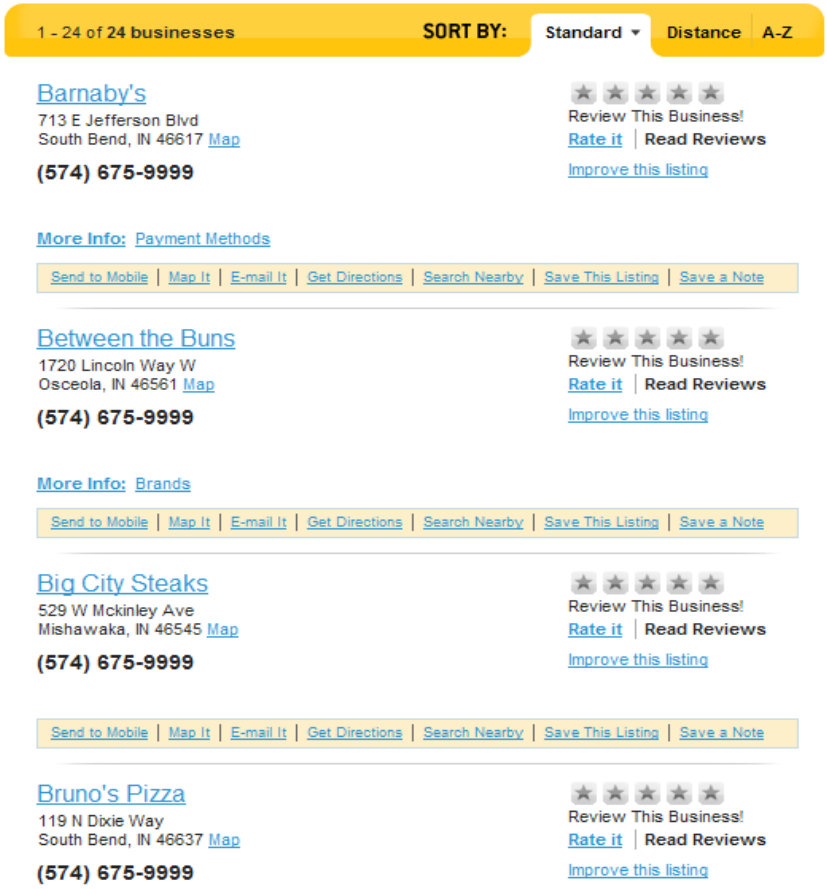

***Which type of listing are they?***

**A**: are the same business

**B**: are different businesses sharing the same phone#

**C**: are different businesses, only one with correct phone#

Copied from: Data Fusion - VLDB 2009 Tutorial - Luna Dong & Felix Naumann



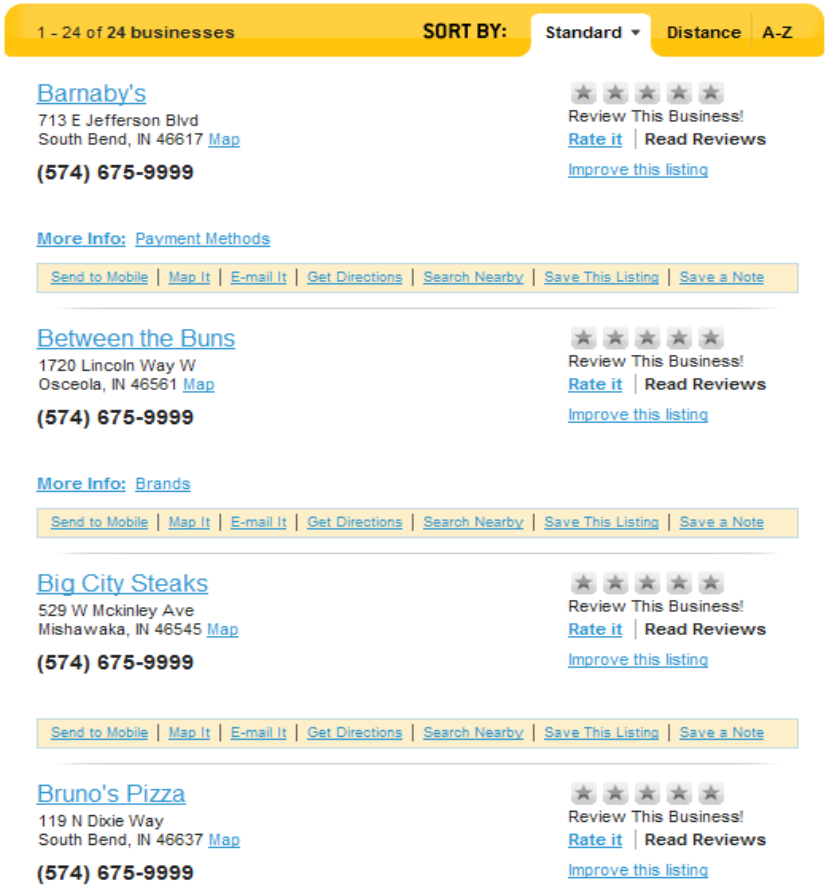

***Which type of listing are they?***

**A**: are the same business

**B**: are different businesses sharing the same phone#

**C**: are different businesses, only one with correct phone#

Copied from: Data Fusion - VLDB 2009 Tutorial - Luna Dong & Felix Naumann




***Which type of listing are they?***

**A**: are the same business

**B**: are different businesses sharing the same phone#

**C**: are different businesses, only one with correct phone#

Copied from: Data Fusion - VLDB 2009 Tutorial - Luna Dong & Felix Naumann

# *Regular Expressions*

Find values that do not match the expected column format

# Regular Expression Discovery

A **regular expression** is a sequence of characters that define a search pattern. … Different syntaxes for writing regular expressions exist.

https://en.wikipedia.org/wiki/Regular_expression

## Example Regular Expression Language

| | |
|---|---|
| **.** | Matches any character |
| ***abc*** | Sequence of characters |
| **[** *abc* **]** | Matches any of the characters inside [ ] |
| ***** | Previous character can be matched zero or more times |
| **?** | Previous character can be matched zero or one time |
| **{***m***}** | Exactly *m* repetitions of previous character |
| **^** | Matches beginning of a line |
| **$** | Matches end of a line |
| **\d** | Matches any decimal digit |
| **\s** | Matches any whitespace character |
| **\w** | Matches any alphanumeric character |
| **\|** | “Or” Operator |

# Regular Expression Discovery (cont.)

| telephone |
| --- |
| (201) 368-1000 |
| (201)373-9599 |
| (718) 206-1088 |
| (718) 206-1121 |
| (718) 206-1420 |
| (718) 206-4420 |
| (718) 206-4481 |
| (718) 262-9072 |
| (718) 868-2300 |
| (718) 206-0545 |
| (814) 681-6200 |
| (888) 8NYC-TRS |
| 800-624-4143 |

## *Challenges*

Do not allow partial matches (^ …$)

Allow for non-matches

General vs. specific patterns

## *Examples*

.*

(201) 368-1000|…|800-624-4143

(?\d{3}[)-] ?\d\w{2}C?-\w{3}\d?

# Regular Expression Discovery (cont.)

| telephone |
| --- |
| 800-624-4143 |
| (201)373-9599 |
| (201) 368-1000 |
| (718) 206-1088 |
| (718) 206-1121 |
| (718) 206-1420 |
| (718) 206-4420 |
| (718) 206-4481 |
| (718) 262-9072 |
| (718) 868-2300 |
| (718) 206-0545 |
| (814) 681-6200 |
| (888) 8NYC-TRS |

| ( | \d | \d | \d | ) | | \d | \w | \w | . | . | \w | \w | \w |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

## *Simple Algorithm (1)*

(1) Group values by length

(2) Find pattern for each group

- Ignore small groups
- Find most specific character at each position

(\d{3}) \d\w{2}.{2}\w{3}

# Regular Expression Discovery (cont.)

| telephone |
| --- |
| 800-624-4143 |
| (201)373-9599 |
| (201) 368-1000 |
| (718) 206-1088 |
| (718) 206-1121 |
| (718) 206-1420 |
| (718) 206-4420 |
| (718) 206-4481 |
| (718) 262-9072 |
| (718) 868-2300 |
| (718) 206-0545 |
| (814) 681-6200 |
| (888) 8NYC-TRS |

| ( | \d | \d | \d | ) | | \d | \d | \d | - | \d | \d | \d | \d |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

## *Simple Algorithm (1)*

(1) Group values by length

(2) Find pattern for each group

- Ignore small groups
- Find most specific character at each position
- Allow for mismatches

(\d{3}) \d{3}-\d{4}

# Regular Expression Discovery (cont.)

| telephone |
| --- |
| (201) 368-1000 |
| (201)373-9599 |
| (718) 206-1088 |
| (718) 206-1121 |
| (718) 206-1420 |
| (718) 206-4420 |
| (718) 206-4481 |
| (718) 262-9072 |
| (718) 868-2300 |
| (718) 206-0545 |
| (814) 681-6200 |
| (888) 8NYC-TRS |
| 800-624-4143 |

## *Simple Algorithm (2)*

(1) Group '*related*' characters into blocks

(2) Find alignment for blocks

[(] [\d{3}] [)] [ ] [\d{3}] [-] [\d{4}]

[(] [\d{3}] [)] [\d{3}] [-] [\d{4}]

---

[(] [\d{3}] [)] [ ]? [\d{3}] [-] [\d{4}]

**(\d{3}) ?\d{3}-\d{4}**

See for example [Fernau. 2009]

*Joinable Columns*

```
SELECT t1.boro, t2.borough, COUNT(*) FROM ds_3h2n_5cm9 t1, ds_pdiy_9ae5 t2
WHERE t1.street = t2.street GROUP BY t1.boro, t2.borough ORDER BY COUNT(*) DESC;
```

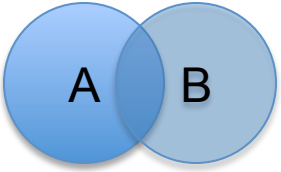
| boro | borough | count |
|---|---|---|
| 1 | MN | 211499381 |
| 3 | MN | 23393719 |
| 4 | MN | 12894922 |
| 1 | QN | 7327541 |
| 1 | BK | 7031201 |
| 2 | MN | 4876745 |
| 3 | BK | 4625184 |
| 4 | QN | 4361375 |
| 2 | BX | 3964086 |
| 1 | BX | 3699564 |
| 3 | QN | 1288089 |
| 5 | MN | 1284778 |
| 4 | BK | 792995 |
| 1 | SI | 671694 |
| 3 | BX | 520046 |
| M | MN | 350751 |
| 4 | BX | 256478 |
| 5 | SI | 250301 |
| 2 | BK | 249341 |
| 2 | QN | 223021 |
| 3 | SI | 101919 |
| 5 | QN | 62336 |
| 4 | SI | 57981 |
| 5 | BK | 55475 |
| Q | MN | 37926 |
| 5 | BX | 32026 |
| 2 | SI | 28607 |
| B | MN | 16856 |

…

# Column Similarity

***How to quantify similarity between columns to find good Join candidates?***

The **Jaccard coefficient** measures similarity between finite sample sets, and is defined as the size of the intersection divided by the size of the union of the sample sets.




https://en.wikipedia.org/wiki/Jaccard_index

**Columns that are (most) similar to DOB Violations column street**

| | | | |
|---|---|---|---|
| 1. | **Historical DOB Permit Issuance** | street | 0.4718 |
| 2. | **311 Service Requests for 2006** | intersection street 1 | 0.4243 |
| **3.** | **311 Service Requests for 2008** | intersection street 1 | 0.4228 |
| | … | ... | ... |
| 95. | **Bureau of Fire Prevention** | street | 0.1281 |

# Column Similarity (cont.)

## *Compare n-Gram Distributions*

3-grams for BROOKLYN:

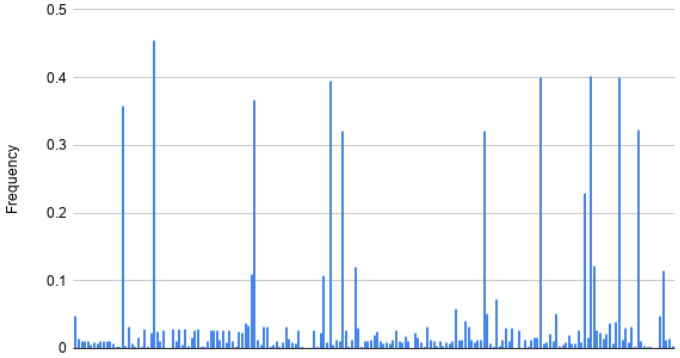
^^BROOKLYN$$ 

^^B
^BR
BRO
ROO
OOK
OKL
KLY
LYN
YN$
N$$

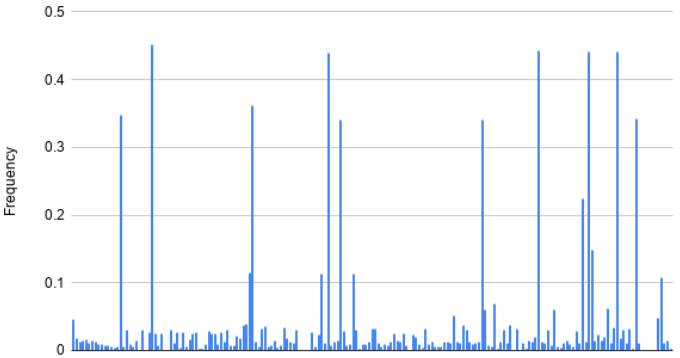
| borough |
| --- |
| {^^B, ^BR, BRO, ROO, OOK, …} |
| {^^B, ^BR, BRO, RON, ONX, …} |
| {^^M, ^MA, MAN, ANH, NHA, …} |
| {^^Q, ^QU, QUE, UEE, EEN, …} |
| {^^^S, ^ST, STA, TAT, ATE, …} |

| | |
| --- | --- |
| ^^B | 2 |
| ^BR | 2 |
| ^^M | 1 |
| ^MA | 1 |
| ^^Q | 1 |
| … | |

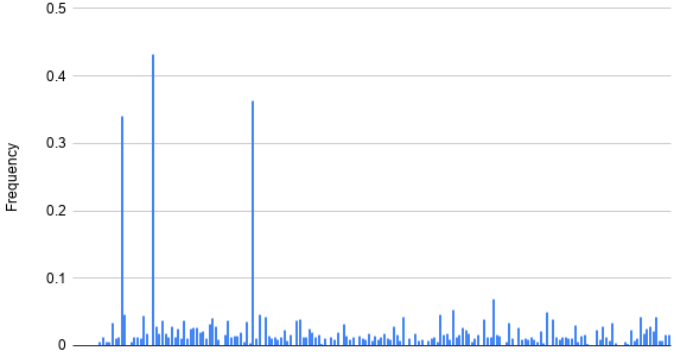
DOB Violations - street



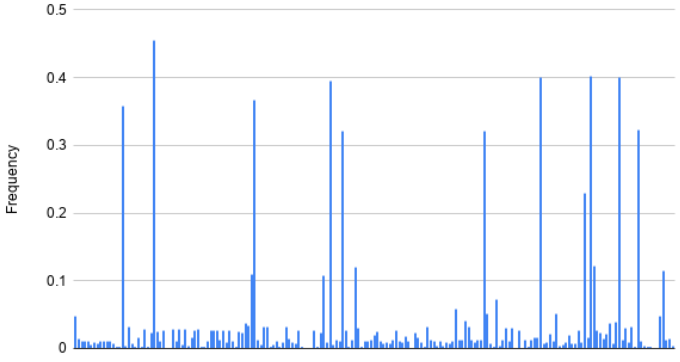

Historical DOB Permit Issuance - street



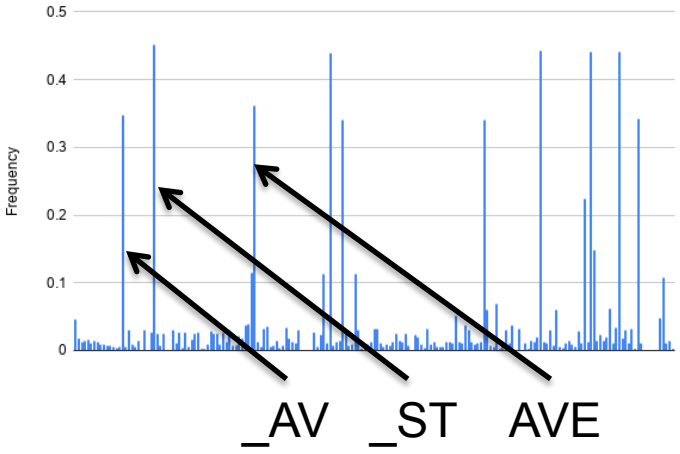

Bureau of Fire Prevention - Certificates of Fitness - street



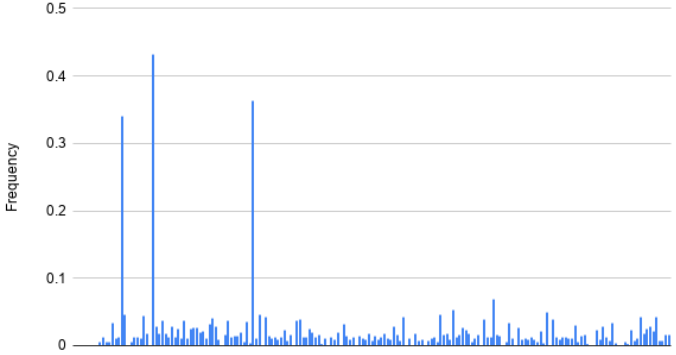
DOB Violations - street



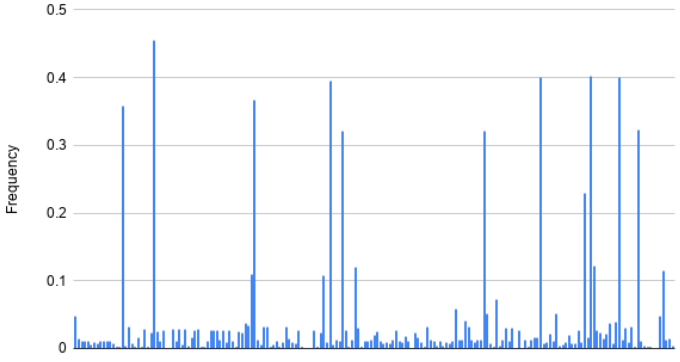

Historical DOB Permit Issuance - street



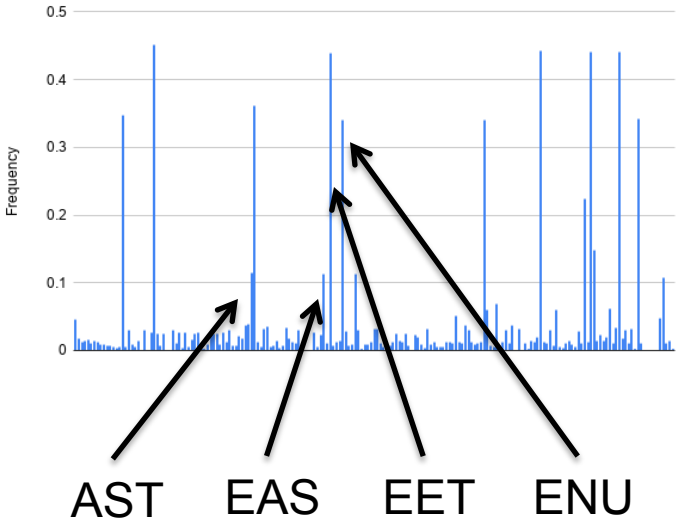

Bureau of Fire Prevention - Certificates of Fitness - street



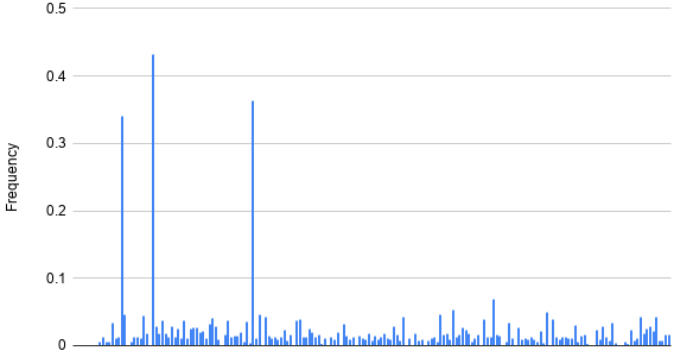
DOB Violations - street




Historical DOB Permit Issuance - street




Bureau of Fire Prevention - Certificates of Fitness - street



```
SELECT t1.boro, t2.borough, COUNT(*) FROM ds_3h2n_5cm9 t1, ds_pdiy_9ae5 t2
WHERE t1.street = t2.street GROUP BY t1.boro, t2.borough
ORDER BY COUNT(*) DESC LIMIT 20;
```

```
 boro|       borough|   count
 ----+--------------+----------
    1|     MANHATTAN| 5267308532
    3|      BROOKLYN|  744539419
    3|     MANHATTAN|  670391402
    4|        QUEENS|  528334681
    1|      BROOKLYN|  434177315
    1|        QUEENS|  211499447
    4|     MANHATTAN|  186754093
    2|         BRONX|  160942856
    2|     MANHATTAN|  143183571
    3|        QUEENS|   96978865
    1|         BRONX|   82875245
    4|      BROOKLYN|   76663413
    5| STATEN ISLAND|   42181567
    1| STATEN ISLAND|   27183497
    5|     MANHATTAN|   26715611
    3|         BRONX|   25753513
    2|      BROOKLYN|   22577970
    M|     MANHATTAN|   11476102
    2|        QUEENS|   10617312
    3| STATEN ISLAND|   10596340
(20 rows)
```

```
SELECT q1.borough, q2.borough, COUNT(*) FROM (
  SELECT street, MAX(boro) AS borough FROM ds_3h2n_5cm9
  GROUP BY street HAVING COUNT(DISTINCT boro) = 1
) AS q1, (
  SELECT street, MAX(borough) as borough FROM ds_bty7_2jhb
  GROUP BY street HAVING COUNT(DISTINCT borough) = 1
) AS q2 WHERE q1.street = q2.street
GROUP BY q1.borough, q2.borough ORDER BY COUNT(*) DESC LIMIT 20;
```

```
borough|       borough|count(1)|
-------+--------------+--------+
      4|        QUEENS|    2426|
      5| STATEN ISLAND|    1739|
      3|      BROOKLYN|    1505|
      1|     MANHATTAN|    1450|
      2|         BRONX|    1380|
      1|      BROOKLYN|      14|
      4|      BROOKLYN|      12|
      3|     MANHATTAN|      10|
      3|        QUEENS|       6|
      2|        QUEENS|       5|
      1|         BRONX|       4|
      2|      BROOKLYN|       4|
      3| STATEN ISLAND|       4|
      2|     MANHATTAN|       4|
      2| STATEN ISLAND|       3|
      B|      BROOKLYN|       2|
      3|         BRONX|       2|
      4| STATEN ISLAND|       2|
      4|         BRONX|       1|
      5|      BROOKLYN|       1|
...
```