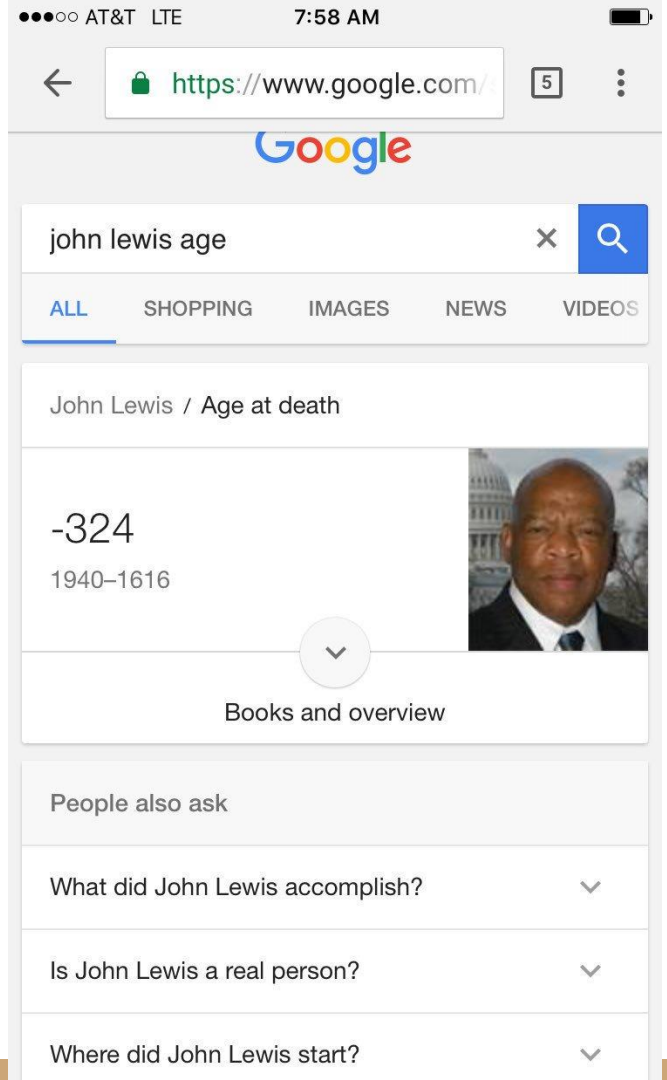
Data Unit Tests with Python

kjamistan

Repo For Installation

- https://github.com/kjam/data-cleaning-101

Data Validation in the Wild



kjamistan

Data Errors are Costly

..one consultant from a large database vendor noted that errors might be found well after some result is reported:

Most of these errors are subtle enough that the analysis will go through e.g., with standard null value semantics of SQL, but give an incorrect answer. Usually is only caught weeks later after someone notices something like… well the Wilmington branch cannot have 1M sales in a week.

Towards Reliable Interactive Data Cleaning: A User Survey and Recommendations
Sanjay Krishnan, Daniel Haas, Michael J. Franklin, 2016

Data Validation is Hard

"How do you determine whether the data is sufficiently clean to trust the analysis?"

Other than common sense we do not have a procedure to do this.

We usually do not do rigorous validation of data cleaning. We typically clean our data until the desired analytics works without error. This is not desirable but practical since in most cases data error is probably overshadowed by errors/inaccuracies in the models themselves.

Towards Reliable Interactive Data Cleaning: A User Survey and Recommendations
Sanjay Krishnan, Daniel Haas, Michael J. Franklin, 2016

Your Story

Quick Question: Do you:

- **Always**
- **Sometimes**
- **Never**

add data validation to your notebooks, scripts or pipelines?

Image: Basketball Wives via Tumblr

Today’s Agenda

Data Validation 101: Definitions and Concepts

Data Validation and Testing with Python: Jupyter Workbooks & Active Learning

Data Unit Tests

Case Study: Data Validation with Machine Environment Data

What’s Happening in Academia?

How we'll learn

- Ask lots of questions
  - No such thing as a stupid or wrong question.
- Help your neighbors
  - Please say hi!
- Tell me when I make mistakes!
  - I am a human too :)

Data Validation 101

kjamistan

Data Quality Evaluation

- ❑ Valid
- ❑ Accurate
- ❑ Complete
- ❑ Consistent
- ❑ Uniform
- ❑ Repeatable

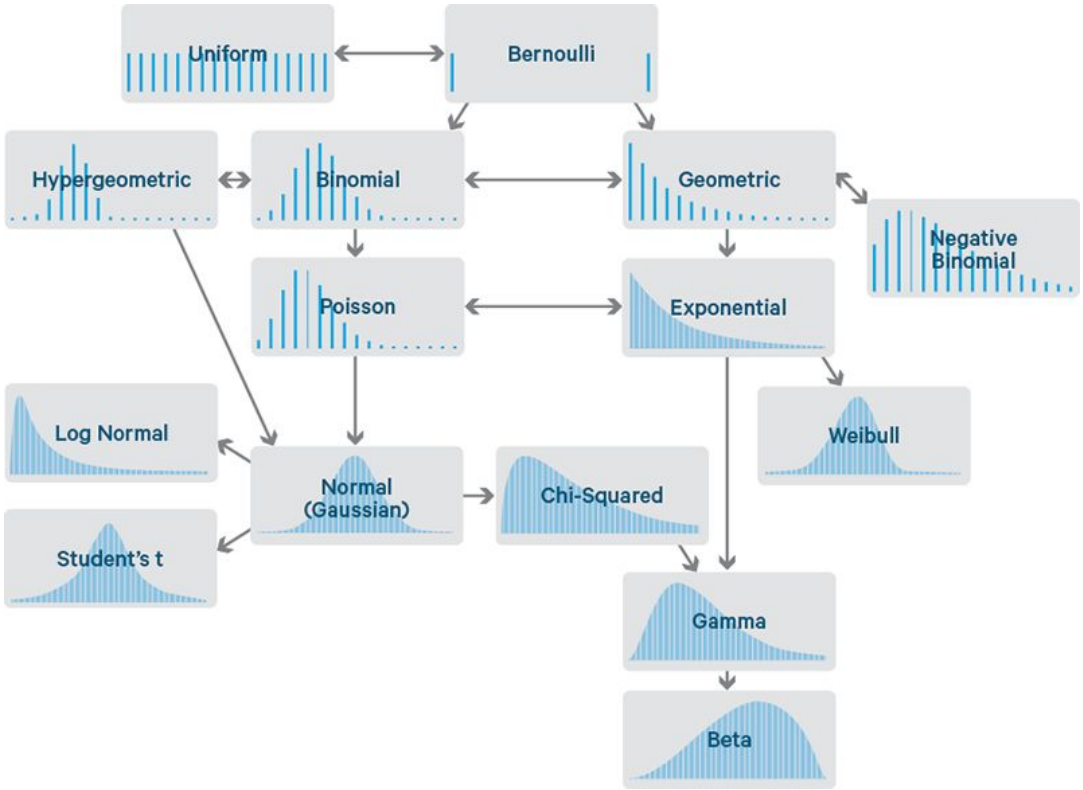
Data Reliability

- Correlation
  - More new users, more traffic and activity
- Temporal Stability
  - Traffic patterns
- Internal Consistency
  - More Clicks == More Page Views
- Determining good tests can be difficult, but useful

Data Validity

- Predictive Validity
  - Testing past models / Predictions
- Measurement or Metric Validity
  - What does it measure? How? Is it a valid measurement?
  - Fallacies of Metrics
- Trends or Noise?
  - Tracking patterns & outliers
  - Understanding biases
  - Handling Non-signals

Statistical Models & Measurements



Source: Cloudera
(http://blog.cloudera.com/blog/2015/12/common-probability-distributions-the-data-scientists-crib-sheet/)

kjamistan

Quick Poll

Statistical measurements or distributions can be used to verify my data:

- Often
- For some problems
- Almost Never

Outliers, Anomalies & Extreme Values

- Do normal outlier detection models work with your data?
- Can you easily predict normal values? (even if that means preprocessing and normalizing for day of week or seasonal trends)
- Do you throw out anomalies? How many? How come?
- Does anyone analyze average occurrence of anomalies or extreme values?

Margin of Error (and Team MoE)

- How relevant are accuracy, error and confidence measurements for your data problems (or models)?
- Team “Margin of Error”
  - What’s the margin of error you need to meet for business purposes (or internal and external needs)?
  - Do you have the ability to test new ideas, models or exploratory analysis and make mistakes somewhere (A/B testing, subsets and samples)?

Data Validation and Testing with Python

kjamistan

Following Along

- http://github.com/kjam/data-cleaning-101
- Python 3
- Requirements:
  - install_reqs.txt

- If you get lost, please ask (no stupid questions!)

Lesson One: Valid Values / Types

- Voluptuous
  - https://github.com/alecthomas/voluptuous
- (pip|conda) install voluptuous
- See also: https://github.com/guyskk/validr

Lesson Two: Dataframe Validation

- Engarde
  - https://github.com/TomAugspurger/engarde
- (pip|conda) install engarde
- See also: https://github.com/jnmclarty/validada

Lesson Three: TDDA

- Test-Driven Data Analysis
  - https://github.com/tdda/tdda

- (pip|conda) install tdda

- See also: https://docs.scipy.org/doc/scipy-0.19.0/reference/stats.html#statistical-functions

Lesson Four: Property-Based Tests

- Hypothesis
  - https://hypothesis.readthedocs.io/
- (pip|conda) install hypothesis
- See also: https://hackage.haskell.org/package/QuickCheck

Many More Libraries

- Schema Validation and Serialization:
  - https://marshmallow.readthedocs.io/en/latest/
  - For JVM / Apache: https://avro.apache.org/
- Model Validity
  - http://scikit-learn.org/stable/modules/cross_validation.html
- Testing ML features: https://github.com/machinalis/featureforge
- Built-in Stats: https://docs.python.org/3/library/statistics.html

Questions? (and a short break)

Source: https://gradientproductions.wordpress.com/

Data Unit Tests

Quick Poll

I write tests for my data science code:

- Every time
- Most of the time
- Occasionally
- Almost never

What is Unit Testing?

- Test a small unit of code
  - Define inputs, outputs and behavior
- Not for outside software, APIs or integration testing
  - Usually internal code and tools only
- Can exist in larger suites
- Often used with automated testing before releases (or even just on merges)
- Code Coverage

Why Test?

It is important to test your own code: don't assume that some testing organization or user will find things for you. But it's easy to delude yourself about how carefully you are testing, so try to ignore the code and think of the hard cases, not the easy ones. To quote Don Knuth describing how he creates tests for the TEX formatter, "I get into the meanest, nastiest frame of mind that I can manage, and I write the nastiest [testing] code I can think of; then I turn around and embed that in even nastier constructions that are almost obscene."

The Practice of Programming, Brian W. Kernighan and Rob Pike, 1999

Why Data Unit Testing?

- Data Science uses Code!
  - data engineering, pipelines, extraction
- Testing small units of data
  - Within expected ranges
  - Display expected heuristics (correlation)
  - Show anomalies or erratic behavior
- More data science code, means more (generally untested) code. There should be tests! ⚖️

BUT HOW?!?

Testing Basics

- Use libraries, don’t reinvent the wheel
- Learn about mocking outside APIs
  - https://docs.python.org/3/library/unittest.mock-examples.html
- Fake the data!
  - https://faker.readthedocs.io/en/master/
  - https://github.com/pereorga/csvfaker
- Watch Ned Batchelder’s testing talk: https://www.youtube.com/watch?v=FxSsnHeWQBY

How to Implement Testing

- Use Version Control
- Use Automated Testing
  - Pytest library: https://docs.pytest.org/en/latest/
  - Continuous Integration Tests (Jenkins, Travis, TeamCity, etc)
- Regular code reviews and merge procedure
  - http://www.bettercode.reviews/

Quick Poll

What is your experience with code reviews?

- Helpful, insightful
- Painful and embarrassing
- Useless, waste of time

Testing for Pipelines

- Does your framework have a built-in testing or validation toolset?
  - Testing Data Quality in Apache Spark: http://blog.cloudera.com/blog/2015/07/how-to-do-data-quality-checks-using-apache-spark-dataframes/
  - https://github.com/FRosner/drunken-data-quality
- Testing with Apache Beam: https://beam.apache.org/documentation/pipelines/test-your-pipeline/
- Tip: Check your framework's documentation first, then look for possible third-party fits

Testing Incoming (or existing) Data

- Tests:
  - Expected thresholds
  - Specific distributions or correlations
  - What to do when criteria not met?
- What determines an anomaly or outlier?
  - How are outliers handled?
- Automated “fixing”
  - Invalid data > Valid data via simple functions

Ok, I’m sold… But...

Testing & Validation Benefits

- Lost Revenue
  - Hours lost chasing bugs, outliers, bad data
  - Poor and costly decisions
  - Inaccurate predictions due to invalid data
- Gained Revenue
  - Happier employees and easier hires
  - Ability to automate (and TRUST) more workflows
  - Easier to find bugs
  - High priority to higher level thinking tasks

Questions? (and a short break)

Source: http://www.passportandtoothbrush.com

Case Study: Validating Router Environment Data

Defining the Problem

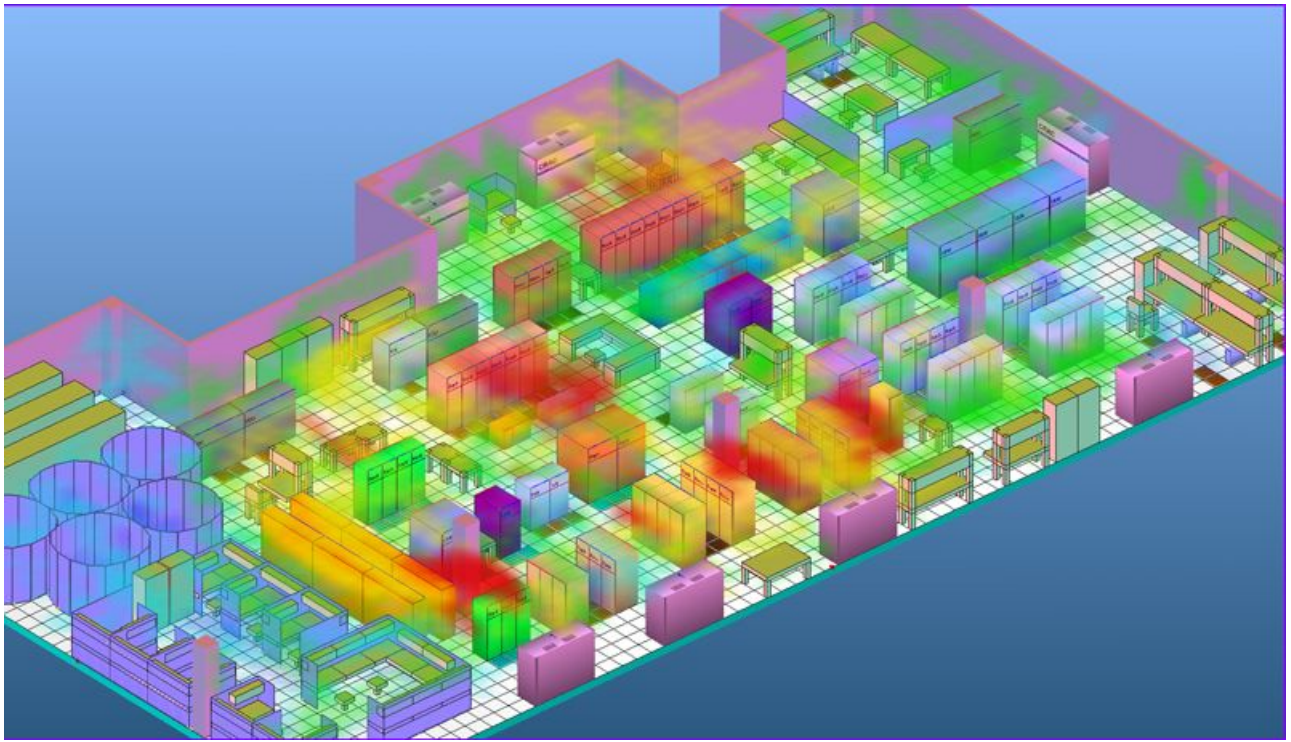


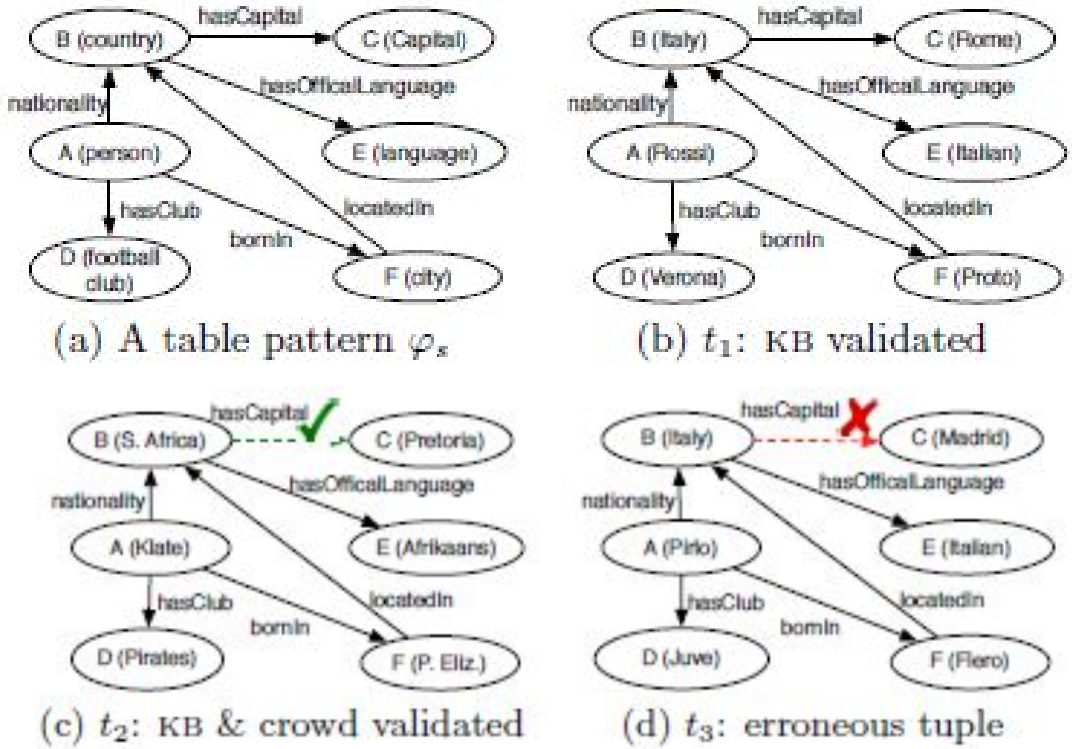
Image Source: http://tileflow.com/

Possible Solutions?

How might you validate incoming sensor data?

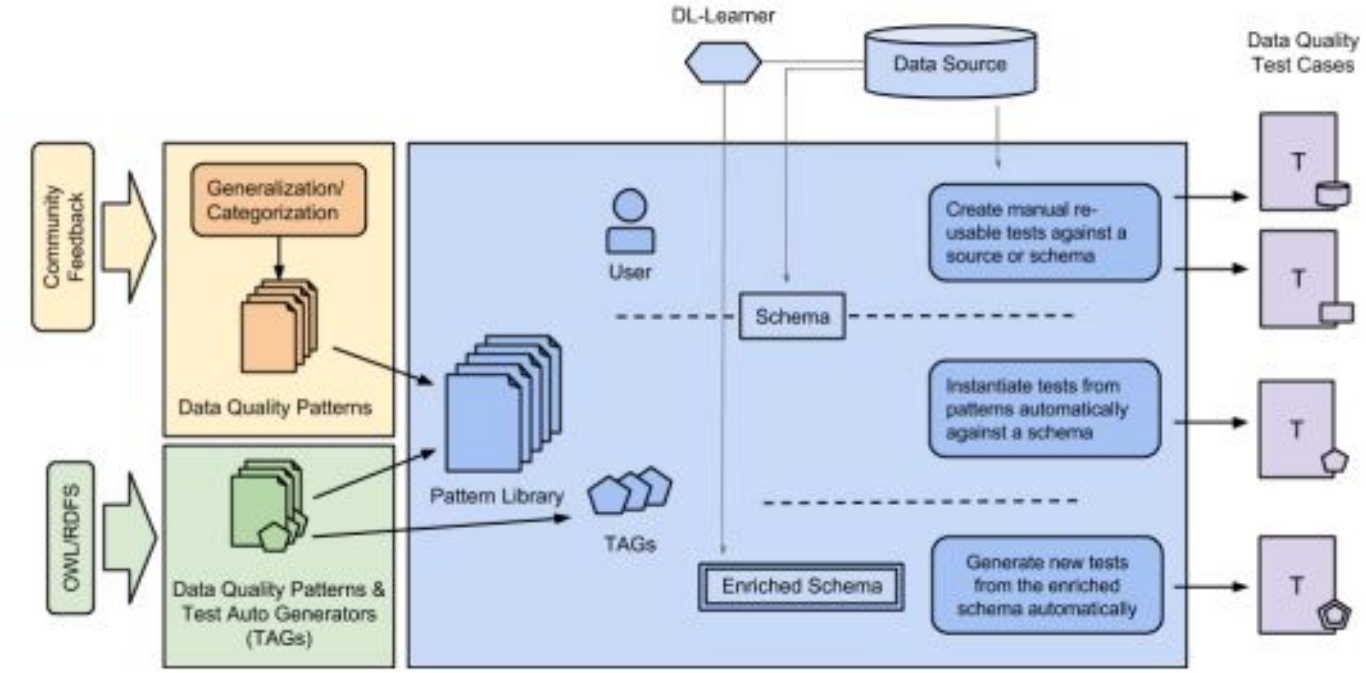
What’s Happening in Academia?

Katara: Crowd + IE



X Chu, Morcos, Ilyas et al. 2015

Databugger



D. Kontokostas, P. Westphal, S. Auer, S. Hellmann, J. Lehmann, R. Cornelissen, and A. Zaveri. (2015)

kjamistan

Unsupervised Anomaly Detection

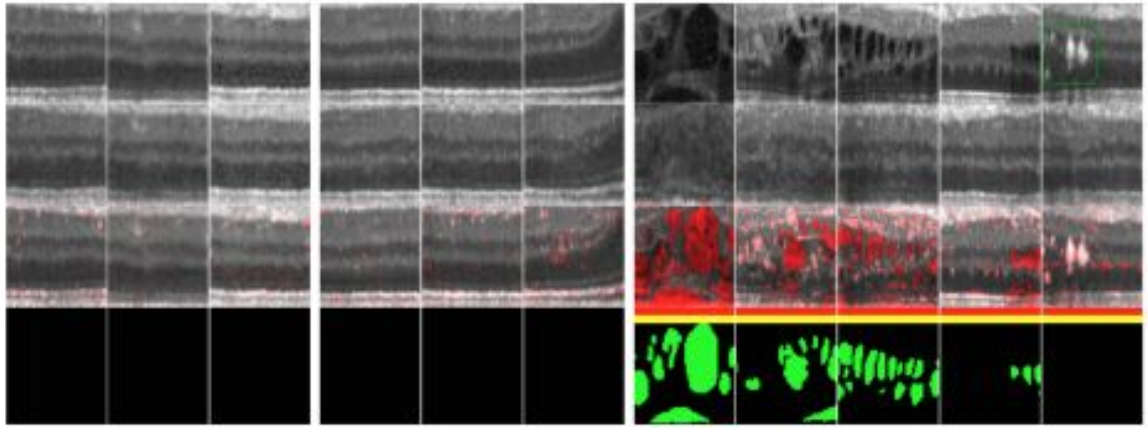


Fig. 3. Pixel-level identification of anomalies on exemplary images. First row: Real input images. Second row: Corresponding images generated by the model triggered by our proposed mapping approach. Third row: Residual overlay. Red bar: Anomaly identification by *residual score*. Yellow bar: Anomaly identification by *discrimination score*. Bottom row: Pixel-level annotations of retinal fluid. First block and second block: Normal images extracted from OCT volumes of healthy cases in the training set and test set, respectively. Third block: Images extracted from diseased cases in the test set. Last column: Hyperreflective foci (within green box). (Best viewed in color)

T Schlegl, P Seeböck, S M. Waldstein, U Schmidt-Erfurth, and G Langs (2017)

kjamistan

Quick Poll

How was your learning today?

Congrats! 🎉

THANK YOU!

- Resource post: https://blog.kjamistan.com/practical-data-cleaning-with-python-resources/
- If you can, please take a minute to give me some feedback
  - http://bit.ly/data-unit-testing-feedback
- Reach out anytime:
  - @kjam on Twitter / GitHub
  - katharine@kjamistan.com