
Data Quality Automation
21 February 2018
DAMA Chicago

Data Quality Automation

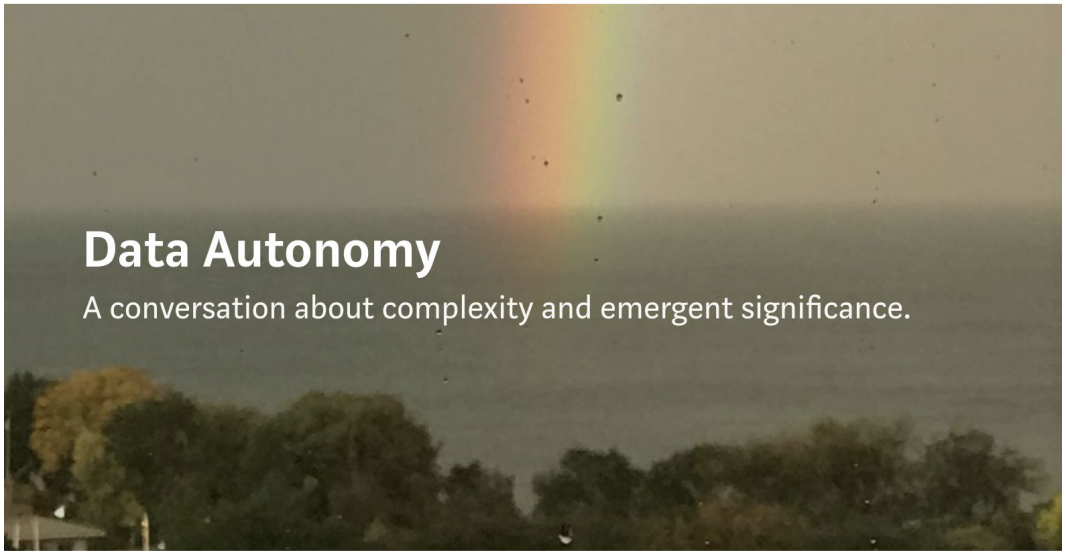
DAMA Chicago
21 February 2018

Kevin Kautz

Data Engineering Manager, CCC Information Services
previously Principal Architect, Nielsen

Data Autonomy

https://medium.com/data-autonomy



FEBRUARY 2018

Meaning in Motion
2 min read · In Data Autonomy · View story · Referrers

Data Is Conversation
3 min read · In Data Autonomy · View story · Referrers

Environmental Protection for Data
3 min read · In Data Autonomy · View story · Referrers

JANUARY 2018

Classifying Classifiers
3 min read · In Data Autonomy · View story · Referrers

Data Persistence, not Data Storage
3 min read · In Data Autonomy · View story · Referrers

Autonomous Data Governance
3 min read · In Data Autonomy · View story · Referrers

Data, Know Thyself
2 min read · In Data Autonomy · View story · Referrers

Autonomous Data as a Beginner
2 min read · In Data Autonomy · View story · Referrers

Data Quality Automation

- **Data Quality**
- Data in Motion
- Metadata that Matters
- Defining a Data Test
- Data Quality Agreements

Data Quality

You can measure its quality only to the extent that you know how you will use it.

Data describes something. What you choose to notice and to record about that something depends entirely on the questions you want to ask and answer.

Data quality is the fitness of the data to answer the questions that you ask.

Data Quality

From the American Society for Quality, we understand the **Cost of Quality** as follows:

- **Prevention** catch, quarantine
- **Appraisal** identify, measure, trend
- **Failure**
  - Internal waste, rework
  - External repair, restatement, lack of trust

http://asq.org/learn-about-quality/cost-of-quality/overview/overview.html

Data Quality

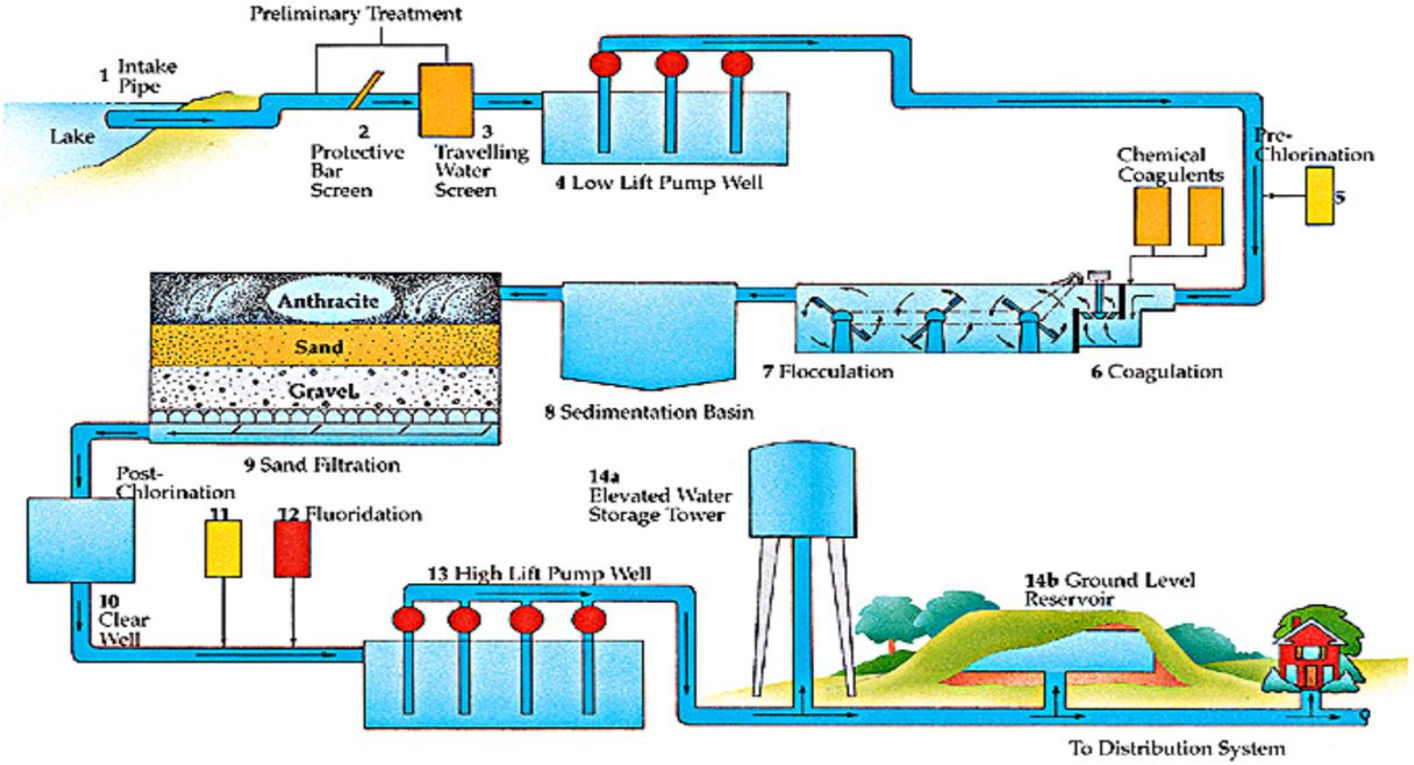
How do you measure the value of data?

We could consider data's trustworthiness, recency, accuracy, or completeness. Or we can measure the data production process for MTBF (mean time between failures) and MTTR (mean time to resolve).

That's good. But no, that's not it.

The value of data is whether it is used. More specifically, whether it is used in a revenue-generating product or service. Data that is not used has no value.

Data Quality



http://cof-cof.ca/surface-water-treatment-plant-flow-diagram/

Data Quality

We will ...

- Decide where and when to measure data quality
- Determine what matters and what does not
- Define a data quality test framework
- Deliver sufficient data quality
- Demonstrate

Data Quality Automation

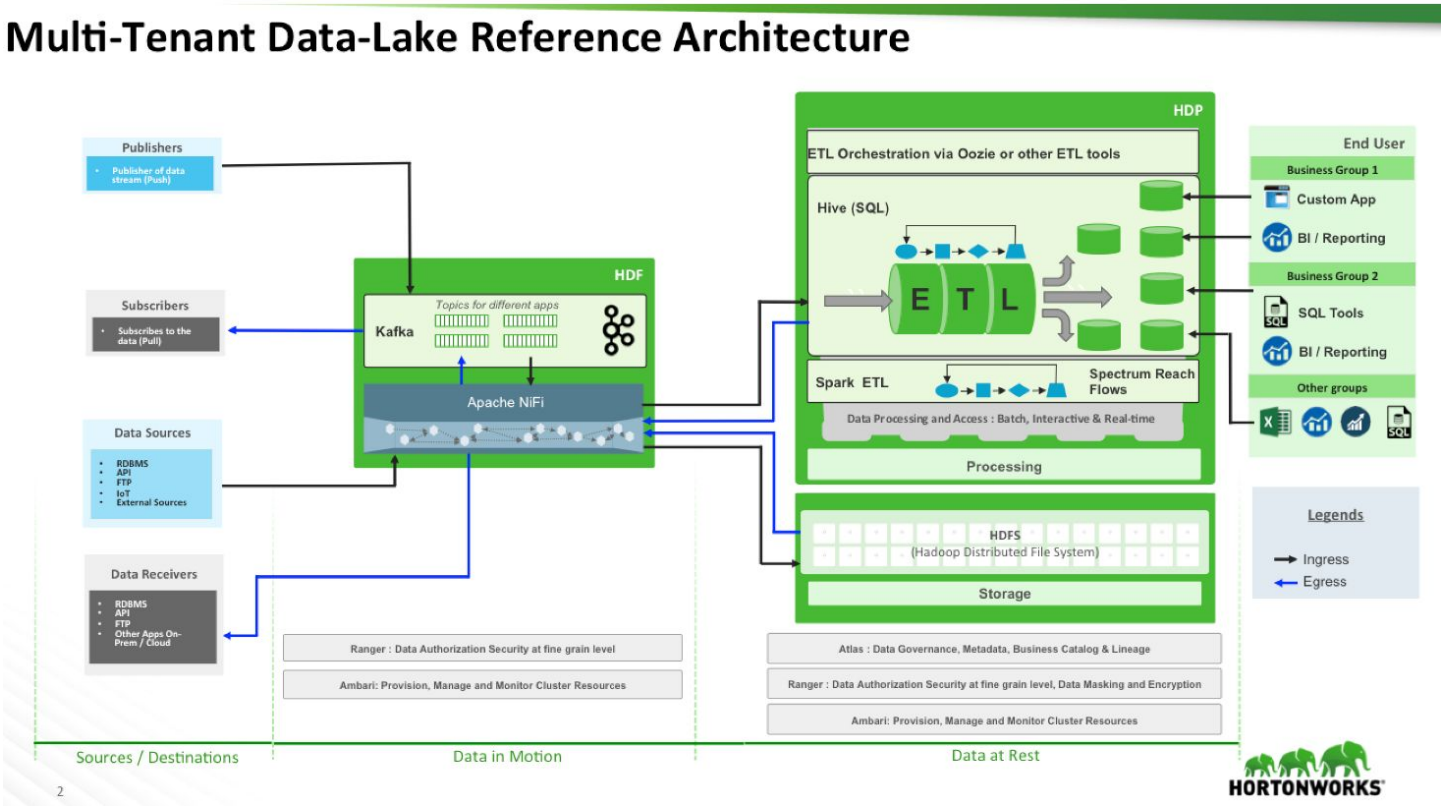
- Data Quality
- **Data in Motion**
- Metadata that Matters
- Defining a Data Test
- Data Quality Agreements

Data in Motion

Data in use is valuable data.

Data at rest is not.

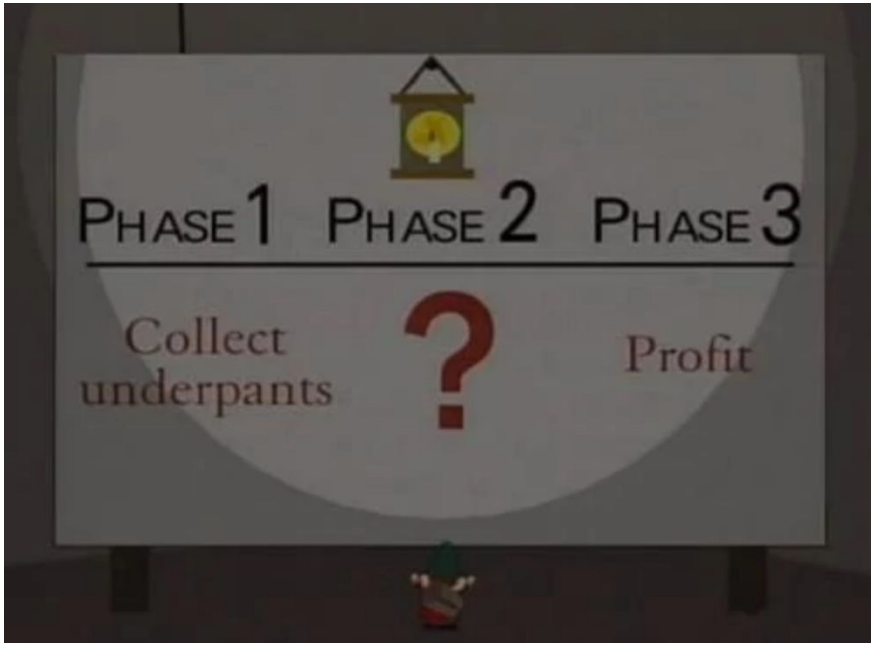
Data in Motion



https://hortonworks.com

K.Kautz 21-Feb-2018

Data in Motion



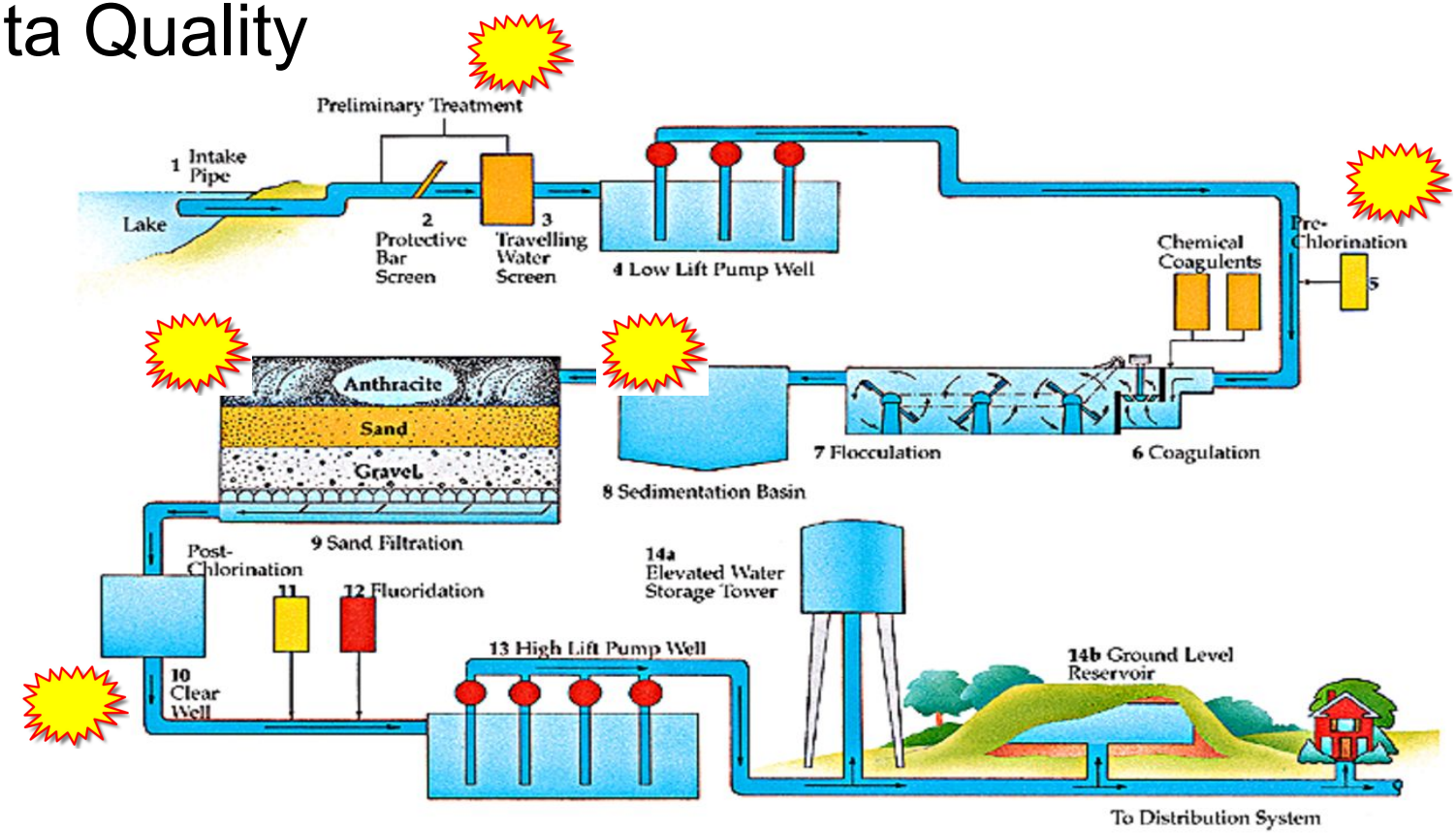
http://southpark.wikia.com/wiki/Underpants_Gnomes

Data in Motion

Trends in data:

FROM…	TO…
Relational only	Relational + JSON + unstructured
Data model is central	Application behavior is central
Structure holds meaning	Meaning by reference
Storage dictates access paths	Access paths dictate storage

Data Quality



http://cof-cof.ca/surface-water-treatment-plant-flow-diagram/

Data in Motion

As data moves through processing engines,

- the data source is no longer known,
- computations and aggregations occur,
- the data is used for different purposes.

Data quality is fitness for use. You have to test it early so that you can identify how to correct it. You have it test it separately for each intended use.

Data in Motion

Consider how data will be used and why it changes in the following situations:

1. Descriptive analytics --> Separate each feature and fact
2. Business analytics --> KPIs that agree to ignore outliers
3. Predictive analytics --> Modeling that avoids over-fitting
4. Prescriptive analytics --> Recommend actions to take

or

5. Proscriptive analytics? --> Forbid actions to take

Data Quality Automation

- Data Quality
- Data in Motion
- **Metadata that Matters**
- Defining a Data Test
- Data Quality Agreements

Metadata that Matters

In philosophy, the word ontology is used to discuss the study of being, of existence. In science, we add the indefinite pronoun, and we speak of “an ontology”. An ontology is a subject area domain. It is the real world context for real world behaviors that your data describes.

Ontology =

- Entities
- Relationships
- Facts
- Features

Yes, you can represent an ontology as a data model. More specifically, an ontology is a combination of the conceptual and the logical models.

It includes representational classes which constrain the data types of the logical model.

Metadata that Matters

Taxonomy is, more or less, classification. But it also includes everything necessary to create sub-groups and distinguish them from each other, because that is needed to classify effectively.

Taxonomy =

- Provenance
- Identity
- Hierarchy
- Significant features
- Range of variation
- Range of values

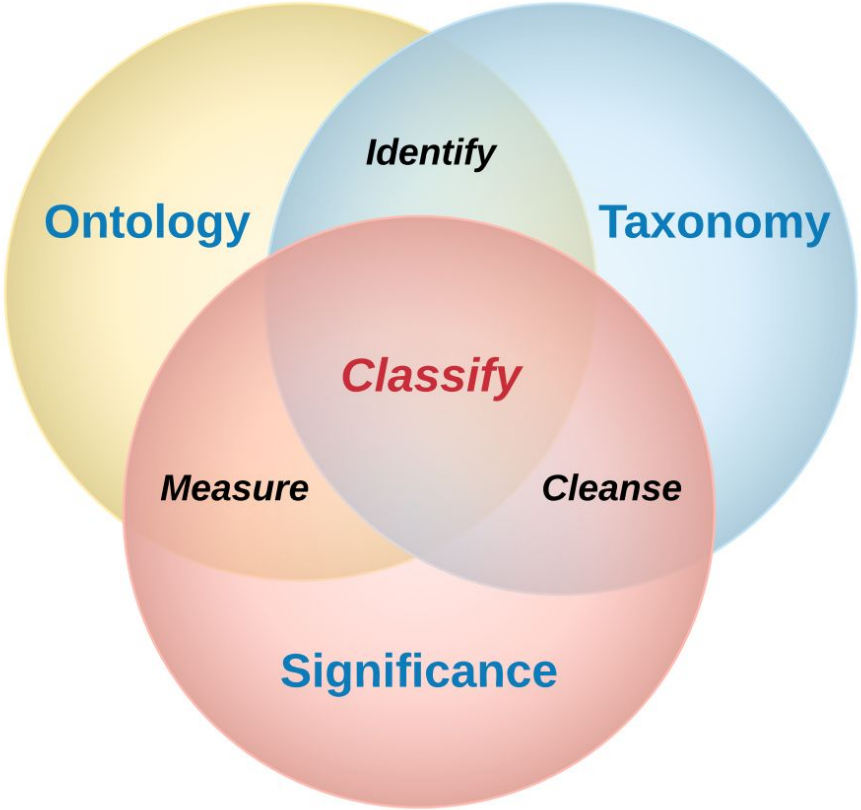
Metadata that Matters

Significance is the most challenging to discuss. It gets back to the original premise of what data is. Do you remember?

> Data describes something. What you choose to notice and to record about that something depends entirely on the questions you want to ask and answer.

Signficance is our decision of what to include or exclude, to filter, to trim, to top-code or bottom-code, to extrapolate or interpolate or simply provide default values for. Significance is where we specify what matters and what does not.

Metadata that Matters



Metadata that Matters

Historically, much of the ontology and taxonomy and significance have been encoded in the third-normal form, or in the data-types, or occasionally, in our carefully chosen table & column naming conventions.

When data in motion moves out of its structure, and the structure held the meaning, how do we preserve the meaning without the structure?

There are several choices. One is metadata management, where dictionaries are more than merely catalogs. Another is to use JSON schemas. And a third technique is to include metadata tags as features within the data itself.

The best idea is to use all three approaches.

Data Quality Automation

- Data Quality
- Data in Motion
- Metadata that Matters
- **Defining a Data Test**
- Data Quality Agreements

Defining a Data Test -- definition

```
- id: X001
  definition:
    column: cost_of_repair
    table: fact_claim_folder
    database: dev_property_pub
    method: amount
```

Defining a Data Test -- definition

```
- id: X002
  definition:
    column: c_estimate_amt
    table: fact_claim_file_detail
    database: dev_property_pub
    method: amount
```

Defining a Data Test -- definition

```
- id: X003
  definition:
    column: clm_typ_cd
    table: claim
    database: common_property
    method: code
```

Defining a Data Test -- execution

```
- id: X001
  execution:
    load:
      type: limit # full/limit/sample(key and size)/latest(watermark)
      size: 100000
      condition: file_src_cd='PW'
      sort_key: clm_nbr
      sort_dir: desc
```

Defining a Data Test -- execution

```
- id: X002
  execution:
    load:
      type: full # full/limit/sample(key and size)/latest(watermark)
      size:
      condition:
      sort_key:
      sort_dir:
```

Defining a Data Test -- execution

```
- id: X003
  execution:
    load:
      type: full # full/limit/sample(key and size)/latest(watermark)
      size:
      condition:
      sort_key:
      sort_dir:
```

Defining a Data Test -- results: filtered “amount”

```
********************************
Log - Initiate
run id     : 01e2170e-21a5-4734-94d5-d31e2c855b90
task id    : X001
start time : 2018-02-04 08:56:36.088634
column     : dev_property_pub.fact_claim_folder.cost_of_repair
sql        : select cost_of_repair from dev_property_pub.fact_claim_folder  where file_src_cd='PW' order
by clm_nbr desc limit 100000
============================
Total count : 100000
Valid count : 100000.0
Null count  : 0.0
Null %      : 0.0%
Min         : 0.0
Max         : 100486.8
Mean        : 3982.45
5%          : 524.06
25%         : 1314.4
50%         : 2640.92
75%         : 5149.87
95%         : 11951.1
============================
Log - Completed
run id   : 01e2170e-21a5-4734-94d5-d31e2c855b90
end time : 2018-02-04 08:56:47.159391
```

Defining a Data Test -- results: unfiltered “amount”

```
********************************
Log - Initiate
run id     : 01e2170e-21a5-4734-94d5-d31e2c855b90
task id    : X002
start time : 2018-02-04 08:56:47.159523
column     : dev_property_pub.fact_claim_file_detail.c_estimate_amt
sql        : select c_estimate_amt from dev_property_pub.fact_claim_file_detail
==============================
Total count : 23348097
Valid count : 13146956.0
Null count  : 10201141.0
Null %      : 43.692%
Min         : -4952.8
Max         : 3078000.0
Mean        : 1849.46
5%          : 0.0
25%         : 0.0
50%         : 200.78
75%         : 2230.47
95%         : 8340.27
==============================
Log - Completed
run id   : 01e2170e-21a5-4734-94d5-d31e2c855b90
end time : 2018-02-04 08:58:34.591261
```

Defining a Data Test -- results: “code”

```
********************************
Log - Initiate
run id     : 01e2170e-21a5-4734-94d5-d31e2c855b90
task id    : X003
start time : 2018-02-04 08:58:34.591382
column     : common_property.claim.clm_typ_cd
sql        : select clm_typ_cd from common_property.claim
==============================
Total count : 50292513
Valid count : 50292381
Null count  : 132
Null %      : 0.0%
Name: clm_typ_cd, dtype: int64
==============================
Log - Completed
run id   : 01e2170e-21a5-4734-94d5-d31e2c855b90
end time : 2018-02-04 09:02:50.722100
```

```
Frequency table:
VE    50035441
MC      177235
RV       15982
HT       15379
OT       14872
HE       12349
TR        8826
ST        5926
WC        2490
BS        2011
SR        1795
EM          68
CA           7
```

Defining a Data Test

amount: [pctnull, count, distinct, min, max, mean, stddev, topten],

text: [pctnull, count, distinct, min, max, topten],

range: [pctnull, count, distinct, toolow, toohigh], ⇐ defn has range

code: [pctnull, count, distinct, invalid] ⇐ defn has list of values

Data Quality Automation

- Data Quality
- Data in Motion
- Metadata that Matters
- Defining a Data Test
- **Data Quality Agreements**

Data Quality Agreements

Using the above data-test definitions as a building block, we add the following behaviors:

1. Define data quality tests as described above.
2. Define comparisons and triggered alerts that compare test results.
3. Establish points during a data pipeline when data quality tests will run.
4. Tag these pipeline moments (such as “raw” and “refined”...)
5. When tests run, record the results in a queryable table.
6. Run comparisons and alerts at appropriate moments in the pipeline.

Data Quality Agreements

Data Test Reporting

Always append to application log file.

If test execution is set to “table”, also append to table:

testId	tag	timestamp	pctnull	min	max	count	distinct	mean	stddev	topten

Data Quality Agreements

Data Test Execution

```
{
        testId: 000001,
        tag: raw,
        sample: 1.0, ← 100% sample
        limit: 0,      ← no limit
        frequency: daily,
        reporting: table
}
```

```
{
        testId: 000001,
        tag: refined,
        sample: 1.0,
        limit: 0,
        frequency: daily,
        reporting: table
}
```

Data Quality Agreements

Data Test Comparison

```
{
        testId: 000001,
        tags: [ raw, refined ],
        expect: [
                count: equal,
                pctnull: nondecreasing
        ],
        . . .
}
```

Data Quality Agreements

Data Test Alerting

```
{
        testId: 000001,
        tags: [ raw, refined ],
        expect: [ . . . ],
        [ { threshold: 0.0,   ← that's 0.0%, so any discrepancy alerts
          alert: warning } ]  ← warning= email; error= stop data flow
}
```

Data Quality -- Python code

```
import yaml
import argparse

from pyspark import SparkConf, SparkContext
from pyspark.sql import HiveContext
from pyspark.sql import Window
from pyspark.sql.functions import rank,desc
from pyspark.sql import functions as F
import pandas as pd

from utils import Utils
```

Data Quality -- Python code

```
parser = argparse.ArgumentParser(description='')
parser.add_argument('--tasks',required=True ,help='Infomation about data you want to test.')

conf = SparkConf().setAppName('SONAR')
sc = SparkContext(conf=conf)
hiveContext = HiveContext(sc)

sc.setLogLevel("ERROR")

myUtils = Utils(hiveContext)
```

Data Quality -- Python code

```
def read_tasks_list():
        args = parser.parse_args()
        my_tasks_file= args.tasks

        info =yaml.load(open(my_tasks_file,'r'))
        tasks = info['tasks']
        return tasks
```

Data Quality -- Python code

```
def create_task_dict(t):
        task_dict= {}
        task_dict['id'] =  t['id']
        task_dict['col'] = t['definition']['column']
        task_dict['tbl'] = t['definition']['table']
        task_dict['db'] = t['definition']['database']
        task_dict['method'] = t['definition']['method']
        task_dict['type'] = t['execution']['load']['type']
        task_dict['size'] = t['execution']['load']['size']
        task_dict['condition'] = t['execution']['load']['condition']
        task_dict['sort_key'] = t['execution']['load']['sort_key']
        task_dict['sort_dir'] = t['execution']['load']['sort_dir'
        task_dict['description'] = str(t)
        return task_dict
```

Data Quality -- Python code

```
def build_sql_strings(task_dict):
    sql_part_one = "select {col} from {db}.{tbl} ".format(col=task_dict['col'],db=task_dict['db'],tbl=task_dict['tbl'])
    sql_part_two =""

    if task_dict['type'] == "full":
        pass
    elif task_dict['type'] == "limit":
        if task_dict['condition'] != None:
            sql_part_two = " where {condition}".format(condition=task_dict['condition'])
        if task_dict['sort_key'] != None:
            sql_part_two = sql_part_two + " order by {key} {sdir}".format(key=
                                                    task_dict['sort_key'],sdir=task_dict['sort_dir'])
            sql_part_two = sql_part_two + " limit {lmt}".format(lmt=task_dict['size'])
    else:
        pass
    sql_final = sql_part_one + sql_part_two
    return sql_final
```

Data Quality -- Python code

```
def genarate_stats(task):

    spark_df = hiveContext.sql(task['sql'])
    pdf = spark_df.toPandas()
    pdf1= pdf.dropna(axis=0, how='all')

    k = pdf1.describe(percentiles=[.05,.25,.50,.75,.95]).to_dict()

    print('==============================')
    total_count = spark_df.count()
    null_count = total_count - k[task['col']]['count']

    print("Total count : " + str(total_count))
    print("Valid count : " + str(round(k[task['col']]['count'],0)))
    print("Null count  : " + str(round(null_count,0)))
    print("Null %      : " + str( round((null_count/total_count)*100,3) )+ '%')
```

Data Quality -- Python code

```
if task['method']=='amount':
        print("Min         : " + str(round(k[task['col']]['min'],2)))
        print("Max         : " + str(round(k[task['col']]['max'],2)))
        print("Mean        : " + str(round(k[task['col']]['mean'],2)))
        print("5%          : " + str(round(k[task['col']]['5%'],2)))
        print("25%         : " + str(round(k[task['col']]['25%'],2)))
        print("50%         : " + str(round(k[task['col']]['50%'],2)))
        print("75%         : " + str(round(k[task['col']]['75%'],2)))
        print("95%         : " + str(round(k[task['col']]['95%'],2)))
elif task['method']=='code':
        print("Frequency table:")
        print(pd.value_counts(pdf1[task['col']]))
else:
        pass
print('==============================')
```

Data Quality -- Python code

```
def task_runner(tasks):
        for tsk in tasks:
                task_dict = create_task_dict(tsk)
                sql_str= build_sql_strings(task_dict)
                task_dict['sql'] = sql_str

                myUtils.initiate_run_log(task_dict)
                genarate_stats(task_dict)
                myUtils.complete_runlog()
```

Data Quality -- Python code

```
def XXX_build_sql_strings(tasks):
        out_list=[]
        #genarate sql string
        for t in tasks:
                def_id = t['id']
                def_col = t['definition']['column']
                def_tbl = t['definition']['table']
                def_db = t['definition']['database']
                def_method = t['definition']['method']

                exe_type = t['execution']['load']['type']
                exe_size = t['execution']['load']['size']
                exe_sort_key = t['execution']['load']['sort_key']
                exe_sort_dir = t['execution']['load']['sort_dir']
```

Data Quality -- Python code

```
            sql_part_one = "select {col} from {db}.{tbl} ".format(col=def_col,db=def_db,tbl=def_tbl)
            sql_part_two =""
            if exe_type == "full":
                pass
            elif exe_type == "limit":
                if exe_sort_key != None:
                    sql_part_two =" order by {key} {sdir}".format(key=exe_sort_key,sdir=exe_sort_dir)
                sql_part_two = sql_part_two + " limit {lmt}".format(lmt=exe_size)
            else:
                pass
            sql_final = sql_part_one + sql_part_two
            out_list.append(sql_final)
        return out_list
T = read_tasks_list()
task_runner(T)
```

Data Quality Automation

- Data Quality
- Data in Motion
- Metadata that Matters
- Defining a Data Test
- Data Quality Agreements

Data Quality

We will ...

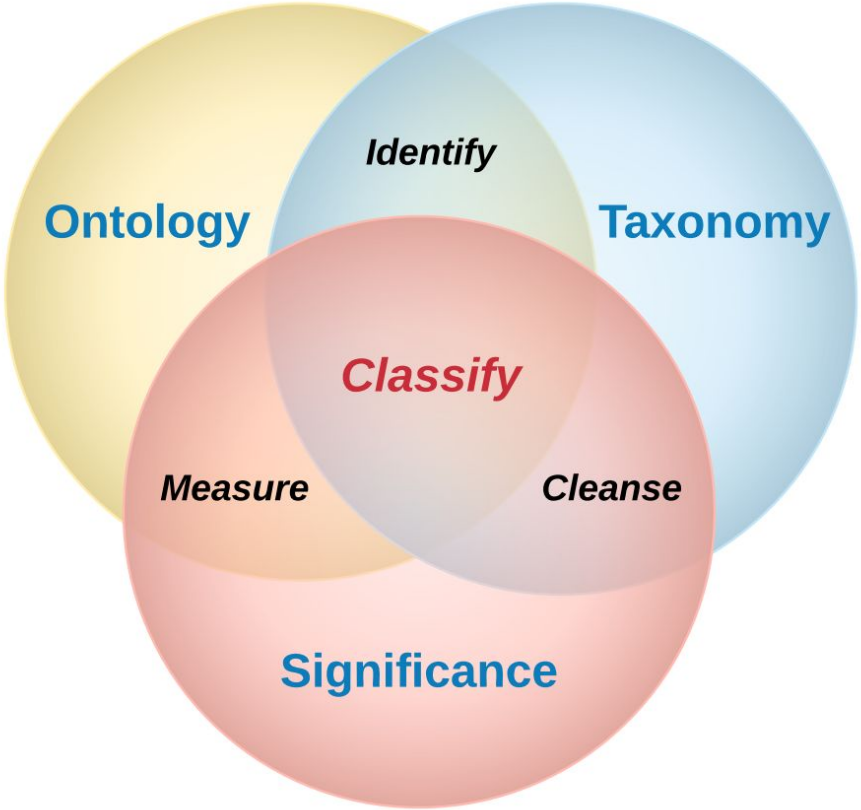
- Decide where and when to measure data quality
- Determine what matters and what does not
- Define a data quality test framework
- Deliver sufficient data quality
- Demonstrate

Data in Motion

Trends in data:

FROM…	TO…
Relational only	Relational + JSON + unstructured
Data model is central	Application behavior is central
Structure holds meaning	Meaning by reference
Storage dictates access paths	Access paths dictate storage

Metadata that Matters



Defining a Data Test -- definition

```
- id: X003
  definition:
    column: clm_typ_cd
    table: claim
    database: common_property
    method: code
```

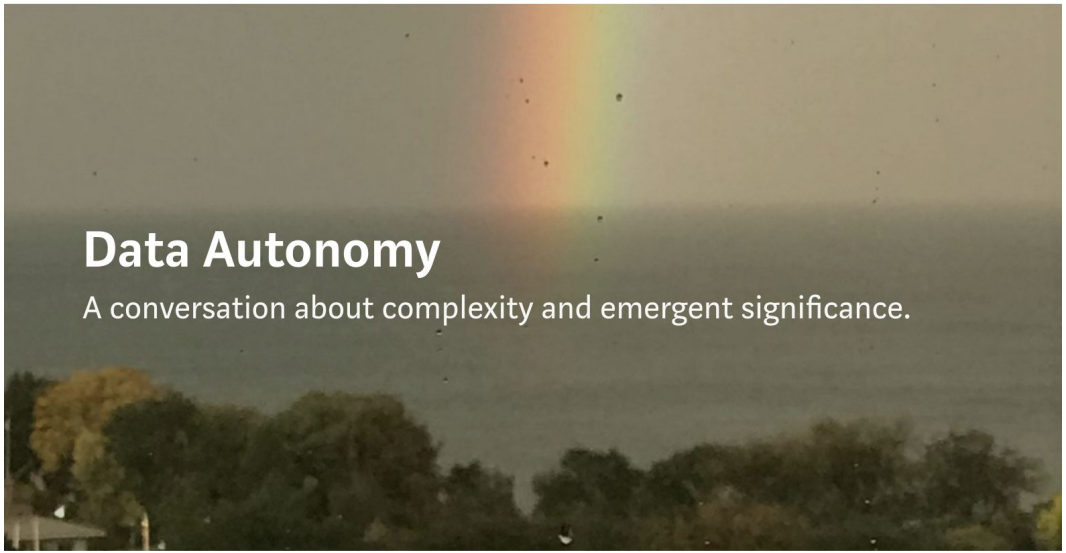
Data Quality Agreements

Data Test Comparison

```
{
        testId: 000001,
        tags: [ raw, refined ],
        expect: [
                count: equal,
                pctnull: nondecreasing
        ],
        . . .
}
```

Data Autonomy

https://medium.com/data-autonomy



FEBRUARY 2018

Meaning in Motion
2 min read · In Data Autonomy · View story · Referrers

Data Is Conversation
3 min read · In Data Autonomy · View story · Referrers

Environmental Protection for Data
3 min read · In Data Autonomy · View story · Referrers

JANUARY 2018

Classifying Classifiers
3 min read · In Data Autonomy · View story · Referrers

Data Persistence, not Data Storage
3 min read · In Data Autonomy · View story · Referrers

Autonomous Data Governance
3 min read · In Data Autonomy · View story · Referrers

Data, Know Thyself
2 min read · In Data Autonomy · View story · Referrers

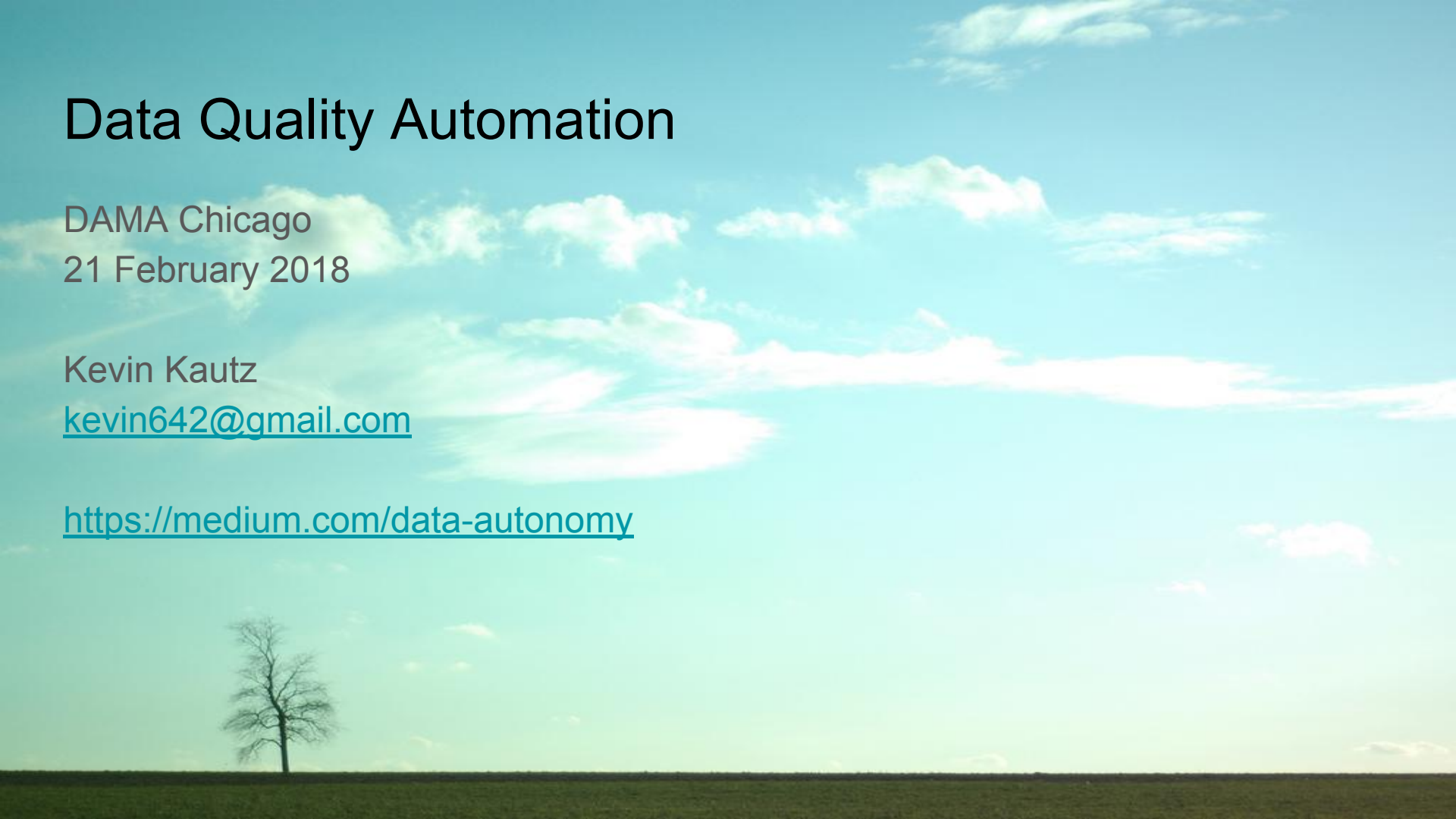
Autonomous Data as a Beginner
2 min read · In Data Autonomy · View story · Referrers

Data Quality Automation

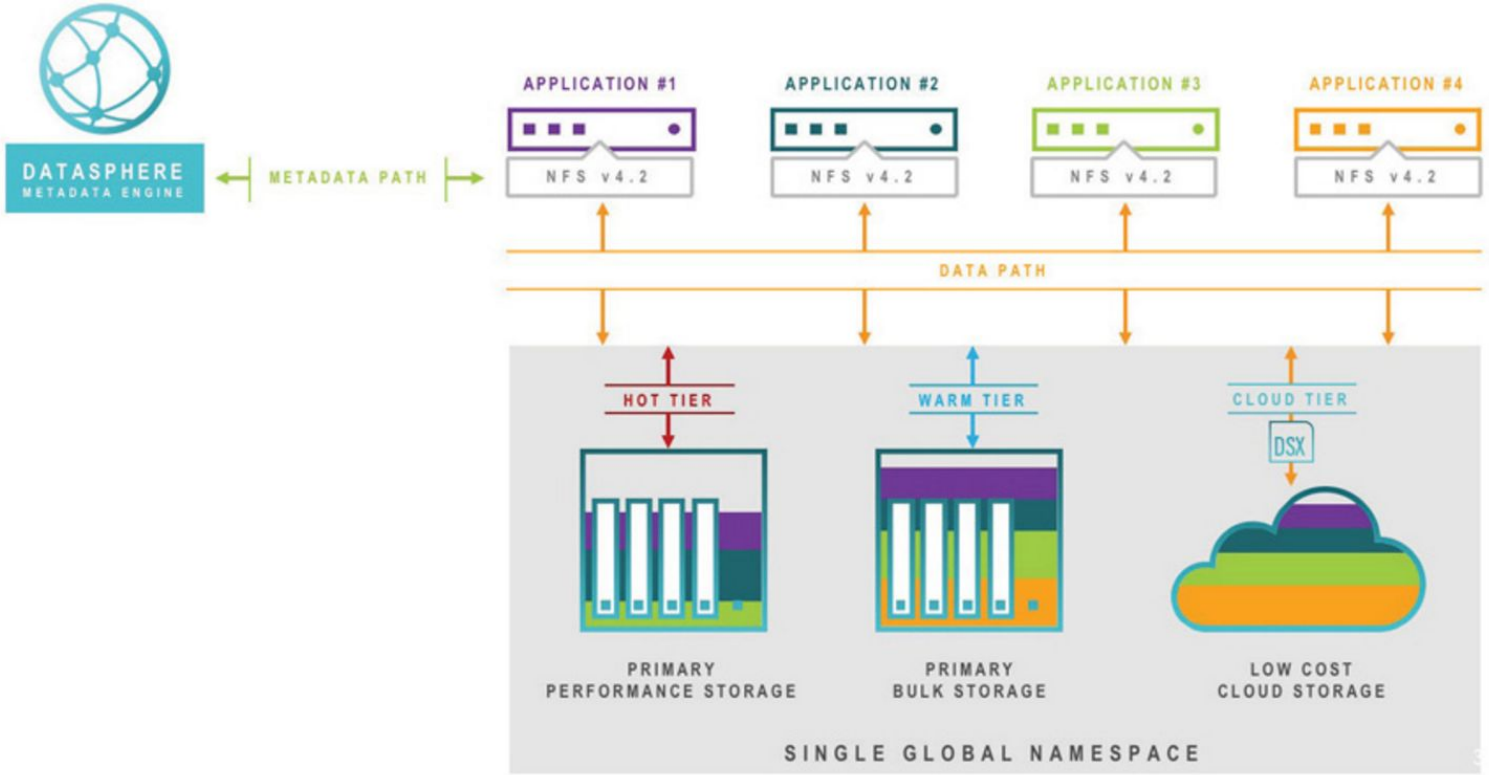
DAMA Chicago
21 February 2018

Kevin Kautz
kevin642@gmail.com

https://medium.com/data-autonomy



Data as a Service (DaaS)



http://primarydata.com/products/datasphere-overview