Important Dates

- Assignment #1: Intro to Python – Opens 9/13
- Research Proposal Instructions will be handed out. (9/13)
- Assignment #2: Applying Analysis Techniques and Statistical Inference to Data: Opens 10/4
- Assignment #1: Due 10/4
- Students Research Proposals Due: (10/4)

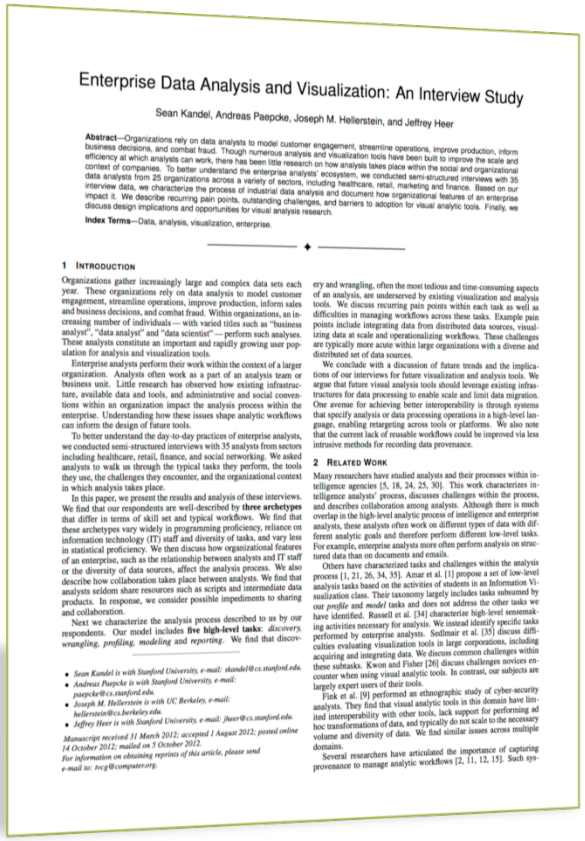
Agenda

- **Guest Lecture: Vlad Barash from Graphika**
- **Lecture 3: Data Munging**
- **Lab:**
  - **Web Scraping**
  - **EDA**
- **Go over Assignment 1.**
- **Go Over Project Proposal Instructions.**

So you want to be a data scientist…

How will you spend your time?

Enterprise Data Analysis and Visualization: An Interview Study



Enterprise Data Analysis and Visualization: An Interview Study

Sean Kandel, Andreas Paepcke, Joseph M. Hellerstein, and Jeffrey Heer

Abstract—Organizations rely on data analysts to model customer engagement, streamline operations, improve production, inform business decisions, and combat fraud. Though numerous analysis and visualization tools have been built to improve the scale and efficiency at which analysts can work, there has been little research on how analysis takes place within the social and organizational context of companies. To better understand the enterprise analysts' ecosystem, we conducted semi-structured interviews with 35 data analysts from 25 organizations across a variety of sectors, including healthcare, retail, marketing and finance. Based on our interview data, we characterize the process of industrial data analysis and document how organizational features of an enterprise impact it. We describe recurring pain points, outstanding challenges, and barriers to adoption for visual analytic tools. Finally, we discuss design implications and opportunities for visual analysis research.

Index Terms—Data, analysis, visualization, enterprise.

1 Introduction

Organizations gather increasingly large and complex data sets each year. These organizations rely on data analysis to model customer engagement, streamline operations, improve production, inform sales and business decisions, and combat fraud. Within organizations, an increasing number of individuals — with varied titles such as "business analyst", "data analyst" and "data scientist" — perform such analyses. These analysts constitute an important and rapidly growing user population for analysis and visualization tools.

Enterprise analysts perform their work within the context of a larger organization. Analysts often work as a part of an analysis team or business unit. Little research has observed how existing infrastructure, available data and tools, and administrative and social conventions within an organization impact the analysis process within the enterprise. Understanding how these issues shape analytic workflows can inform the design of future tools.

To better understand the day-to-day practices of enterprise analysts, we conducted semi-structured interviews with 35 analysts from sectors including healthcare, retail, finance, and social networking. We asked analysts to walk us through the typical tasks they perform, the tools they use, the challenges they encounter, and the organizational context in which analysis takes place.

In this paper, we present the results and analysis of these interviews. We find that our respondents are well-described by **three archetypes** that differ in terms of skill set and typical workflows. We find that these archetypes vary widely in programming proficiency, reliance on information technology (IT) staff and diversity of tasks, and vary less in statistical proficiency. We then discuss how organizational features of an enterprise, such as the relationship between analysts and IT staff or the diversity of data sources, affect the analysis process. We also describe how collaboration takes place between analysts. We find that analysts seldom share resources such as scripts and intermediate data products. In response, we consider possible impediments to sharing and collaboration.

Next we characterize the analysis process described to us by our respondents. Our model includes **five high-level tasks**: *discovery, wrangling, profiling, modeling* and *reporting*. We find that discovery and wrangling, often the most tedious and time-consuming aspects of an analysis, are underserved by existing visualization and analysis tools. We discuss recurring pain points within each task as well as difficulties in managing workflows across these tasks. Example pain points include integrating data from distributed data sources, visualizing data at scale and operationalizing workflows. These challenges are typically more acute within large organizations with a diverse and distributed set of data sources.

We conclude with a discussion of future trends and the implications of our interviews for future visualization and analysis tools. We argue that future visual analysis tools should leverage existing infrastructures for data processing to enable scale and limit data migration. One avenue for achieving better interoperability is through systems that specify analysis or data processing operations in a high-level language, enabling retargeting across tools or platforms. We also note that the current lack of reusable workflows could be improved via less intrusive methods for recording data provenance.

2 Related Work

Many researchers have studied analysts and their processes within intelligence agencies [5, 18, 24, 25, 30]. This work characterizes intelligence analysts' process, discusses challenges within the process, and describes collaboration among analysts. Although there is much overlap in the high-level analytic process of intelligence and enterprise analysts, these analysts often work on different types of data with different analytic goals and therefore perform different low-level tasks. For example, enterprise analysts more often perform analysis on structured data than on documents and emails.

Others have characterized tasks and challenges within the analysis process [1, 21, 26, 34, 35]. Amar et al. [1] propose a set of low-level analysis tasks based on the activities of students in an Information Visualization class. Their taxonomy largely includes tasks subsumed by our *profile* and *model* tasks and does not address the other tasks we have identified. Russell et al. [34] characterize high-level sensemaking activities necessary for analysis. We instead identify specific tasks performed by enterprise analysts. Sedlmair et al. [35] discuss difficulties evaluating visualization tools in large corporations, including acquiring and integrating data. We discuss common challenges within these subtasks. Kwon and Fisher [26] discuss challenges novices encounter when using visual analytic tools. In contrast, our subjects are largely expert users of their tools.

Fink et al. [9] performed an ethnographic study of cyber-security analysts. They find that visual analytic tools in this domain have limited interoperability with other tools, lack support for performing ad hoc transformations of data, and typically do not scale to the necessary volume and diversity of data. We find similar issues across multiple domains.

Several researchers have articulated the importance of capturing provenance to manage analytic workflows [2, 11, 12, 15]. Such sys-

- *Sean Kandel is with Stanford University, e-mail: skandel@cs.stanford.edu.*
- *Andreas Paepcke is with Stanford University, e-mail: paepcke@cs.stanford.edu.*
- *Joseph M. Hellerstein is with UC Berkeley, e-mail: hellerstein@cs.berkeley.edu.*
- *Jeffrey Heer is with Stanford University, e-mail: jheer@cs.stanford.edu.*

Manuscript received 31 March 2012; accepted 1 August 2012; posted online 14 October 2012; mailed on 5 October 2012.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

Interview study of 35 analysts

25 companies
Healthcare
Retail, Marketing
Social networking
Media
Finance, Insurance

Various titles
Data analyst
Data scientist
Software engineer
Consultant
Chief technical officer

Kandel et al. "Enterprise Data Analysis and Visualization: An Interview Study. IEEE Visual Analytics Science & Technology (VAST), 2012

http://db.cs.berkeley.edu/papers/vast12-interview.pdf

“I spend more than half of my time integrating, cleansing and transforming data without doing any actual analysis. Most of the time I’m lucky if I get to do any ‘analysis’ at all…

… Most of the time once you transform the data ... the insights can be scarily obvious.”

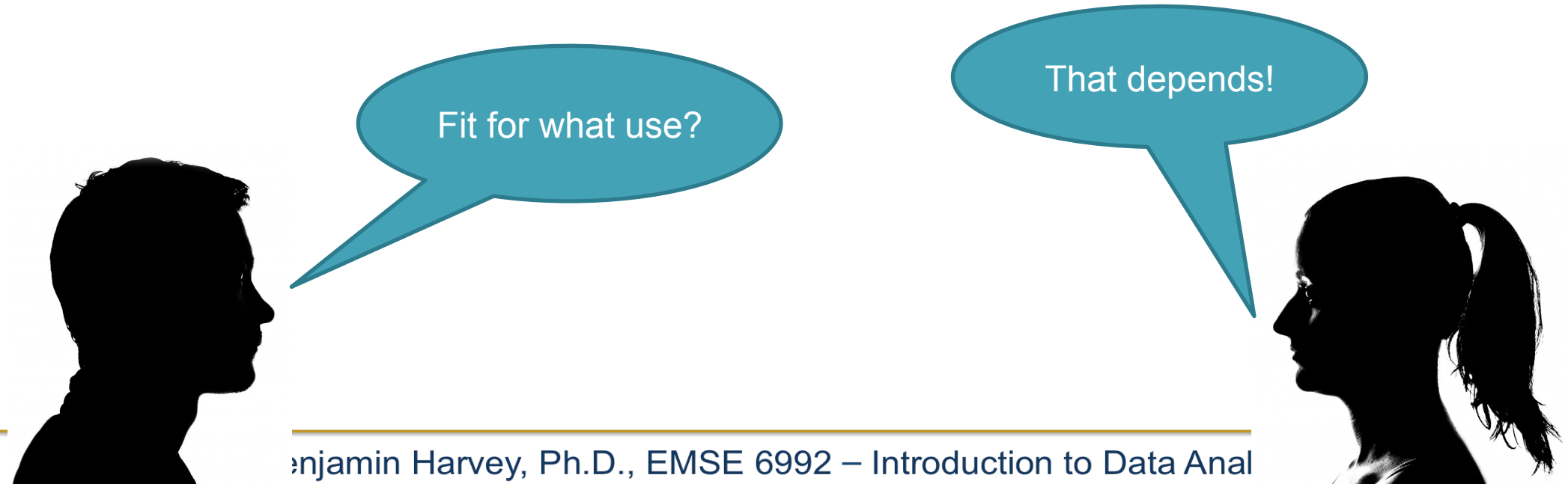
“Once you play with the data you realize you made an assumption that is completely wrong. It’s really useful, it’s not just a waste of time, even though you may be banging your head.”

“In practice it tends not to be just data prep, you are learning about the data at the same time, you are learning about what assumptions you can make.”

Data Wrangling

Aka Data Prep, Data Munging, Data Transformation

*Assessing and transforming raw data to make it **fit for use***



Data Wrangling

Aka Data Prep, Data Munging, Data Transformation

*Assessing and transforming raw data to make it **fit for use***

This is how you "get your head in the game"

- Understand what you have
- Assess strengths and weaknesses of your data
- Hypothesize about what to do with your data
- Get it ready

Nobody will know your data as well as you do while wrangling

- Not even the "you" of a few days later

Discussion

When is data “dirty”?

How does that happen?

Today: Data Unboxing and Wrangling

Basic tools:

- UNIX command line
- SublimeText editor

Trifacta: a free visual data wrangling tool

- Born at Berkeley/Stanford
- Codifies some good practices you can also follow “by hand”

Later: Python’s Pandas library

You may choose to use other tools too

- Good data scientists maintain a well-stocked toolbox

A bit of background before we jump in

Data Preparation overview

- ETL
  - We need to **extract** data from the **source(s)**
  - We need to **load** data into the **sink**
  - We need to **transform** data at the source, sink, or in a **staging area**

  - Sources: file, database, event log, web site, HDFS…
  - Sinks: Python, R, SQLite, RDBMS, NoSQL store, files, HDFS…

Data Preparation overview

- Process model
  - The construction of a new data preparation process is done in many phases
    - Data **characterization**
    - Data **cleaning**
    - Data **integration**
  - We must efficiently move data around in space and time
    - Data **transfer**
    - Data **serialization** and **deserialization** (for files or network)

Data Preparation overview

- Workflow
  - The transformation **pipeline** or **workflow** often consists of many steps
    - For example: Unix pipes and filters
    - $ cat data_science.txt | wc | mail -s "word count" myname@some.com
  - If the workflow is to be used more than once, it can be **scheduled**
    - Scheduling can be time-based or event-based
    - Use publish-subscribe to register interest (e.g. Twitter feeds)
  - Recording the execution of a workflow is known as capturing **lineage** or **provenance**

The Businessperson

- Data Sources
  - Web pages
  - Excel
- ETL
  - Copy and paste
- Data Warehouse
  - Excel
- Business Intelligence and Analytics
  - Excel functions
  - Excel charts
  - Visual Basic?!

The Programmer

- Data Sources
  - Web scraping, web services API
  - Excel spreadsheet exported as CSV
  - Database queries
- ETL
  - wget, curl, Beautiful Soup, lxml
- Data Warehouse
  - Flat files
- Business Intelligence and Analytics
  - Numpy, Matplotlib, R, Matlab

The Enterprise

- Data Sources
  - Application databases
  - Intranet files
  - Application server log files
- ETL
  - Informatica, IBM DataStage, Ab Initio, Talend
- Data Warehouse
  - Teradata, Oracle, IBM DB2, Microsoft SQL Server
- Business Intelligence and Analytics
  - Business Objects, Cognos, Microstrategy
  - SAS, SPSS, R

The Web Company

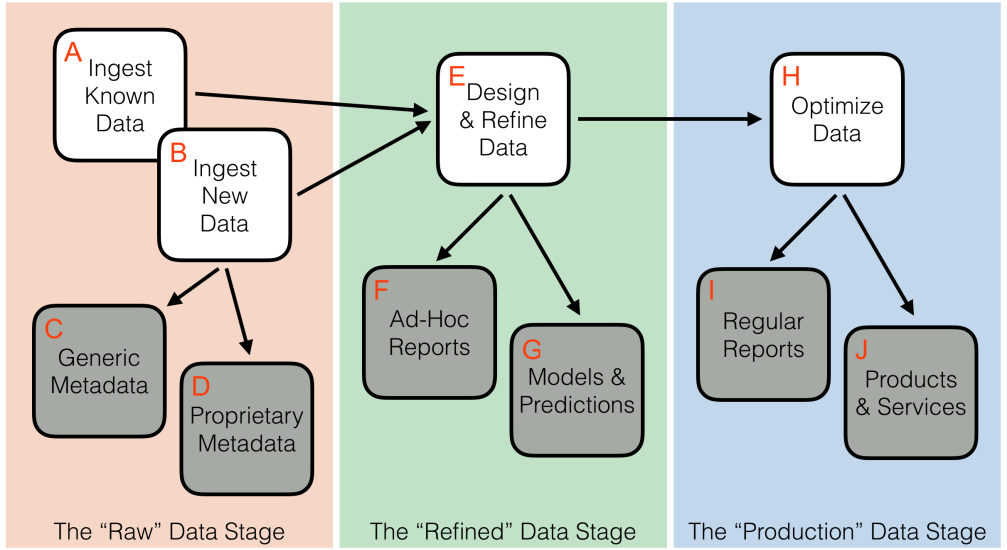
- Data Sources
  - Application databases
  - Logs from the services tier
  - Web crawl data
- ETL
  - Flume, Sqoop, Pig, Crunch, Oozie
- Data Warehouse
  - Hadoop/Hive, Spark/Shark
- Business Intelligence and Analytics
  - Custom dashboards: Argus, BirdBrain
  - R

Stages of Wrangling

Raw: Data ingestion & discovery ("unboxing")
- What: Exploratory *ad hoc* analysis
- Who: The individual wrangler

Refined: Curating data for reuse
- What: Data warehousing, canonical models
- Who: Data curators, IT engineers, actuaries, etc.

Production: Ensuring feeds and workflows
- What: Recurrent, automated use cases:
  - Traditional (e.g. reporting) + New (e.g. recommenders)
- Who: Often involves SW engineers and IT/ops folks



Rattenbury, et al. "Data Wrangling: Techniques and Concepts for Agile Analytics". To appear, O'Reilly Media, 2017.

Today

We will focus on the “Raw→Refined” stage

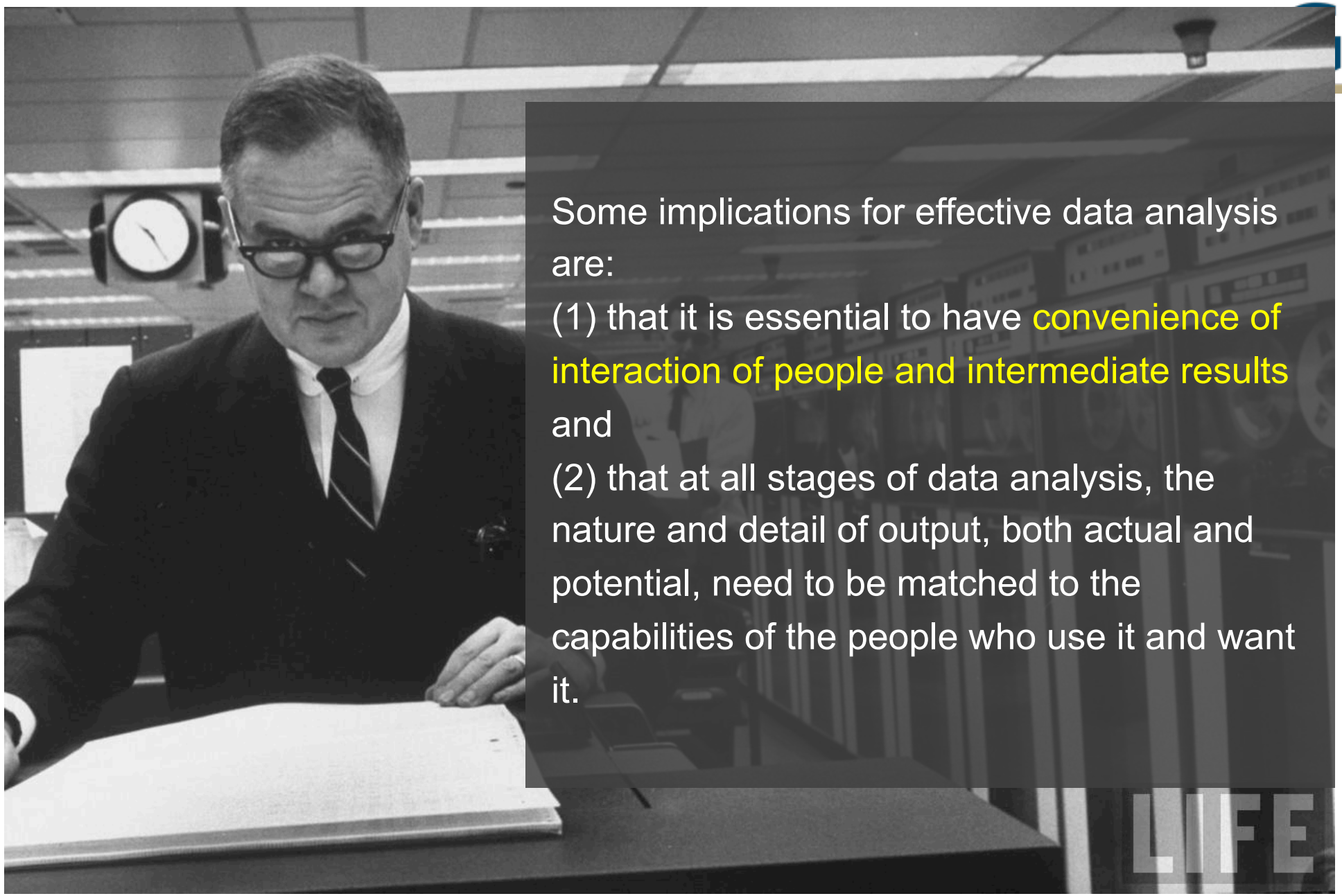
- Unboxing
- Transformation to analytics-ready structure
- Assessment/mitigation of quality issues

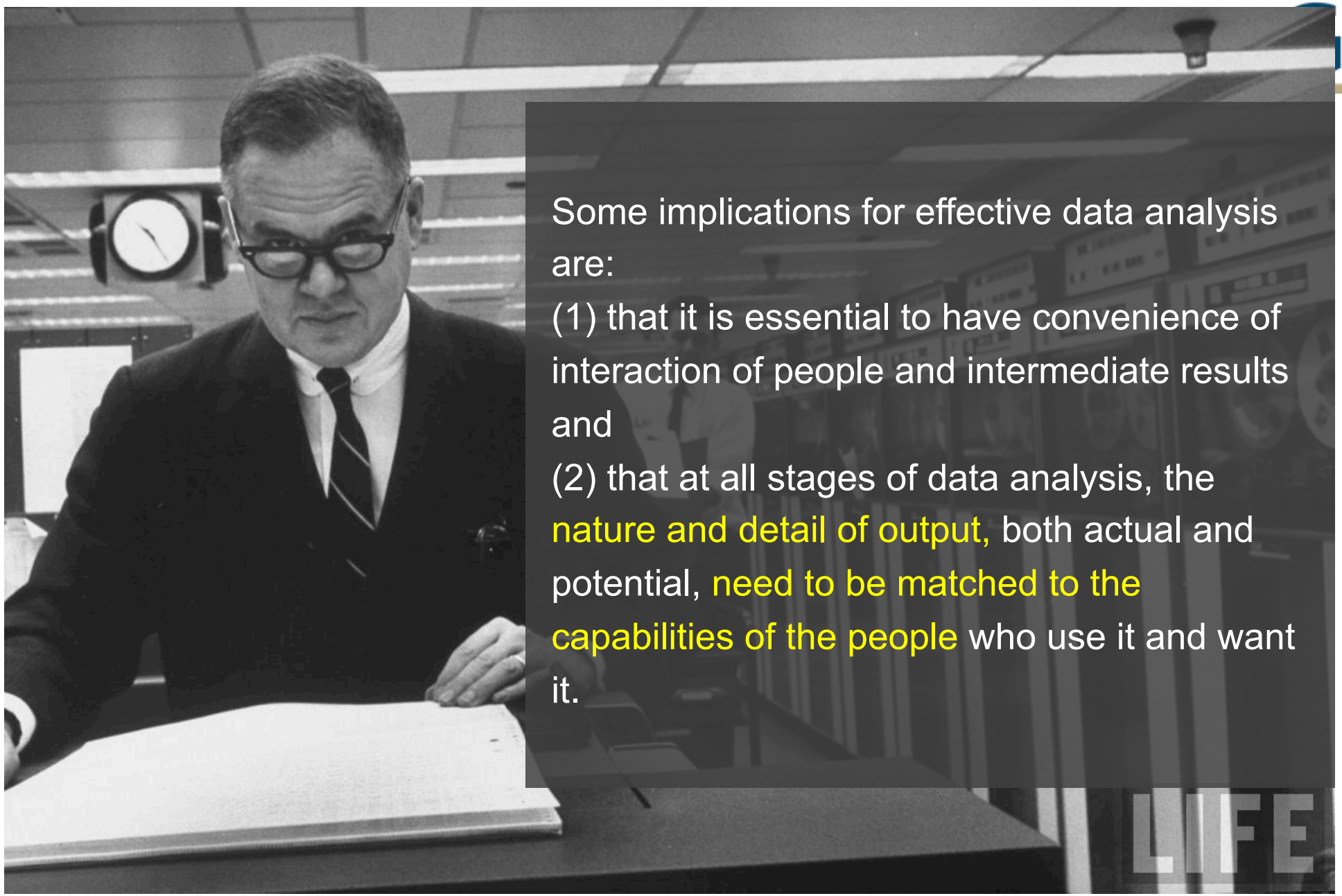
This is just a taste

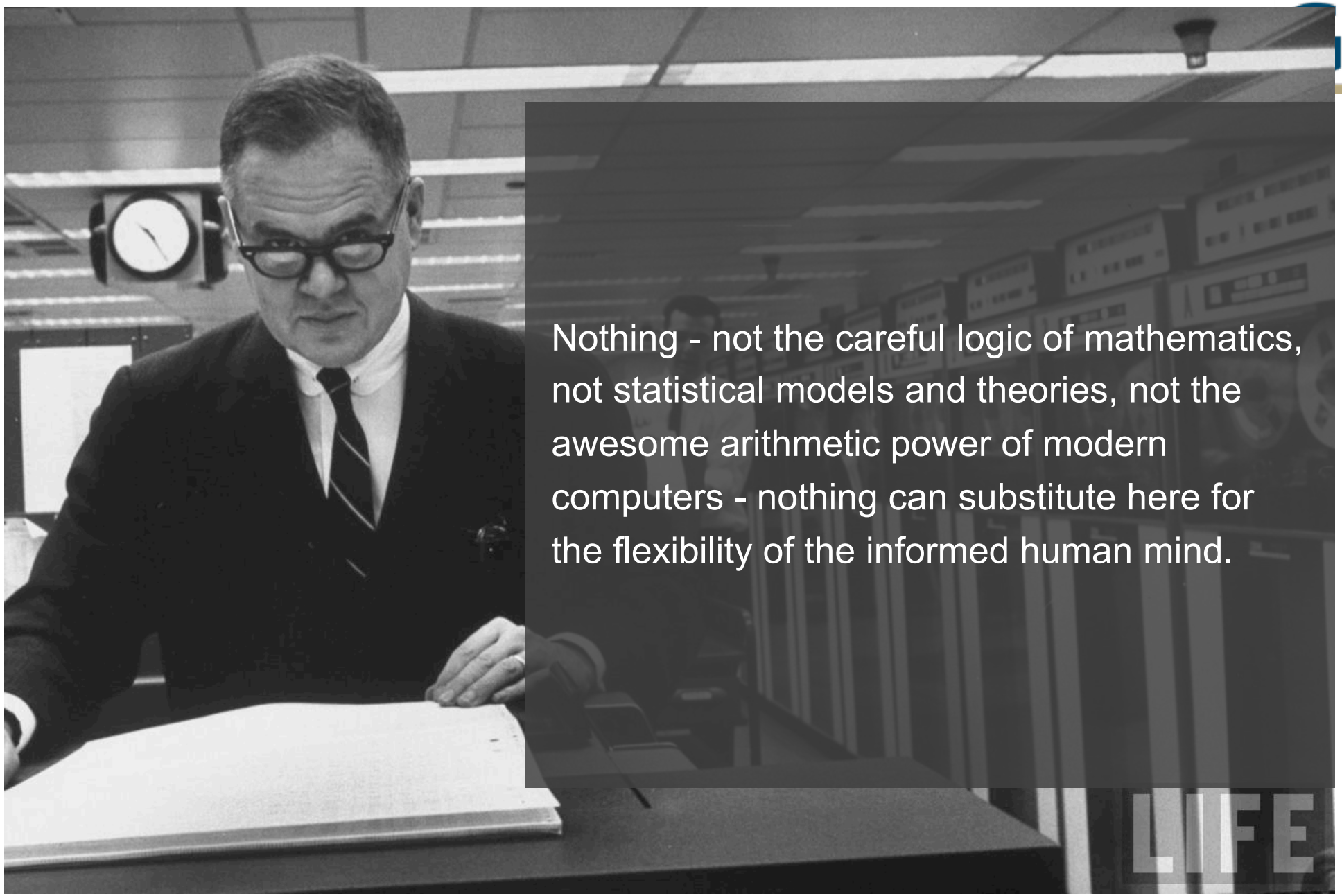
- Soon you will learn to do this in Python & Pandas

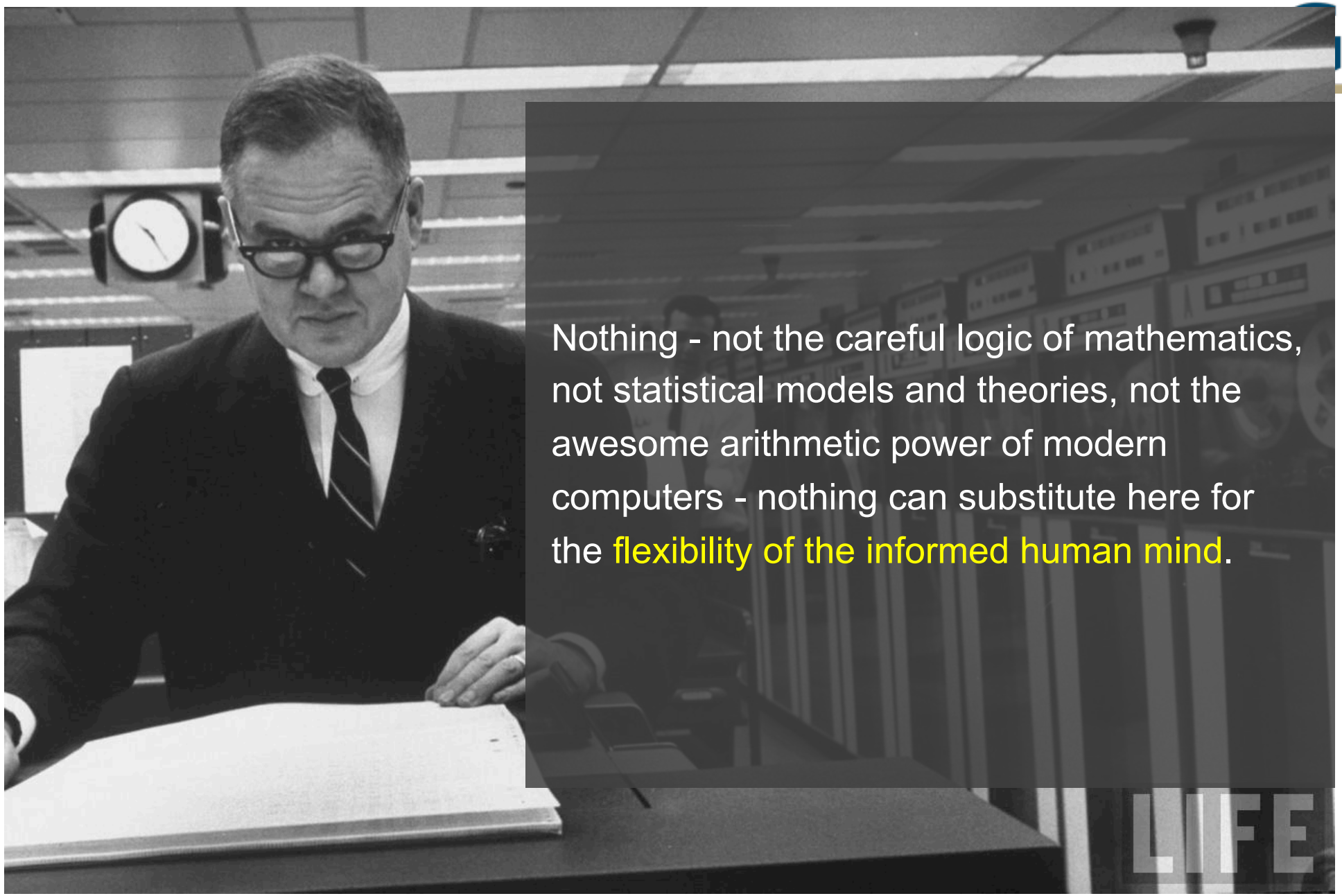
More on this in future lectures!

Data Analysis & Statistics, Tukey 1965


Some implications for effective data analysis are:
(1) that it is essential to have convenience of interaction of people and intermediate results and
(2) that at all stages of data analysis, the nature and detail of output, both actual and potential, need to be matched to the capabilities of the people who use it and want it.
LIFE


Some implications for effective data analysis are:
(1) that it is essential to have convenience of interaction of people and intermediate results and
(2) that at all stages of data analysis, the nature and detail of output, both actual and potential, need to be matched to the capabilities of the people who use it and want it.
LIFE


Nothing - not the careful logic of mathematics, not statistical models and theories, not the awesome arithmetic power of modern computers - nothing can substitute here for the flexibility of the informed human mind.
LIFE


Nothing - not the careful logic of mathematics, not statistical models and theories, not the awesome arithmetic power of modern computers - nothing can substitute here for the flexibility of the informed human mind.
LIFE

Unboxing Data

- What do I have here?
- What do I want to do with it?

These questions rarely have pat answers.

- Typically *contextual* and *user-driven*
- Typically subject to *iterative* cycles of wrangling and analysis

Rough Guide to Wrangling Issues

an outline for today's lecture

- Structure: the "shape" of a data file
- Granularity: how fine/coarse is each datum
- Faithfulness: how well does the data capture "reality"
- Temporality: how is the data situated in time
- Scope: how (in)complete is the data

Many of these are ***subjective*** qualities! Depend on *context*.

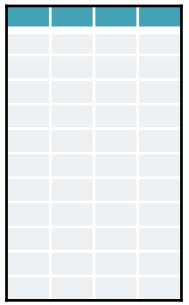
(Data) Science is a human process.

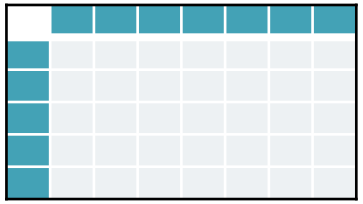
Structure: Rectangular Data

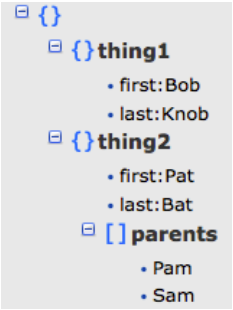
Natural for data analysis: easy to access, filter, tabulate

Two main variants

1. Relations (a.k.a. tables, data-frames)
   - Manipulate with Relational Algebra
2. Matrices
   - Manipulate with Linear Algebra



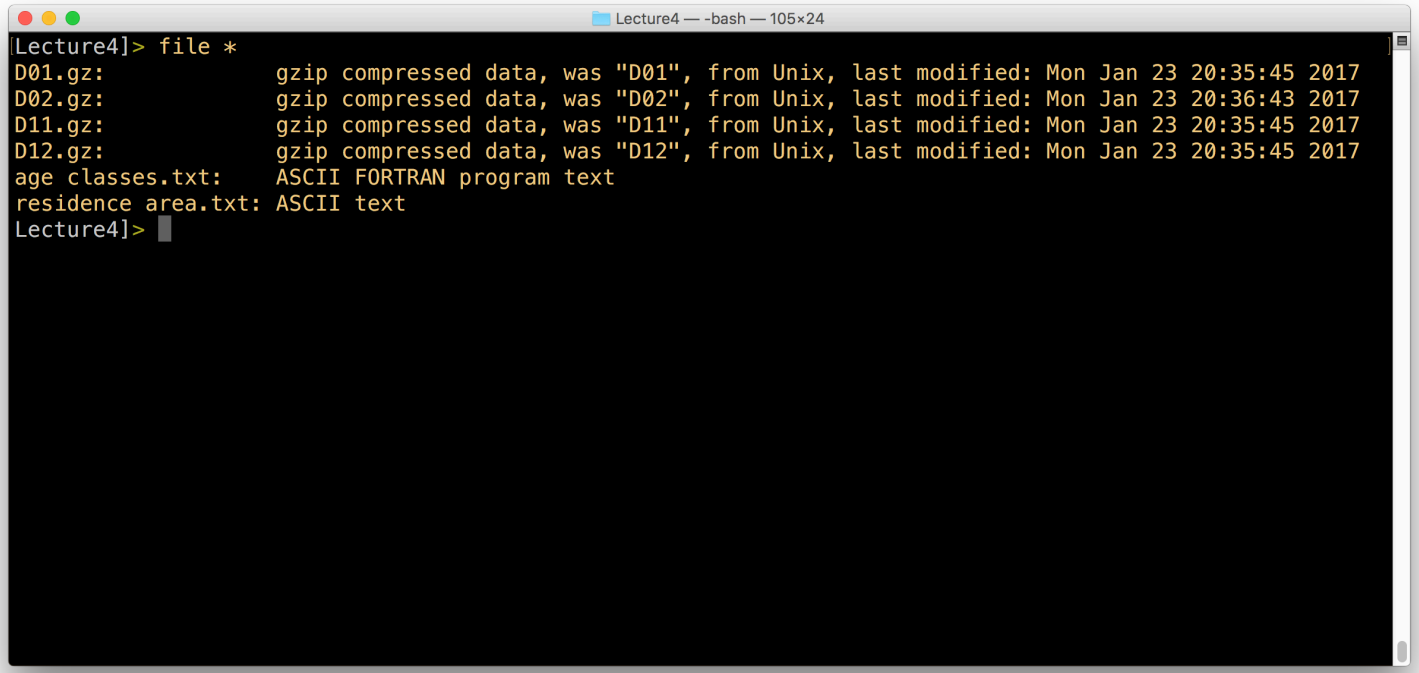




There are non-rectangular structures as well, e.g. JSON, XML.
For analysis, you will typically need to convert these to rectangular structures.

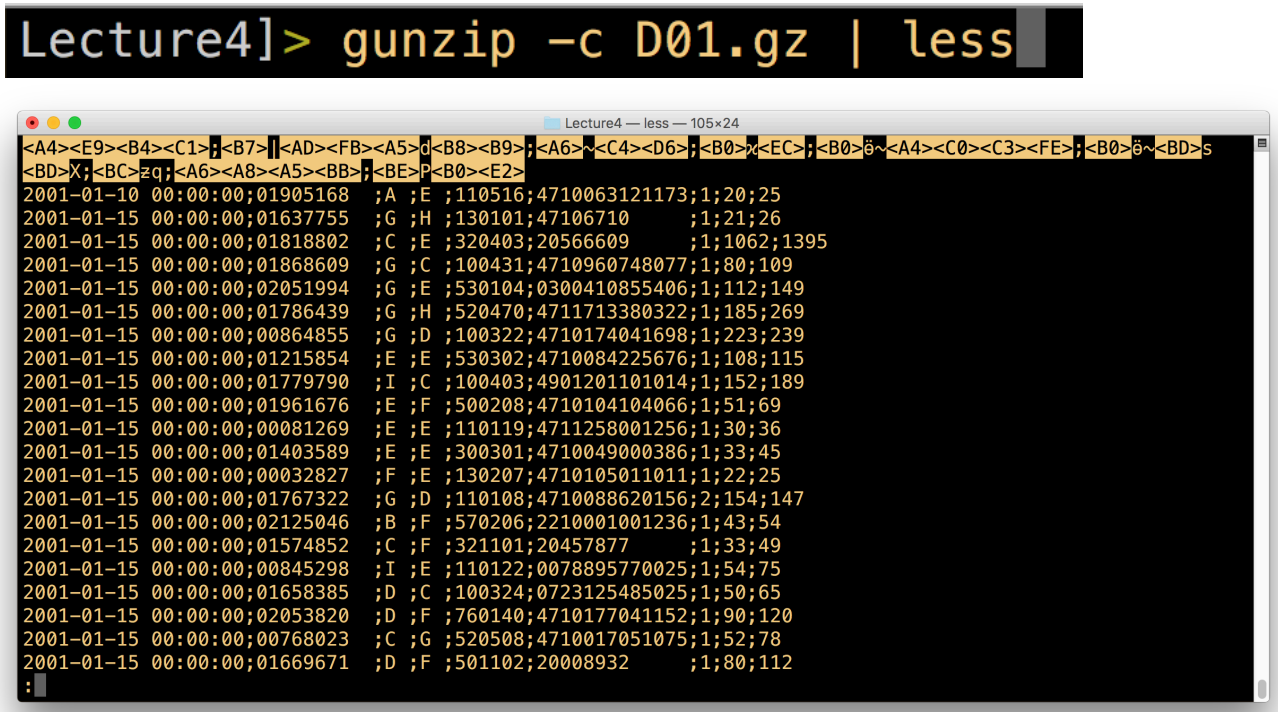
Unboxing with UNIX Command Line

- File metadata
  - `ls -lh`
  - `file`
  - `wc`
- File (de)compression
  - `gunzip`, `zip`, `bzip`, etc.
- `stdout` and the pipe
- File content:
  - `cat`
  - `head`
  - `tail`
  - `less`
  - <ctrl>-C



Structure Questions: A Checklist

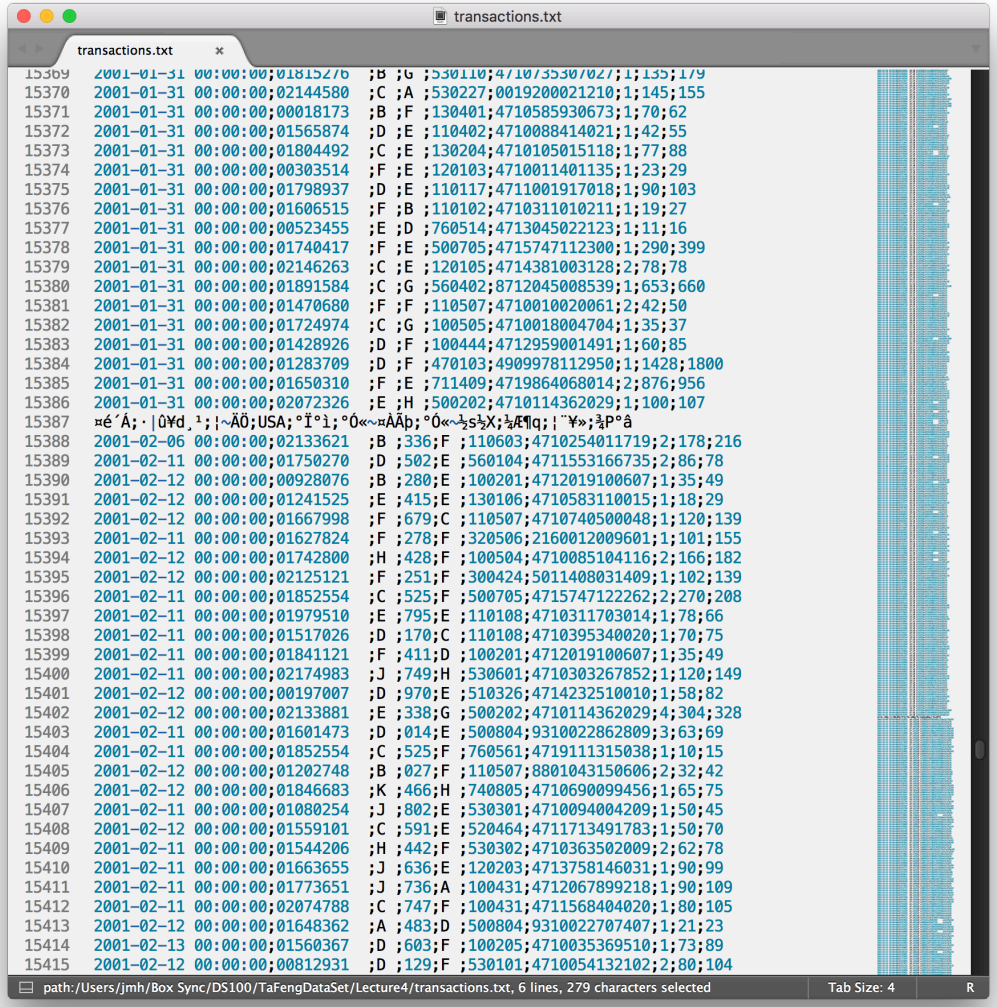
- Coarse structure
  - Is the data structured as a collection of *records*?
  - How are the individual records delimited in the dataset?
  - How are the record *fields* delimited from one another?
  - Do all records in the dataset contain the same fields?
  - How to access the same fields across records?
    By position? Name?
- Are the records nested?
  - No: one atomic (singular) value per field
  - Yes: collection of 0-to-many values in a field
- Encoding
  - How are values encoded? Strings? Codes? Binary?
  - How complex are the individual values?
    - Primitive: numbers and short strings?
    - Unstructured data: natural language text, audio, video
  - Can you decode?



Modern Text Editor: SublimeText

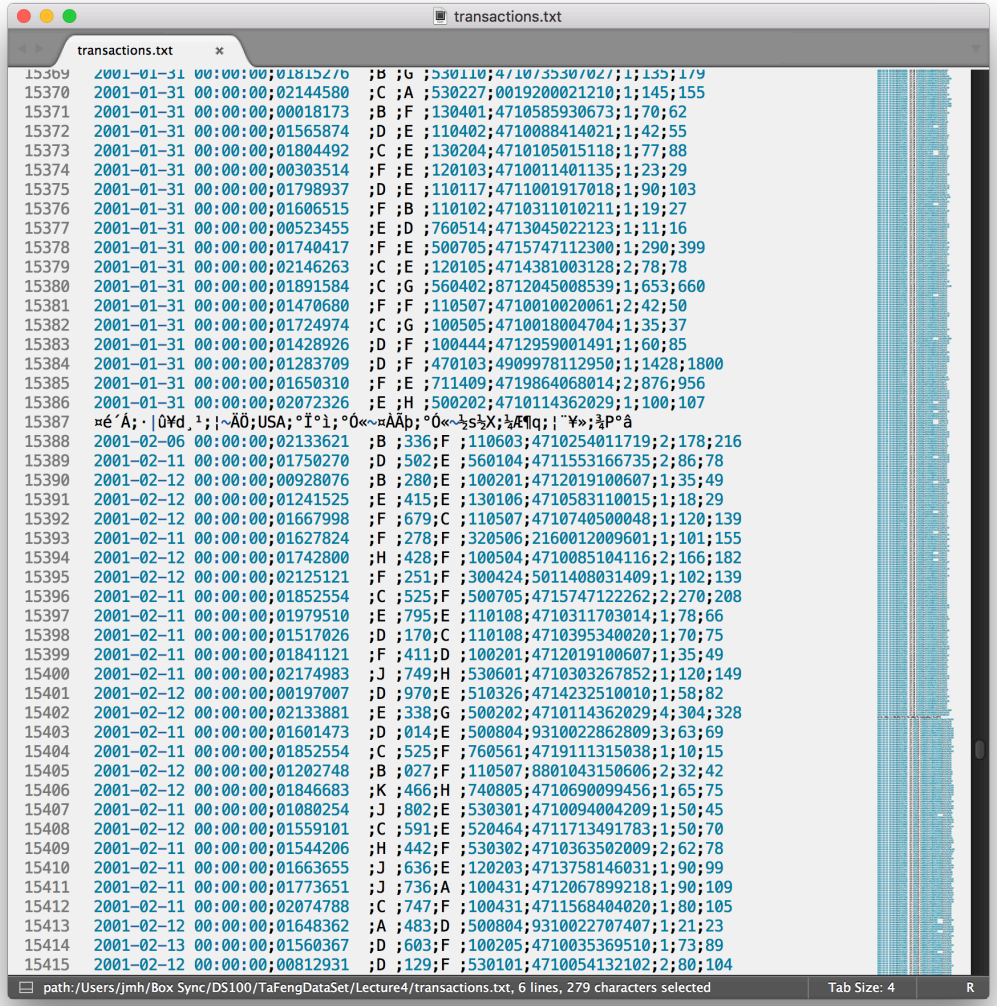
- Assessing
  - Coloring
  - Minimap
- Transformation
  - Do not destroy!
  - Names/Versions?

More on this soon!



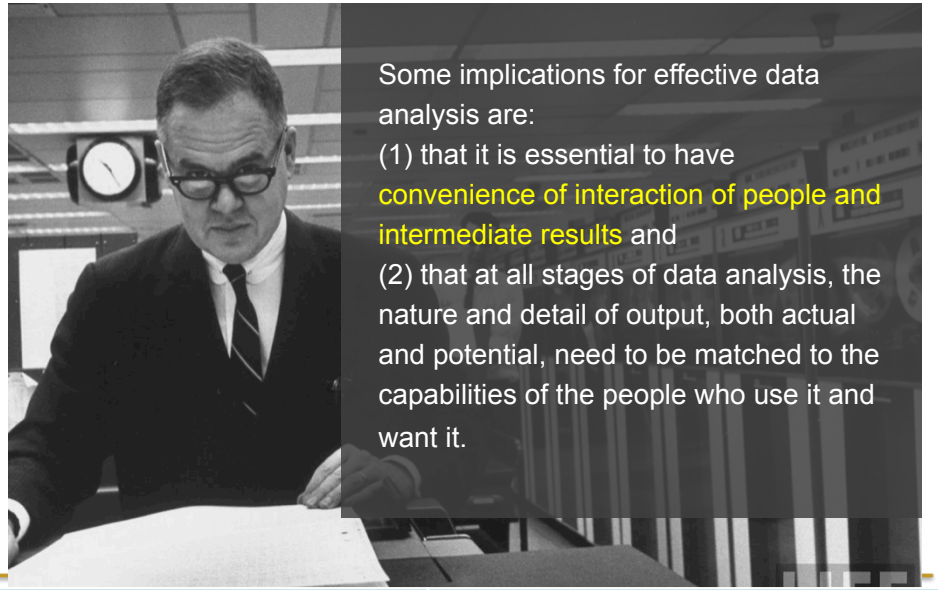
Structure Questions: A Checklist

- Coarse structure
  - Is the data structured as a collection of records?
  - How are the individual records delimited in the dataset?
  - How are the record *fields* delimited from one another?
  - Do all records in the dataset contain the same fields?
  - How to access the same fields across records?
    By position? Name?
- Is the data nested?
  - No: one atomic (singular) value per field
  - Yes: collection of 0-to-many values in a field
- Encoding
  - How are values encoded? Strings? Codes? Binary?
  - How complex are the individual values?
    - Primitive: numbers and short strings?
    - Unstructured data: natural language text, audio, video
  - Can you decode?



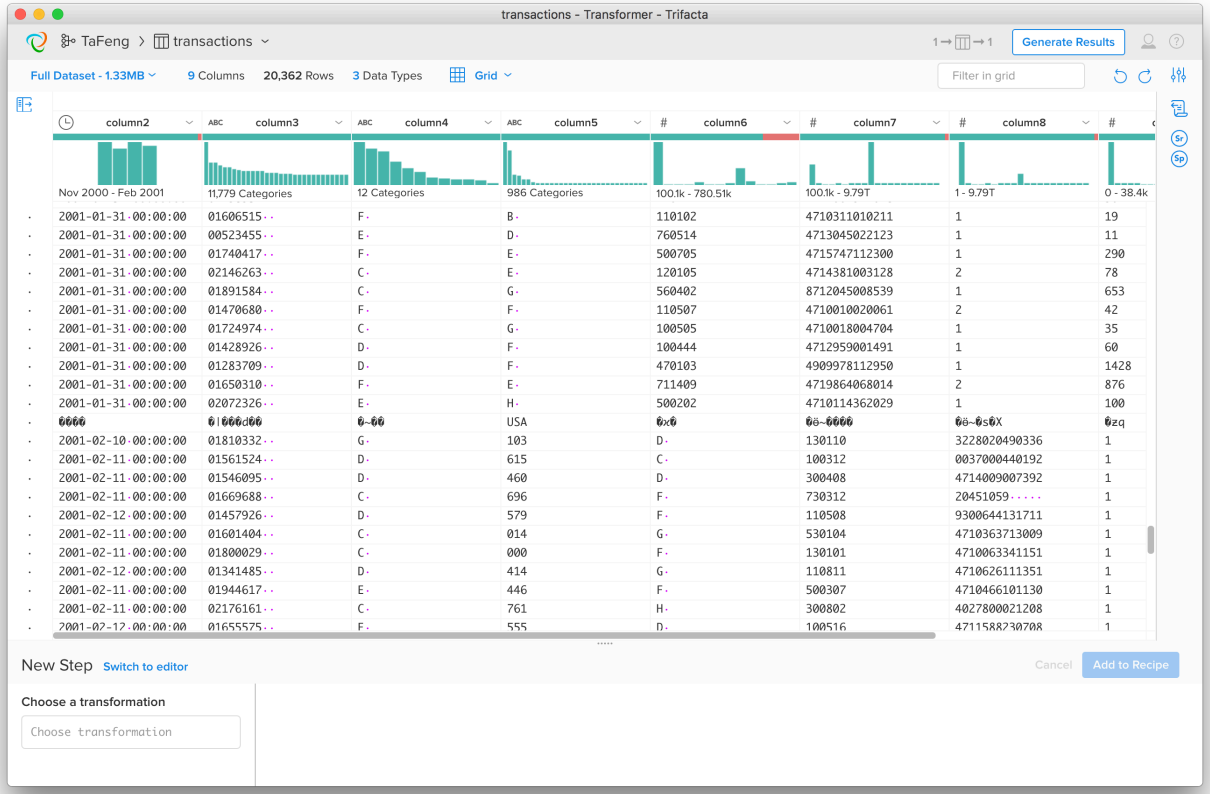
Unboxing with Trifacta

- Visual Profiling + Transformation
  - Iterative and ongoing
- Predictive Interaction
- Transformation language: Wrangle
  - Superset of relational algebra
  - Wrangle (and Pandas) both related to SQL
  - Generates a “recipe” (script)
    - You can run it on data to generate new data
- Scale & Interoperability
  - Paid version works with “big data”
    - Spark/Hadoop



Structure Questions: A Checklist

- Coarse structure
  - Is the data structured as a collection of records?
  - How are the individual records delimited in the dataset?
  - How are the record *fields* delimited from one another?
  - Do all records in the dataset contain the same fields?
  - How to access the same fields across records?
    By position? Name?
- Is the data nested?
  - No: one atomic (singular) value per field
  - Yes: collection of 0-to-many values in a field
- Encoding
  - How are values encoded? Strings? Codes? Binary?
  - How complex are the individual values?
    - Primitive: numbers and short strings?
    - Unstructured data: natural language text, audio, video
  - Can you decode?



So Far: Assessing Structure

- Basic tools
  - UNIX commands
  - Text editors
- Simple data transformation
  - Edit-in-place, character-by-character or Find/Replace
    - Save versions of files
  - Scripted transformations
    - Save scripts, inputs, outputs

Value Encodings: Primitive Types

Numbers and Strings

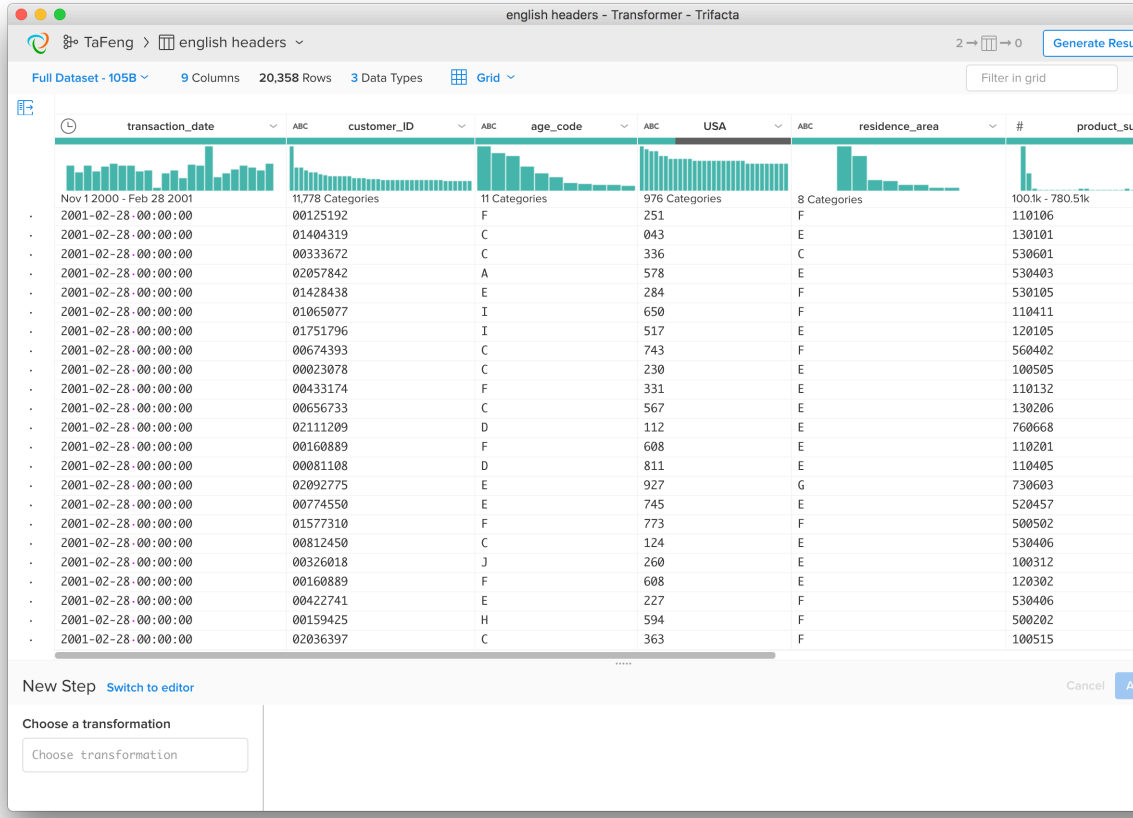
We'll spend time here today

Common special-case "primitives"

– Date/Time

– Geolocations

Space and time can be surprisingly subtle/messy!

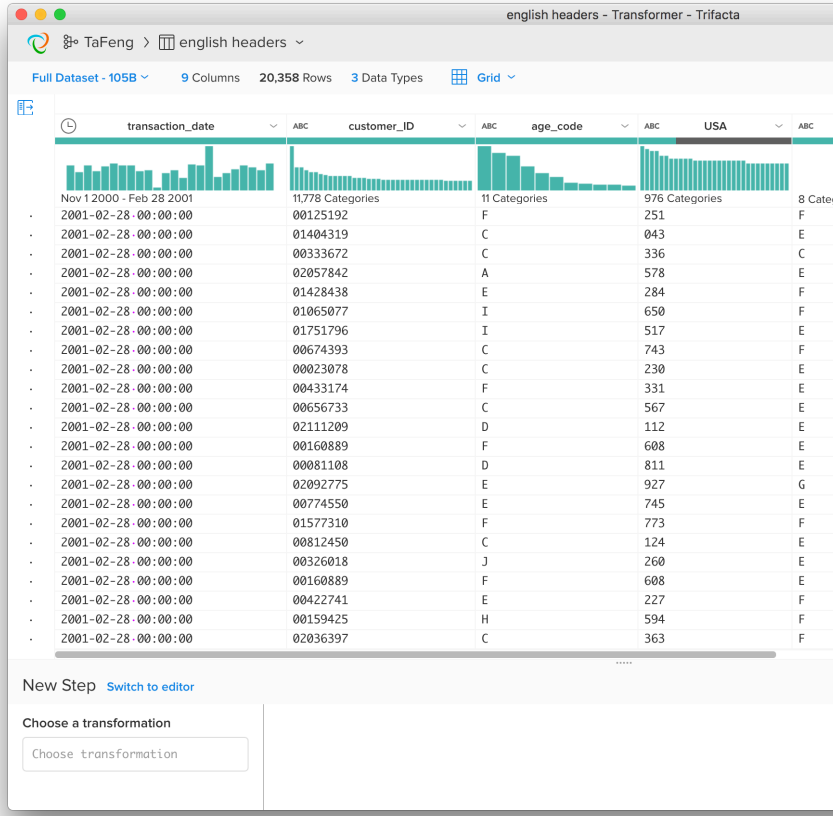


Primitive Type Semantics

- Categorical #1: Nominal
  - E.g. political party affiliation
  - Note: Sometimes even numeric data is nominal
- Categorical #2: Ordinal
  - E.g. education level (none, HS, College, Post-College)
  - Ordered, but not particularly *quantitative*
- Quantitative: amounts, measures
  - Negative or positive
  - Arithmetic “makes sense” (e.g. difference, ratio)
  - Integers vs. Rational numbers

Who cares?

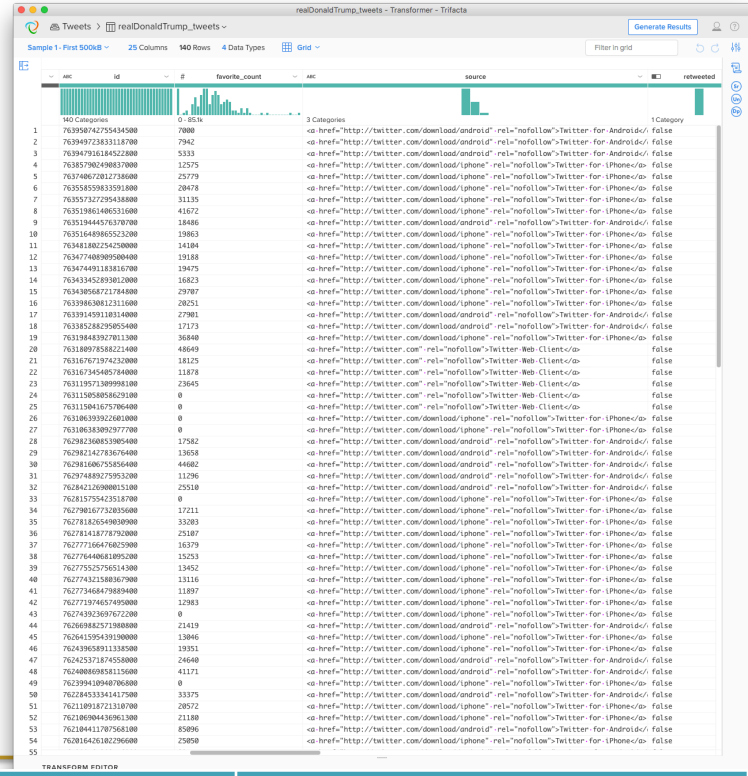
- Affects how you interpret the data
- E.g. In visualization or summarization



In Sum: Tools and Structure

- Coarse structure
  - Identifying data “shape”, records/fields and delimiters
  - Value encodings
  - Categorical, Ordinal, Quantitative
- Basic tools
  - UNIX commands
  - Text editors
- Simple data transformation
  - Edit-in-place, character-by-character or Find/Replace
  - Scripted transformations

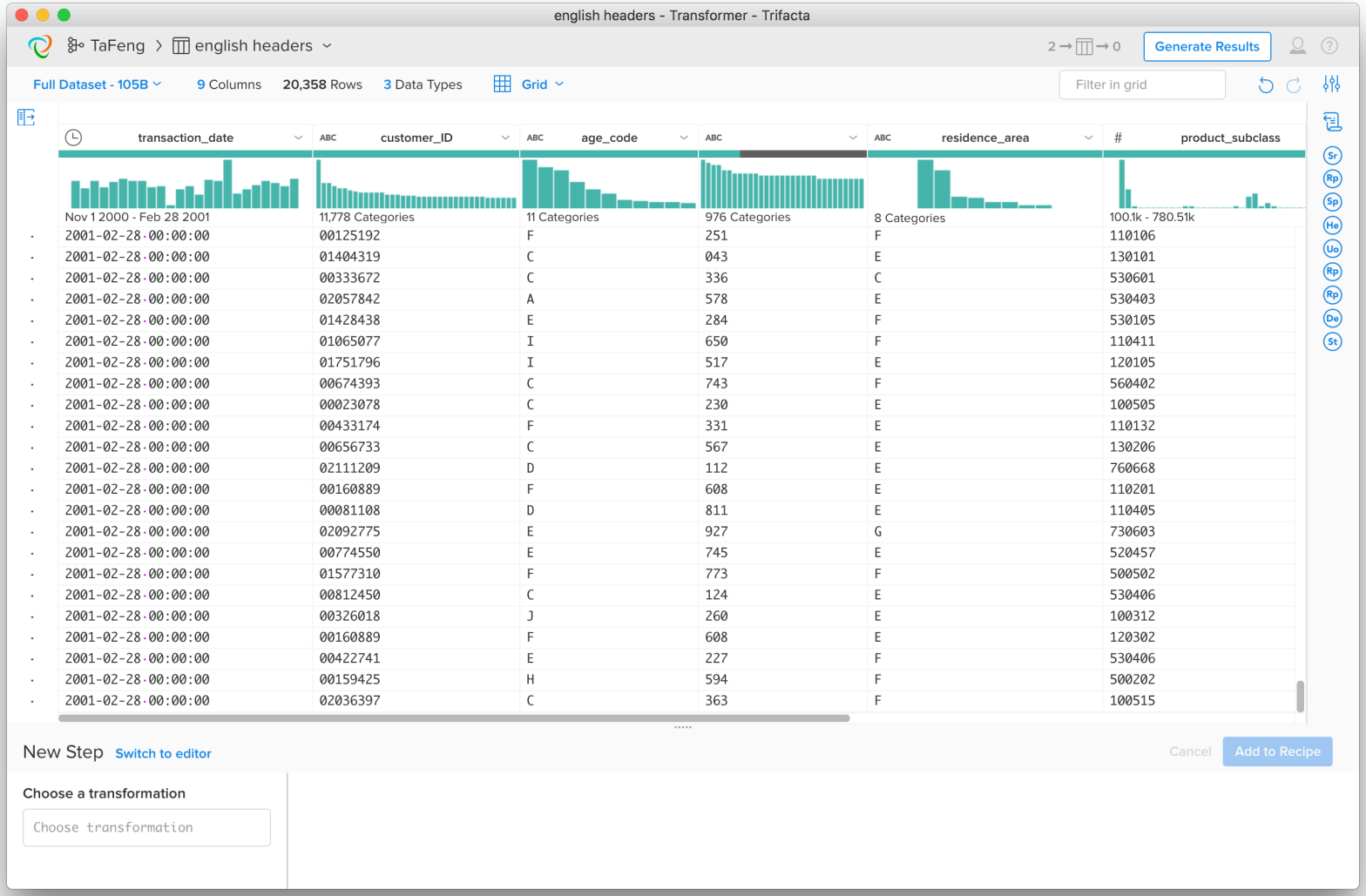
Assessing Granularity



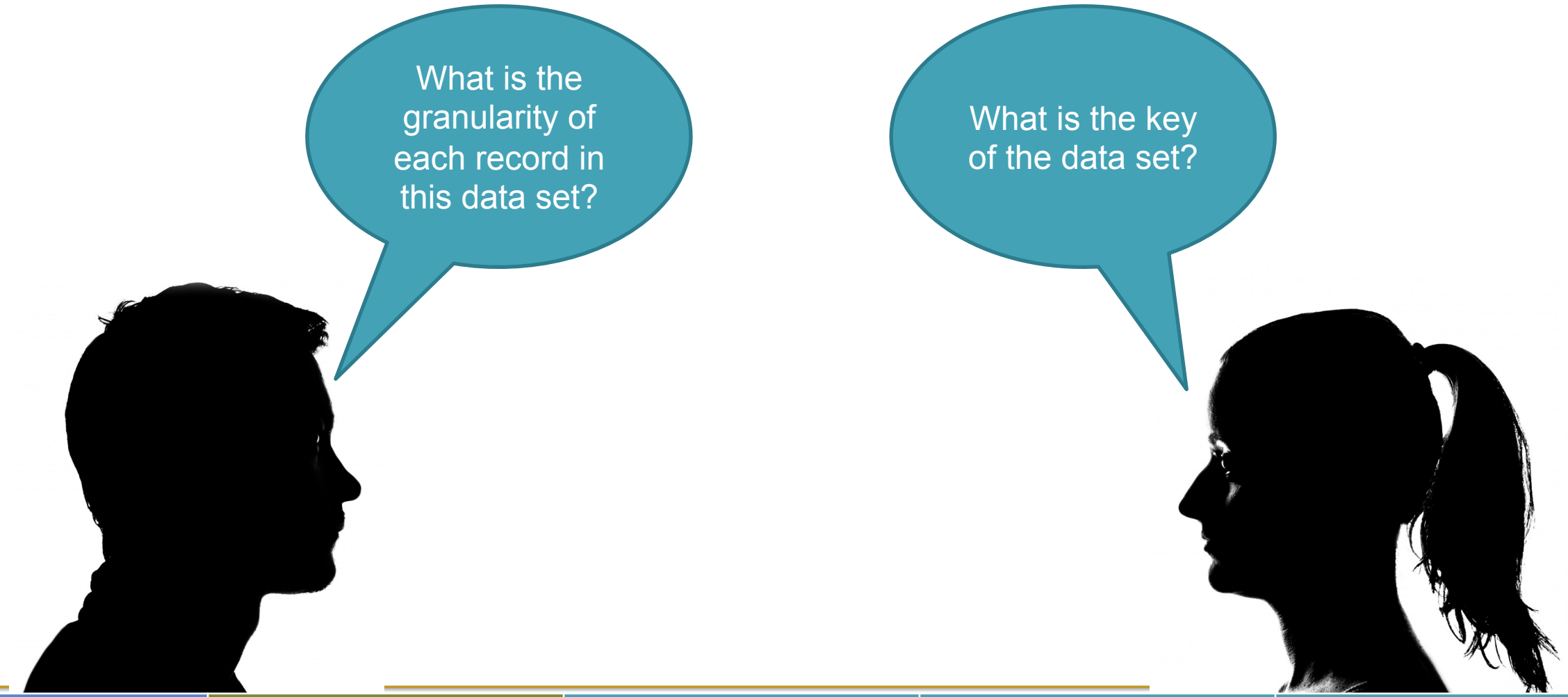
What does each TaFeng record represent?

Theories?

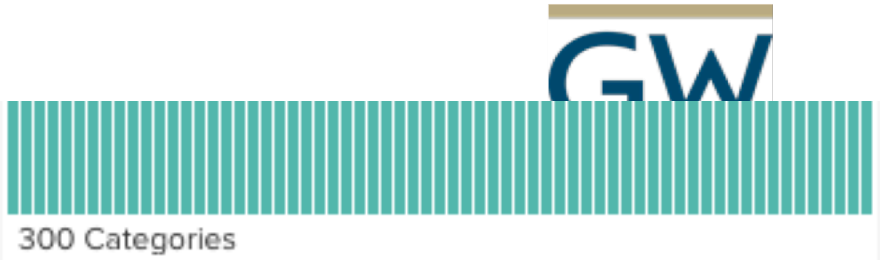
Justification?



Assessing Granularity



Key ("primary key")



Given a relation, a key attribute uniquely determines the values in each record

- E.g. an *identifier* like `transaction_id`
- Can be *composite*: e.g. `(City, State)`

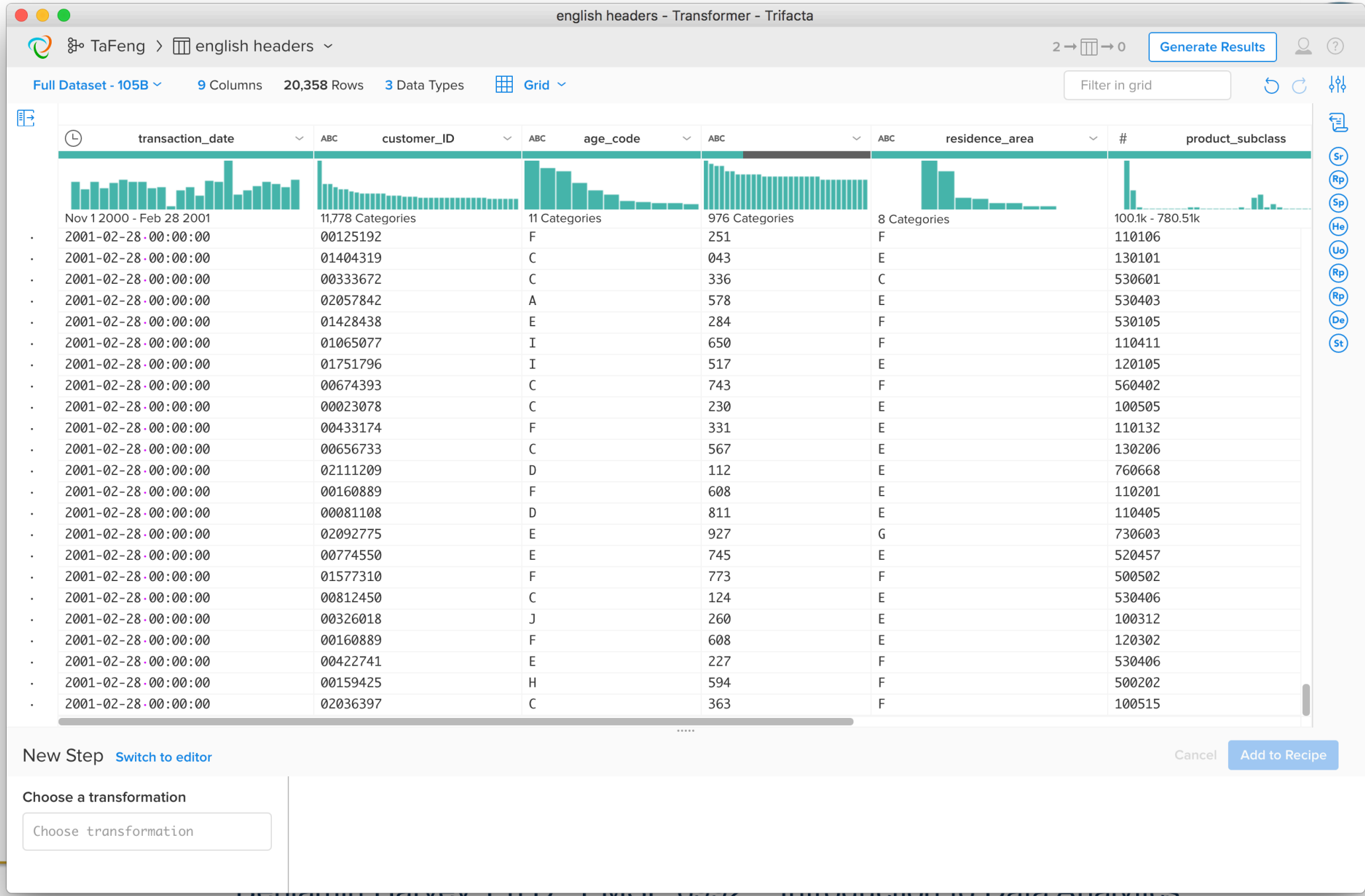
Each value occurs **at most once** in the key column(s).

Hence the semantics (meaning) of the key determines granularity

- `customer_id?` `transaction_time?`
- `(age_code, day_of_week)`

Be careful!

- Goofy key choice? Goofy data.
- Real-world data may have "noisy", duplicated keys to clean up


english headers - Transformer - Trifacta
TaFeng
english headers
2 → 0
Generate Results
Full Dataset - 105B
9 Columns
20,358 Rows
3 Data Types
Grid
Filter in grid
transaction_date
customer_ID
age_code
residence_area
product_subclass
Nov 1 2000 - Feb 28 2001
11,778 Categories
11 Categories
976 Categories
8 Categories
100.1k - 780.51k
2001-02-28 00:00:00 00125192 F 251 F 110106
2001-02-28 00:00:00 01404319 C 043 E 130101
2001-02-28 00:00:00 00333672 C 336 C 530601
2001-02-28 00:00:00 02057842 A 578 E 530403
2001-02-28 00:00:00 01428438 E 284 F 530105
2001-02-28 00:00:00 01065077 I 650 F 110411
2001-02-28 00:00:00 01751796 I 517 E 120105
2001-02-28 00:00:00 00674393 C 743 F 560402
2001-02-28 00:00:00 00023078 C 230 E 100505
2001-02-28 00:00:00 00433174 F 331 E 110132
2001-02-28 00:00:00 00656733 C 567 E 130206
2001-02-28 00:00:00 02111209 D 112 E 760668
2001-02-28 00:00:00 00160889 F 608 E 110201
2001-02-28 00:00:00 00081108 D 811 E 110405
2001-02-28 00:00:00 02092775 E 927 G 730603
2001-02-28 00:00:00 00774550 E 745 E 520457
2001-02-28 00:00:00 01577310 F 773 F 500502
2001-02-28 00:00:00 00812450 C 124 E 530406
2001-02-28 00:00:00 00326018 J 260 E 100312
2001-02-28 00:00:00 00160889 F 608 E 120302
2001-02-28 00:00:00 00422741 E 227 F 530406
2001-02-28 00:00:00 00159425 H 594 F 500202
2001-02-28 00:00:00 02036397 C 363 F 100515
New Step
Switch to editor
Cancel
Add to Recipe
Choose a transformation
Choose transformation

Joining tables with keys

- Keys in our data
  - `age classes.txt: code`
  - `residence area.txt: code`
  - `transactions.txt: transaction_id`

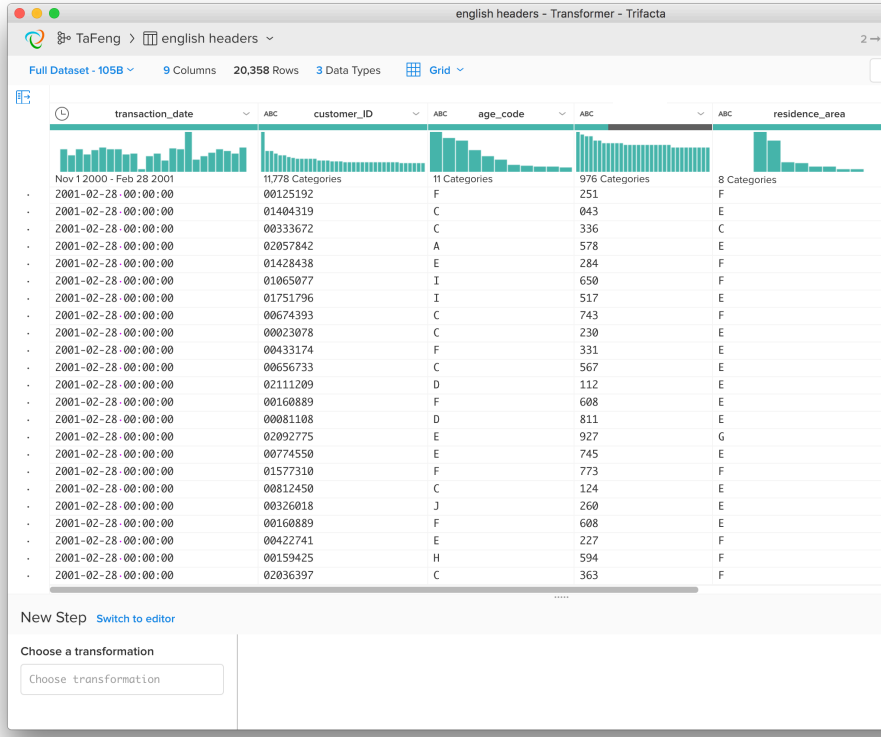
- Transactions.txt also has *foreign keys*
  - i.e. attributes that reference the key of another table
  - `age_code` is a foreign key to `age classes.txt`
  - `residence_area` is a foreign key to `residence area.txt`

- Joining on a foreign key is a "lookup"
  - At most one match for each row of `TaFengTransactions.txt`

More on this in future lectures!

Granularity Questions: a Checklist

- What kind of thing does each record represent?
- Do all records capture granularity at the same level?
- What alternative interpretations of the records are there?
  - E.g. Is this a list of transactions?
    Or a list of customers?
- What kinds of aggregation is possible/desirable?
  - From individual people to demographic groups?
  - From individual events to totals across time or regions?
  - Hierarchies (city/county/state, second/minute/hour/day)



So Far: Assessing Granularity

Identifying (primary) keys can help assess Granularity

- Each record is the information *per key*

Foreign Keys are pointers to individual rows in other data sets

- To lookup a row in another table: join the foreign key to its primary key

Assessing Faithfulness

Theme: the Faithfulness of a record can only be evaluated in context

- Application context
- Context in your data set
  - Across records

Students

id: integer	DOB: date	GPA: float	Risk: float
123457	01/16/1997	3.2	465
123458	01/24/2017	2.7	28
123457	01/16/2002	5.0	27
123459	03/22/1996	3.6	31
123460	06/13/1997	2.2	43

Faithfulness Across Records: Outliers

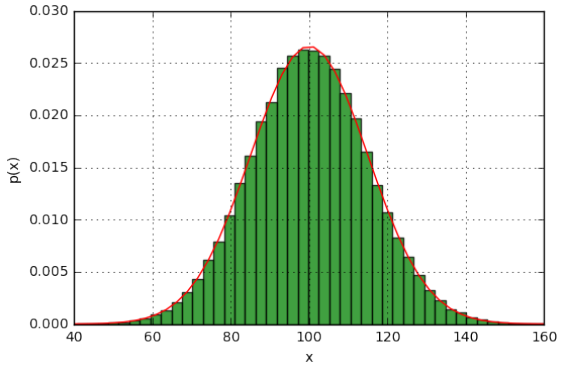
What is an “outlier”?

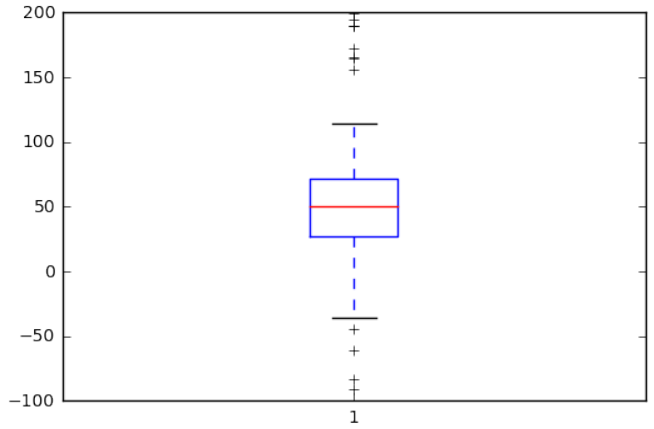
More on this in future lectures!

- A value that is “far” from the “center”

Distribution-based definition

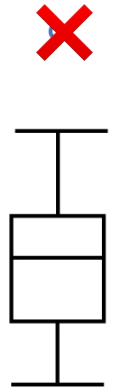
  - Center (e.g. average, median)
  - Spread (e.g. standard deviation, IQR)





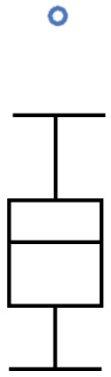
What to do with Outlie

Delete. (“trimming”)



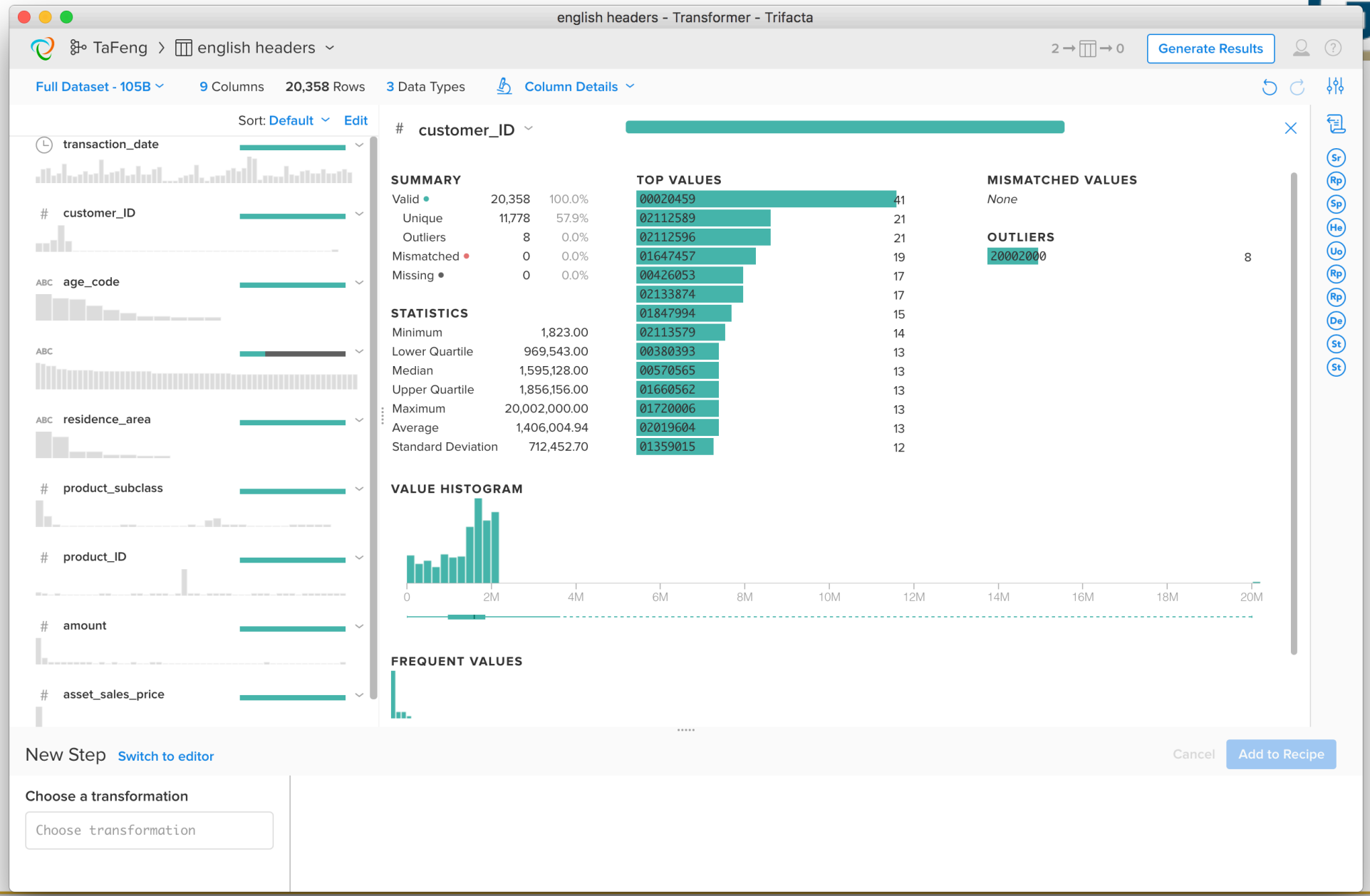
Set to a default

- E.g. the nearest non-outlier (“Winsorizing”)

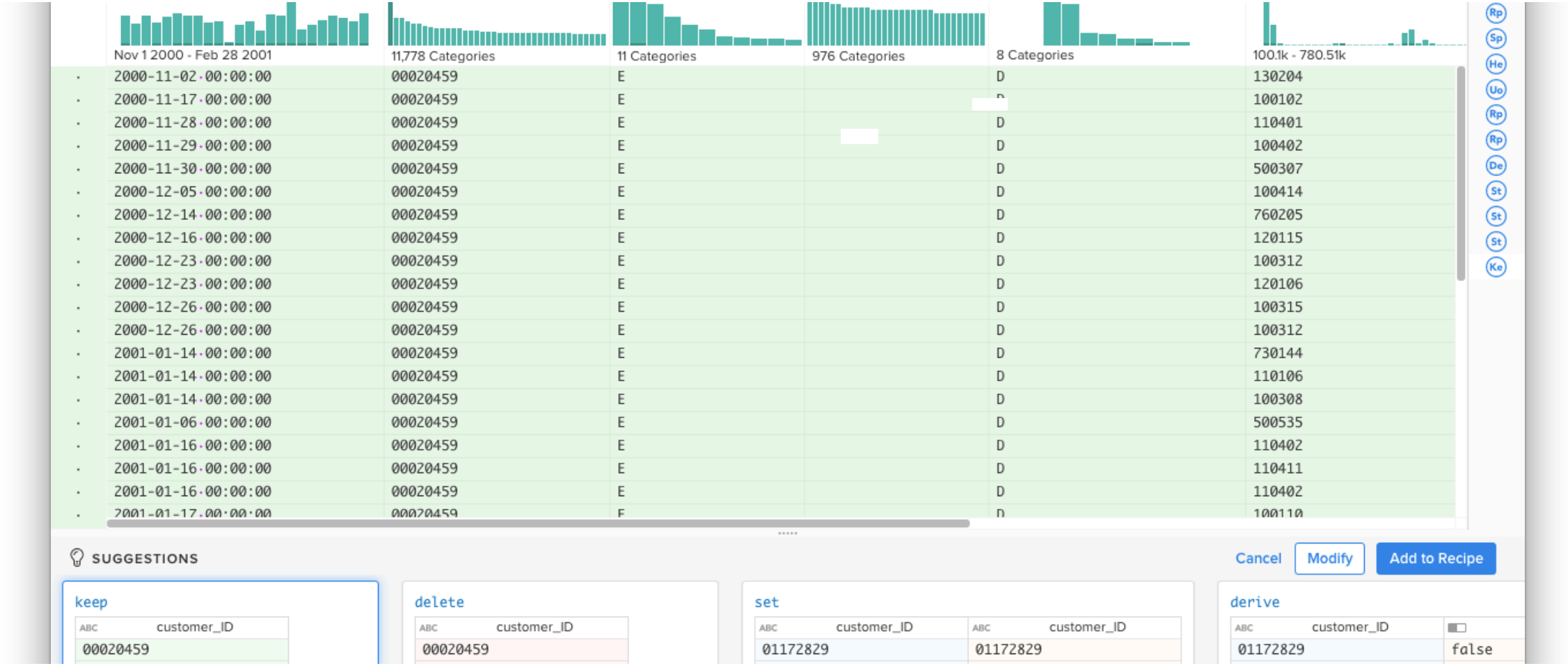


Good Hygiene:

- Leave the original column
- Derive an indicator column to flag presence of outlier
- Derive a clean column for your use


english headers - Transformer - Trifacta
TaFeng
english headers
Generate Results
Full Dataset - 105B
9 Columns
20,358 Rows
3 Data Types
Column Details
Sort: Default
Edit
transaction_date
customer_ID
age_code
residence_area
product_subclass
product_ID
amount
asset_sales_price
customer_ID
SUMMARY
Valid 20,358 100.0%
Unique 11,778 57.9%
Outliers 8 0.0%
Mismatched 0 0.0%
Missing 0 0.0%
STATISTICS
Minimum 1,823.00
Lower Quartile 969,543.00
Median 1,595,128.00
Upper Quartile 1,856,156.00
Maximum 20,002,000.00
Average 1,406,004.94
Standard Deviation 712,452.70
TOP VALUES
00020459 41
02112589 21
02112596 21
01647457 19
00426053 17
02133874 17
01847994 15
02113579 14
00380393 13
00570565 13
01660562 13
01720006 13
02019604 13
01359015 12
MISMATCHED VALUES
None
OUTLIERS
20002000 8
VALUE HISTOGRAM
0 2M 4M 6M 8M 10M 12M 14M 16M 18M 20M
FREQUENT VALUES
New Step
Switch to editor
Cancel
Add to Recipe
Choose a transformation
Choose transformation

Assessing Faithfulness Within Records: Dependencies and Correlations



Functional Dependencies (FDs)

More on this in future!

- Generalization of Keys
- Attribute A *determines* Attribute B
  - `customer_id -> age_code`
  - i.e. age_code = f(customer_id)
    - Function represented as lookup table
- More generally, a set of columns determines another set of columns
  - `{transaction_time,customer_id} -> {age_code, residence_area}`
- Primary Keys are special FDs
  - Right-hand-size is the set of *all* attributes in the relation

Correlations

More on this in future!

Dependence (i.e. lack of independence!) between 2 random variables

Think of the attributes in a relational schema

- An *instance* of that relation was generated from some real-world process
- Each column of that relation is a "random variable" generated by the process

A Functional Dependency is a "deterministic" correlation

Correlations are more general: statistical relationships

- `amount` and `sales_price` are correlated

What to do about bad FDs/Correlations

- Cleaning Noisy FDs: `customer_id -> age_code` (kinda)
  - For a few customer IDs, there are multiple values of age_code
  - Set offending right-hand-side values to all match
  - Set offending left-hand-side values to NULL

- Cleaning Correlations: `height` correlated with `weight`
  - Some rows don't seem to follow the correlation (how do we decide?)
  - Can *impute* a likely value for one side or the other
  - Can set one side or the other to NULL

Careful! More on this in future.

- Don't forget Good Hygiene!
  - Leave the original column
  - Derive new columns (indicators and/or cleaned data)

Faithfulness Questions: A Checklist

- Type-specific Faithfulness checks
  - Are dates and addresses legal/reasonable?
  - Numeric codes legal?: E.g. phone numbers, credit cards, social-security numbers etc.
  - Can you validate network endpoints? Email addresses, IP addresses, social network names
  - Need to deduplicate named entities: misspellings, acronyms (UCB vs Berkeley)
- Checks for data entry problems
  - Frequency outliers for common default entry values (00000, 1234567)
  - Misspellings (compare to dictionaries)
  - Sensor drift – often timeseries-based
  - "Curbstoning" in surveys
- Quantitative dirty data
  - Outliers, FDs, Correlations
- Check distribution of inaccuracies
  - Do inaccuracies seem to affect a large fraction of records?
  - Are they concentrated in a particular subset of records?

Tricky!

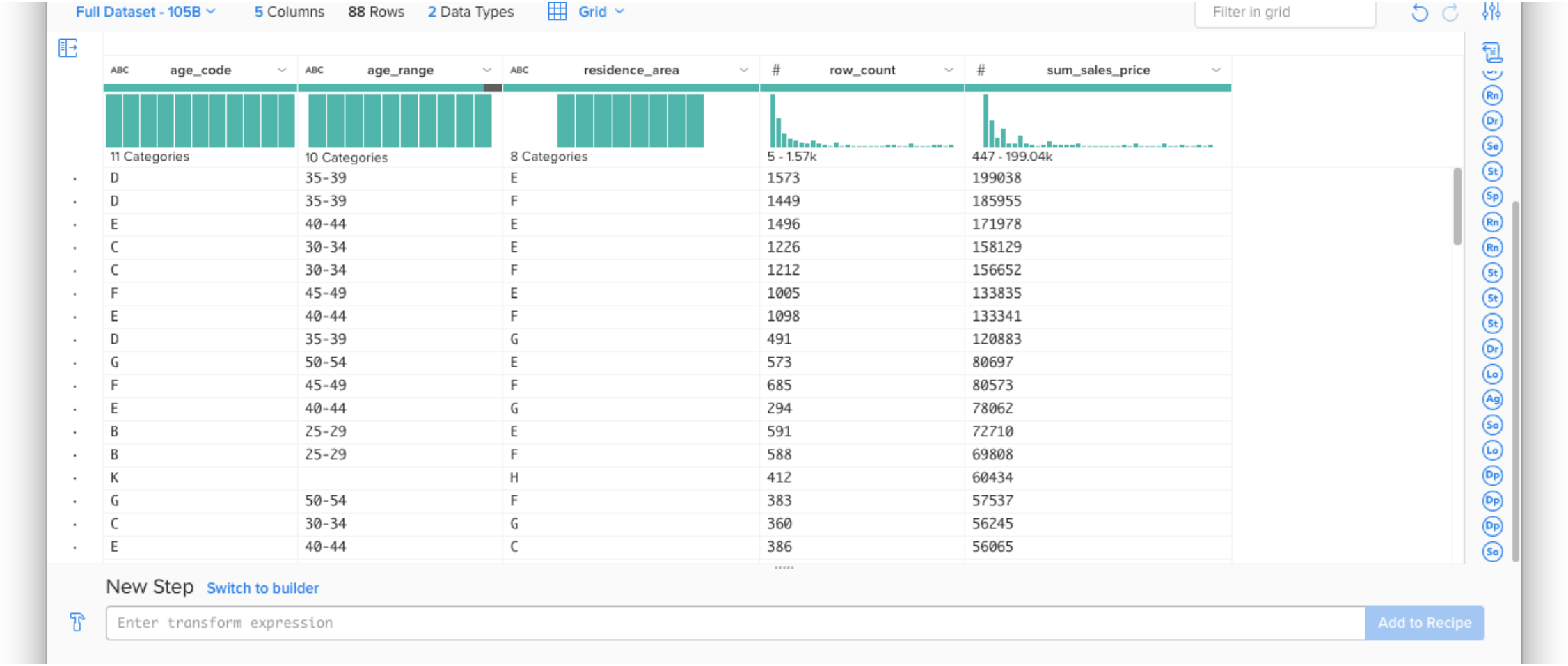
Summing Up: Faithfulness

- Outliers
- Functional Dependencies & Correlations
- Good Data Cleaning Hygiene
  - Don’t overwrite: use indicators and derived columns

Granularity *Transformations*

- We can coarsen the granularity by picking new keys and “rolling up”
  - GroupBy and Aggregation
- GroupBy
  - Choose a new primary key (the group-by columns)
  - Result will have one row per distinct values of this primary key
- Aggregation
  - Summary (rollup) results per group
  - E.g. count(), or aggregation functions on attributes (sum(x), average(x), stdev(x) etc.)

Finalizing Ta Feng



What About Non-Rectangular Data

- Nested Data
- Free Text (i.e. for humans)
- Example that has both: Twitter feed

Unnesting Nested Data Types

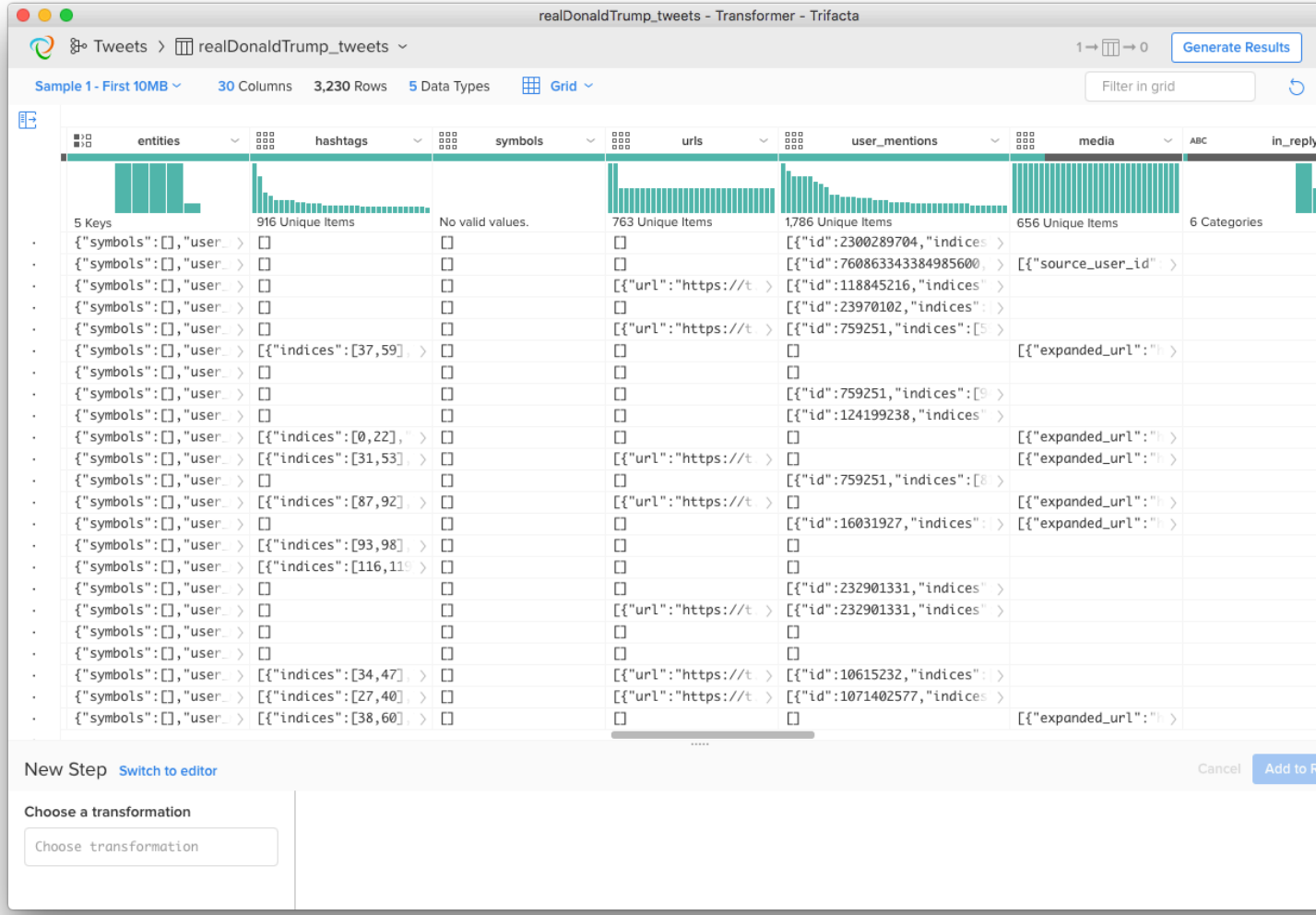
More on this in future lectures!

Maps

- A.k.a. dictionaries, hashes
- A set of key:val pairs

Arrays

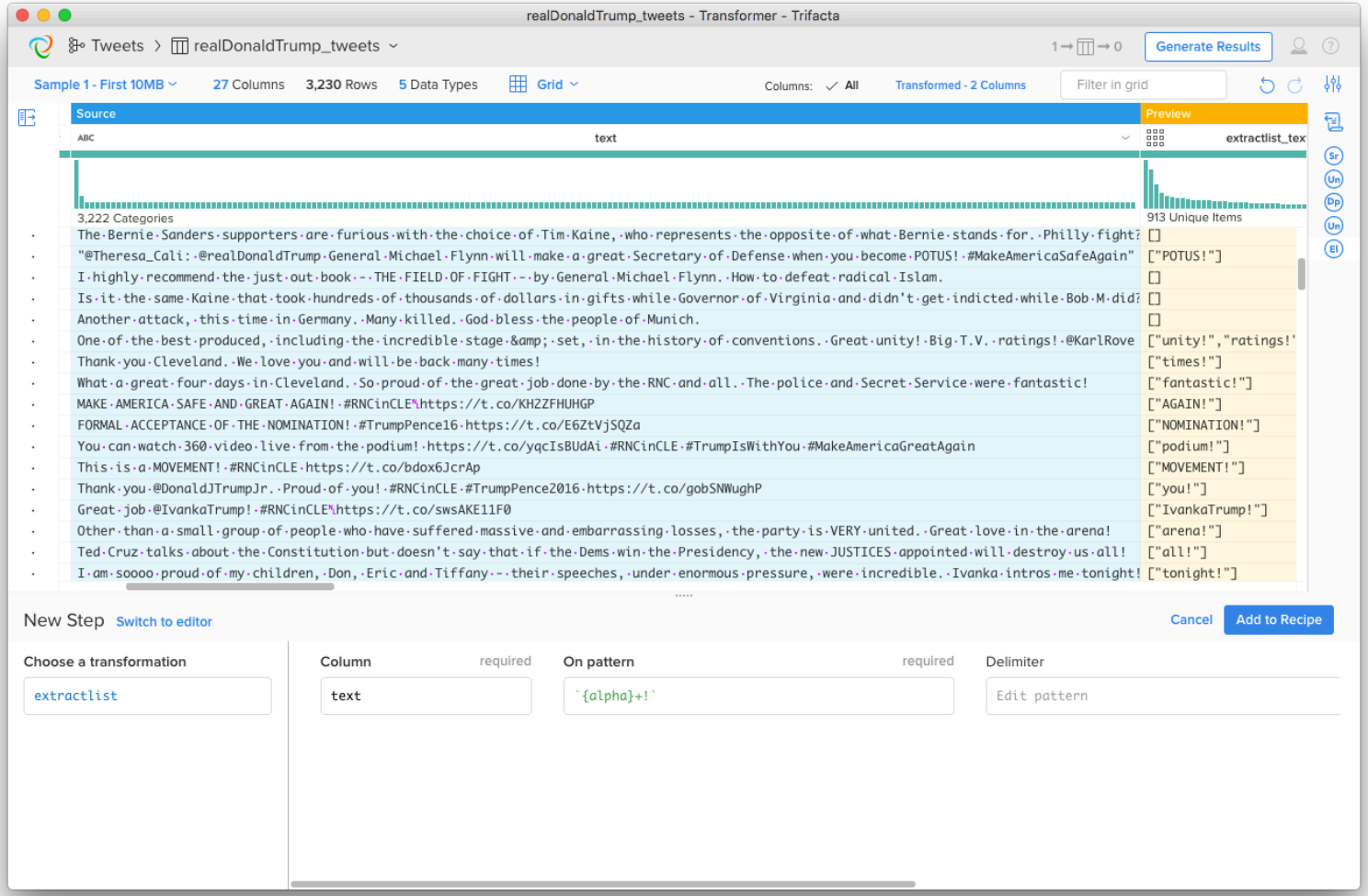
- Or lists



Playing with Text I: String Manipulation

More on this in future lectures!

Regular expression (regex) extraction



Playing with Text II: Natural Language

Natural Language Processing: a slippery slope

- Entity Resolution
  - IBM vs. International Business machines
- Named Entity Recognition
  - Steve Jobs munched on an apple as he announced new jobs at Apple.
- Sentiment
  - My windows box crashed yet again.
- Etc. Etc.
- Hot topic today: Dialog systems:
  - Alexa, where did I leave my keys?
    - I don't know. It's a pity you don't have a tracker on your key fob.
  - A what?
    - A bluetooth tracker – there are variety of them for sale
  - Stop trying to get me to buy stuff!
    - Sorry about that; I know you've been trying to save this month…
    - …but if you had one I could locate your keys.

Playing with Text II: Natural Language

- Some simple things
  - Term frequencies
  - Simple Sentiment analysis

Quick Note: Assessing Temporality

- Often two kinds of time in data
  - Time of data entry
  - Time of a recorded phenomenon being “true”
    - E.g. A physical time of an event happening
    - E.g. An “effective” time, e.g. date that a subscription will start
- Often more
- Time is tricky!
  - Periodicities (recurring patterns in Days of the week, or Months of the year)
  - Non-uniform hierarchy of units (# days in a month, # of days in a year, etc.)
  - Time zones are complex: especially daylight savings (summer) time
  - Clocks can be skewed
  - Relativity: true perception of event may vary (yep!)
- Assess timestamps in data carefully!!

Quick Note: Assessing Scope

- Do you have all the data you need
  - Missing columns
    - Join in external data
  - Missing rows/values?
    - Look for sequential patterns with breaks
    - Is this a good sample?
      - Granularity
      - Extent
    - How to *impute* reasonable stand-in values

More on this in future lectures!

- Can be quite application specific/subjective!

Classification of postal codes [edit]

Below is the list of postal codes in Taiwan.

1 – Taipei [edit]

Code	Division name	Chinese
	Taipei City	
100	Zhongzheng District	中正區
103	Datong District	大同區
104	Zhongshan District	中山區
105	Songshan District	松山區
106	Da'an District	大安區
108	Wanhua District	萬華區
110	Xinyi District	信義區
111	Shilin District	士林區
112	Beitou District	北投區
114	Neihu District	內湖區
115	Nangang District	南港區
116	Wenshan District	文山區

2 – Keelung, New Taipei, Matsu, Yilan [edit]

Code	Division name	Chinese
	Keelung City	
200	Ren'ai District	仁愛區
201	Xinyi District	信義區
202	Zhongzheng District	中正區
203	Zhongshan District	中山區
204	Anle District	安樂區
205	Nuannuan District	暖暖區

Looking Back: Data Transformations

Modifying structure
- Splitting rows and columns (“splitrows”, “split”)
- Unnesting (“flatten”)

Hiding Things
- Rows (“keep”, “delete” a.k.a. “select”)
- Columns (“drop”, “aggregate”)

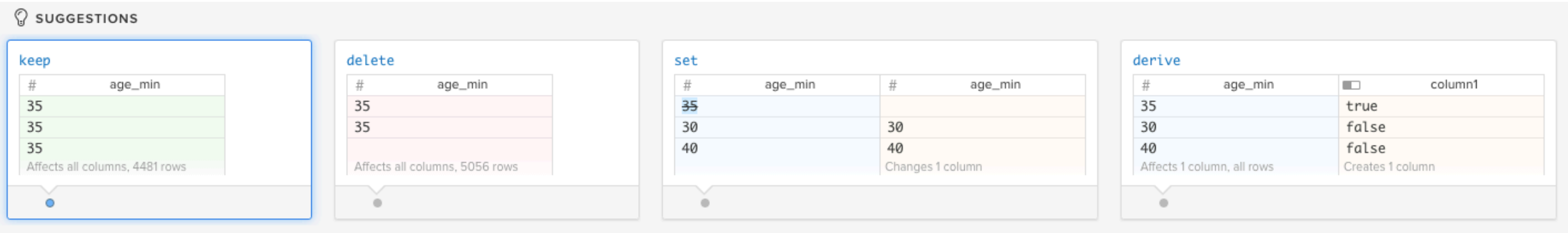
Adding Things
- “derive” a.k.a. “map”, “apply”

Changing Things
- “replace”/”set”

Text manipulation
- “extract”, “replace”, “set”

GroupBy/Aggregate arithmetic
- “aggregate”, a.k.a. “group by”, “reduce”

Multi-table Operations
- “join”, “lookup”
- “union”



Looking Back: Bigger Picture

*What **decisions** did we make along the way?*

- Data that got hidden
  - Choice of columns to drop
  - Values we cleaned: indicators, cleaned columns
  - Filtering of rows (vs. indicators)
- Data that got added
  - Other derived columns: calculations, splits, extractions
  - Joins, Unions
- Changes in granularity:
  - Coarser: grouping keys and aggregates
  - Finer: unnesting

*How might the data **wrangling** influence the **analyses** we can do?*

- Sometimes we won't figure this out until analysis begins
- And we loop back to wrangling!

Impediments to Collaboration

- Diversity of tools and PLs makes it hard to share
- Finding a script or computed result is harder than just writing the program from scratch!
  - Q: How could we fix this?
- View that much of the analysis work is “throw away”

Data Sources at Web Companies

- Examples from Facebook
  - Application databases
  - Web server logs
  - Event logs
  - API server logs
  - Ad server logs
  - Search server logs
  - Advertisement landing page content
  - Wikipedia
  - Images and video

Tabular Data

- What is a table?
  - A **table** is a collection of **rows** and **columns**
  - Each row has an **index**
  - Each column has a **name**
  - A **cell** is specified by an (index, name) pair
  - A cell may or may not have a **value**

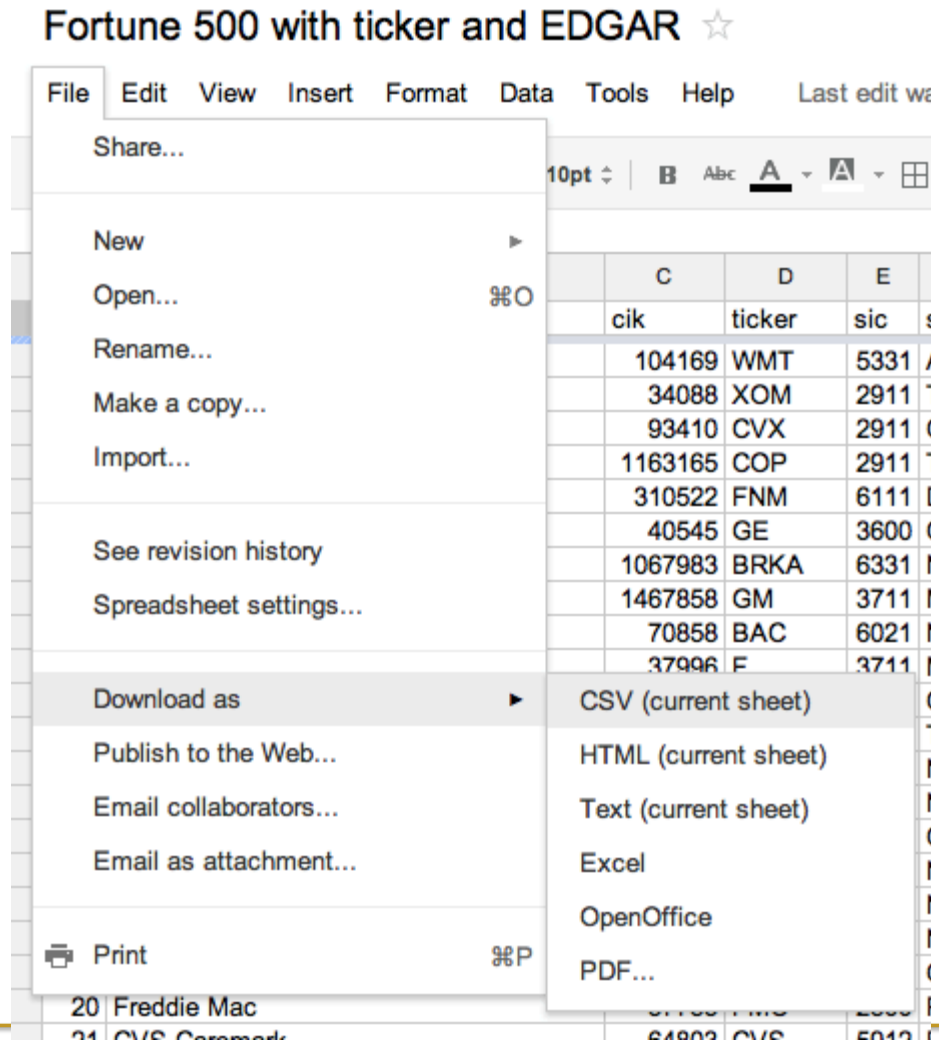
Tabular Data

- Fortune 500

	A	B	C	D	E	F	G	H	I
1	rank	company	cik	ticker	sic	state_location	state_of_incorporation	revenues	profits
2	1	Wal-Mart Stores	104169	WMT	5331	AR	DE	421849	16389
3	2	Exxon Mobil	34088	XOM	2911	TX	NJ	354674	30460
4	3	Chevron	93410	CVX	2911	CA	DE	196337	19024
5	4	ConocoPhillips	1163165	COP	2911	TX	DE	184966	11358
6	5	Fannie Mae	310522	FNM	6111	DC	DC	153825	-14014
7	6	General Electric	40545	GE	3600	CT	NY	151628	11644
8	7	Berkshire Hathaway	1067983	BRKA	6331	NE	DE	136185	12967
9	8	General Motors	1467858	GM	3711	MI	MI	135592	6172
10	9	Bank of America Corp.	70858	BAC	6021	NC	DE	134194	-2238
11	10	Ford Motor	37996	F	3711	MI	DE	128954	6561
12	11	Hewlett-Packard	47217	HPQ	3570	CA	DE	126033	8761
13	12	AT&T	732717	T	4813	TX	DE	124629	19864
14	13	J.P. Morgan Chase & Co.	19617	JPM	6021	NY	DE	115475	17370
15	14	Citigroup	831001	C	6021	NY	DE	111055	10602
16	15	McKesson	927653	MCK	5122	CA	DE	108702	1263
17	16	Verizon Communications	732712	VZ	4813	NY	DE	106565	2549
18	17	American International Group	5272	AIG	6331	NY	DE	104417	7786
19	18	International Business Machines	51143	IBM	3570	NY	NY	99870	14833
20	19	Cardinal Health	721371	CAH	5122	OH	OH	98601.9	642.2
21	20	Freddie Mac	37785	FMC	2800	PA	DE	98368	-14025

Tabular Data

- Fortune 500



Tabular Data (csv)

- Fortune 500

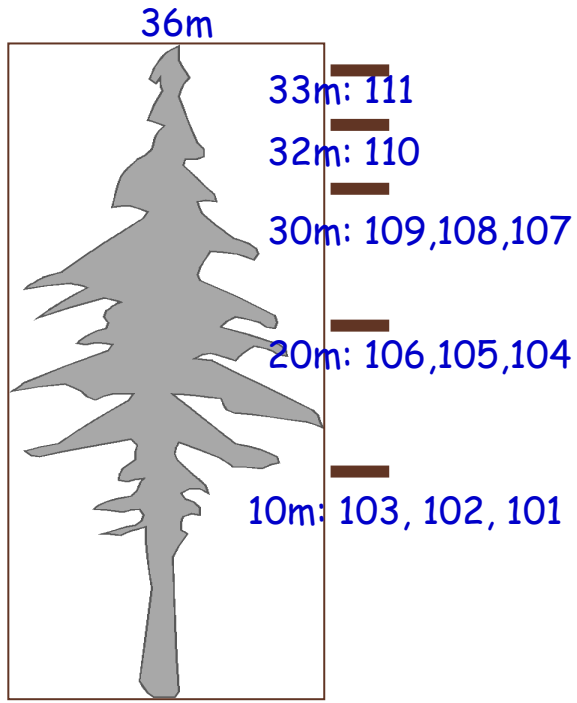
Fortune 500 with ticker and EDGAR – Plus Ticker and EDGAR.txt

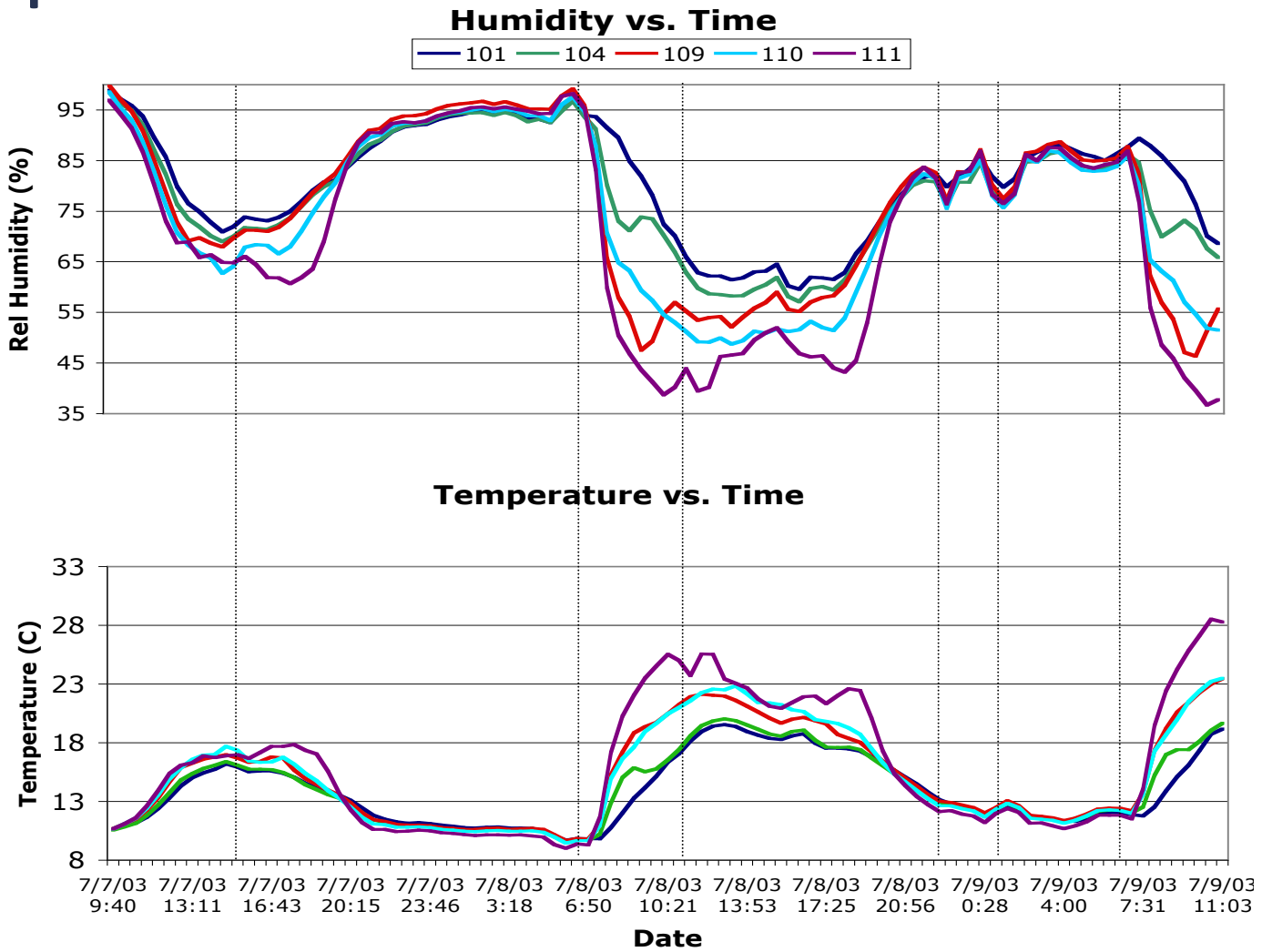
```
rank,company,cik,ticker,sic,state_location,state_of_incorporation,revenues,profits
1,Wal-Mart Stores,104169,WMT,5331,AR,DE,421849,16389
2,Exxon Mobil,34088,XOM,2911,TX,NJ,354674,30460
3,Chevron,93410,CVX,2911,CA,DE,196337,19024
4,ConocoPhillips,1163165,COP,2911,TX,DE,184966,11358
5,Fannie Mae,310522,FNM,6111,DC,DC,153825,-14014
6,General Electric,40545,GE,3600,CT,NY,151628,11644
7,Berkshire Hathaway,1067983,BRKA,6331,NE,DE,136185,12967
8,General Motors,1467858,GM,3711,MI,MI,135592,6172
9,Bank of America Corp.,70858,BAC,6021,NC,DE,134194,-2238
10,Ford Motor,37996,F,3711,MI,DE,128954,6561
11,Hewlett-Packard,47217,HPQ,3570,CA,DE,126033,8761
12,AT&T,732717,T,4813,TX,DE,124629,19864
13,J.P. Morgan Chase & Co.,19617,JPM,6021,NY,DE,115475,17370
14,Citigroup,831001,C,6021,NY,DE,111055,10602
15,McKesson,927653,MCK,5122,CA,DE,108702,1263
16,Verizon Communications,732712,VZ,4813,NY,DE,106565,2549
17,American International Group,5272,AIG,6331,NY,DE,104417,7786
18,International Business Machines,51143,IBM,3570,NY,NY,99870,14833
19,Cardinal Health,721371,CAH,5122,OH,OH,98601.9,642.2
20,Freddie Mac,37785,FMC,2800,PA,DE,98368,-14025
21,CVS Caremark,64803,CVS,5912,RI,DE,96413,3427
22,UnitedHealth Group,731766,UNH,6324,MN,MN,94155,4634
23,Wells Fargo,72971,WFC,6021,CA,DE,93249,12362
24,Valero Energy,1035002,VLO,2911,TX,DE,86034,324
25,Kroger,56873,KR,5411,OH,OH,82189.4,1116.3
26,Procter & Gamble,80424,PG,2840,OH,OH,79689,12736
27,AmerisourceBergen,1140859,ABC,5122,PA,DE,77954,636.7
28,Costco Wholesale,909832,COST,5331,WA,WA,77946,1303
29,Marathon Oil,101778,MRO,2911,TX,DE,68413,2568
30,Home Depot,354950,HD,5211,GA,DE,67997,3338
```

Protein Data Bank

```
HEADER    APOPTOSIS                               05-OCT-10   3IZA
TITLE     STRUCTURE OF AN APOPTOSOME-PROCASPASE-9 CARD COMPLEX
COMPND    MOL_ID: 1;
COMPND   2 MOLECULE: APOPTOTIC PROTEASE-ACTIVATING FACTOR 1;
COMPND   3 CHAIN: A, B, C, D, E, F, G;
COMPND   4 SYNONYM: APAF-1;
COMPND   5 ENGINEERED: YES
SOURCE    MOL_ID: 1;
SOURCE   2 ORGANISM_SCIENTIFIC: HOMO SAPIENS;
SOURCE   3 ORGANISM_COMMON: HUMAN;
SOURCE   4 ORGANISM_TAXID: 9606;
SOURCE   5 GENE: APAF1, KIAA0413;
SOURCE   6 EXPRESSION_SYSTEM: SPODOPTERA FRUGIPERDA;
SOURCE   7 EXPRESSION_SYSTEM_TAXID: 7108;
SOURCE   8 EXPRESSION_SYSTEM_STRAIN: SF21;
SOURCE   9 EXPRESSION_SYSTEM_VECTOR_TYPE: INSECT VIRUS;
SOURCE  10 EXPRESSION_SYSTEM_PLASMID: PFASTBAC1
KEYWDS    APOPTOSOME, APAF-1, PROCASPASE-9 CARD, APOPTOSIS
EXPDTA    ELECTRON MICROSCOPY
AUTHOR    S.YUAN,X.YU,M.TOPF,S.J.LUDTKE,X.WANG,C.W.AKEY
REVDAT   1   03-NOV-10 3IZA    0
SPRSDE     03-NOV-10 3IZA      3IYT
JRNL        AUTH   S.YUAN,X.YU,M.TOPF,S.J.LUDTKE,X.WANG,C.W.AKEY
JRNL        TITL   STRUCTURE OF AN APOPTOSOME-PROCASPASE-9 CARD COMPLEX
JRNL        REF    STRUCTURE                     V.  18   571 2010
```

Internet of Things: Example measurements





Tabular Data from Sensors

Challenges

- May be many missing fields (a particular sensor may not produce all types of output).
- Device may go offline for a while.
- Device may be damaged (permanently or intermittently).
- Timestamps usually critical but may not be accurate.
- Other meta-data (location, device ID) may have errors.

Log Files – Example Apache Web Log

Processes, usually daemons, create logs
e.g., `httpd, mysqld, syslogd`

- `66.249.65.107 - - [08/Oct/2007:04:54:20 -0400] "GET /support.html HTTP/1.1" 200 11179 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)"`

- `111.111.111.111 - - [08/Oct/2007:11:17:55 -0400] "GET / HTTP/1.1" 200 10801 "http://www.google.com/search?q=log+analyzer&ie=utf-8&oe=utf-8&aq=t&rls=org.mozilla:en-US:official&client=firefox-a" "Mozilla/5.0 (Windows; U; Windows NT 5.2; en-US; rv:1.8.1.7) Gecko/20070914 Firefox/2.0.0.7"`

- `111.111.111.111 - - [08/Oct/2007:11:17:55 -0400] "GET /style.css HTTP/1.1" 200 3225 ""http://www.loganalyzer.net/" "Mozilla/5.0 (Windows; U; Windows NT 5.2; en-US; rv:1.8.1.7) Gecko/20070914 Firefox/2.0.0.7"`

Syslog – A Standard for System Messages

- Developed by Eric Allman (at Berkeley) as part of the Sendmail project
- Standardized by the IETF in RFC 3164 and RFC 5424
- Listens on port 514 using UDP
- Puts data in /var/log/messages by default

- Enables rich analysis:

splunk>

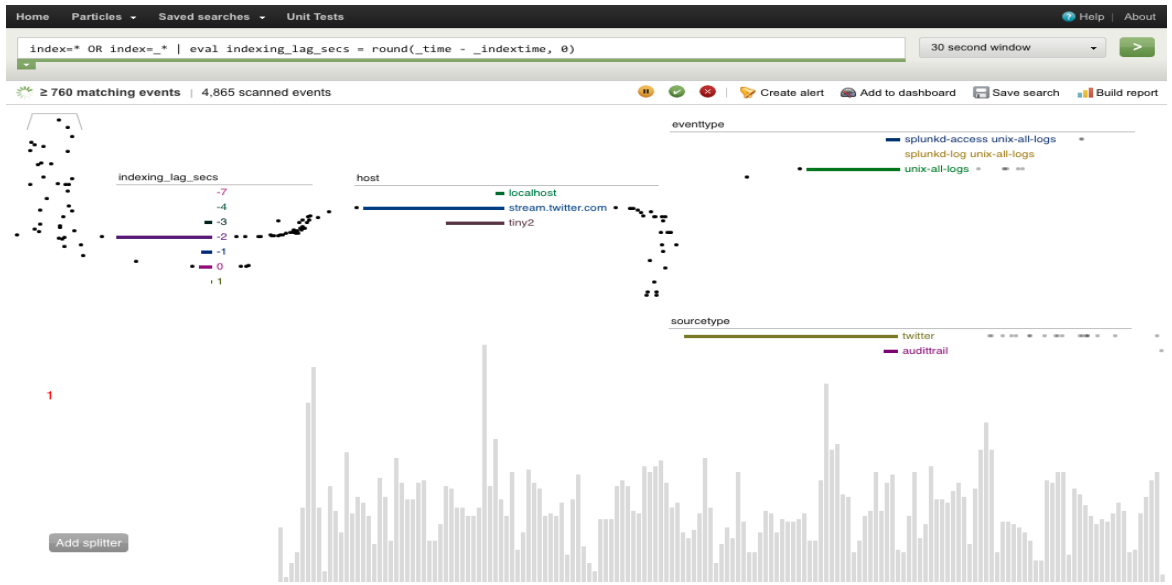
Syslog

dhcp-47-129:DataScienceF14> syslog -w 10

Feb 3 15:18:11 dhcp-47-129 Evernote[1140] <Warning>: -[EDAMAccounting read:]: unexpected field ID 23 with type 8. Skipping.
Feb 3 15:18:11 dhcp-47-129 Evernote[1140] <Warning>: -[EDAMUser read:]: unexpected field ID 17 with type 12. Skipping.
Feb 3 15:18:11 dhcp-47-129 Evernote[1140] <Warning>: -[EDAMAuthenticationResult read:]: unexpected field ID 6 with type 11. Skipping.
Feb 3 15:18:11 dhcp-47-129 Evernote[1140] <Warning>: -[EDAMAuthenticationResult read:]: unexpected field ID 7 with type 11. Skipping.
Feb 3 15:18:11 dhcp-47-129 Evernote[1140] <Warning>: -[EDAMAccounting read:]: unexpected field ID 19 with type 8. Skipping.
Feb 3 15:18:11 dhcp-47-129 Evernote[1140] <Warning>: -[EDAMAccounting read:]: unexpected field ID 23 with type 8. Skipping.
Feb 3 15:18:11 dhcp-47-129 Evernote[1140] <Warning>: -[EDAMUser read:]: unexpected field ID 17 with type 12. Skipping.
Feb 3 15:18:11 dhcp-47-129 Evernote[1140] <Warning>: -[EDAMSyncState read:]: unexpected field ID 5 with type 10. Skipping.
Feb 3 15:18:49 dhcp-47-129 com.apple.mtmd[47] <Notice>: low priority thinning needed for volume Macintosh HD (/) with 18.9 <= 20.0 pct free space

“Splunking”

- Grab data from many machines
- Index it
- Check for unusual events:
  - Disk problems
  - Network congestion
  - Security attacks
- Monitor Resources
  - Network
  - Memory usage
  - Disk use, latency
  - Threads
- Dashboard for cloud administration.



Outline for this Evening

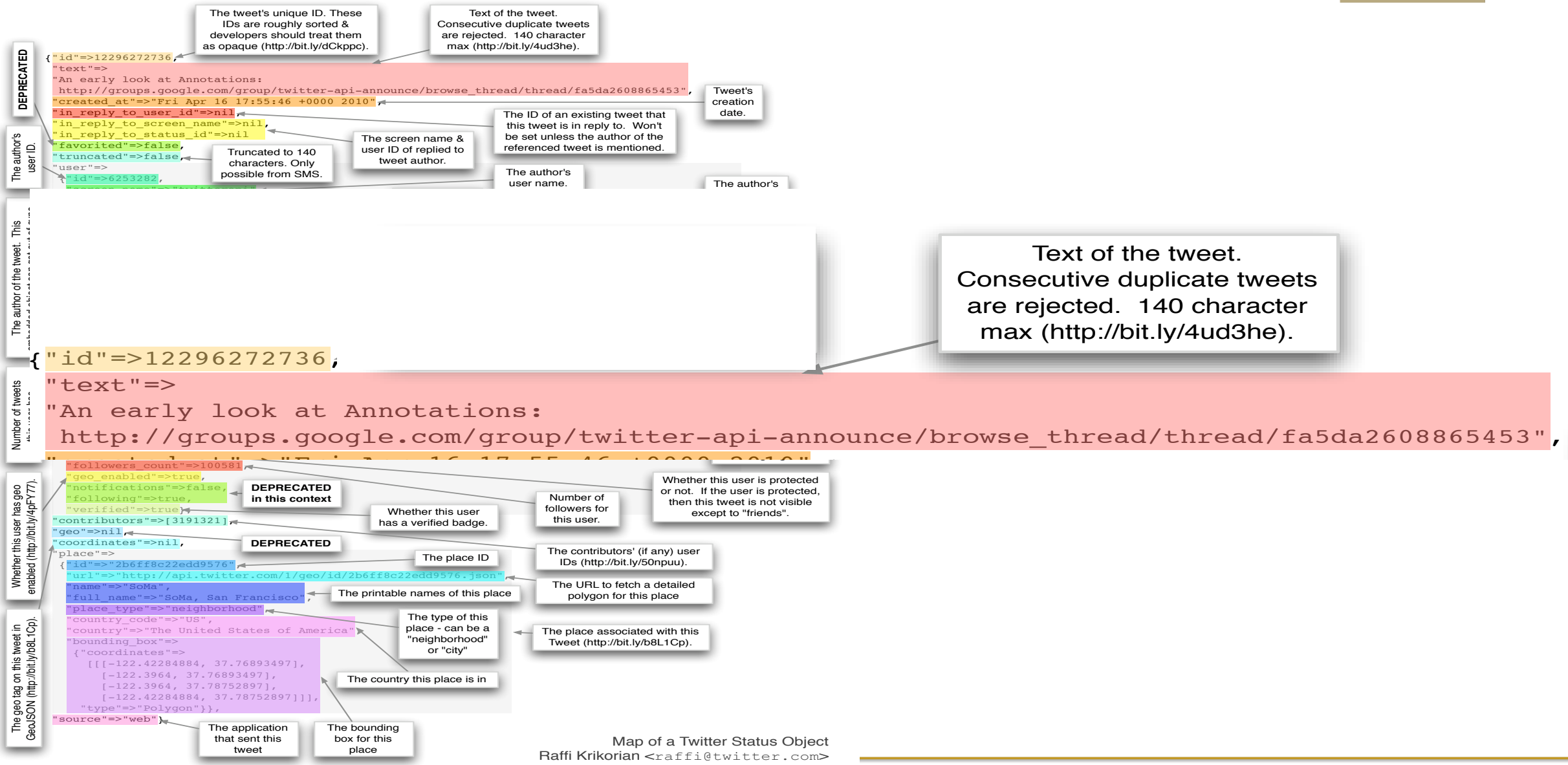
- Some Ideas from Kandel et al. Paper
- Data Types and Sources – “Document” data
- Data Preparation and Manipulation
- Files

Some Questions

1) How Many Characters are there in a Tweet?

2) How Many Bytes are there in a Tweet (msg)?

Tweet JSON Format



Map of a Twitter Status Object
Raffi Krikorian <raffi@twitter.com>
18 April 2010

Tweet JSON Format

So how do we process the JSON from Twitter?

Stay tuned, but first some concepts from the XML world…

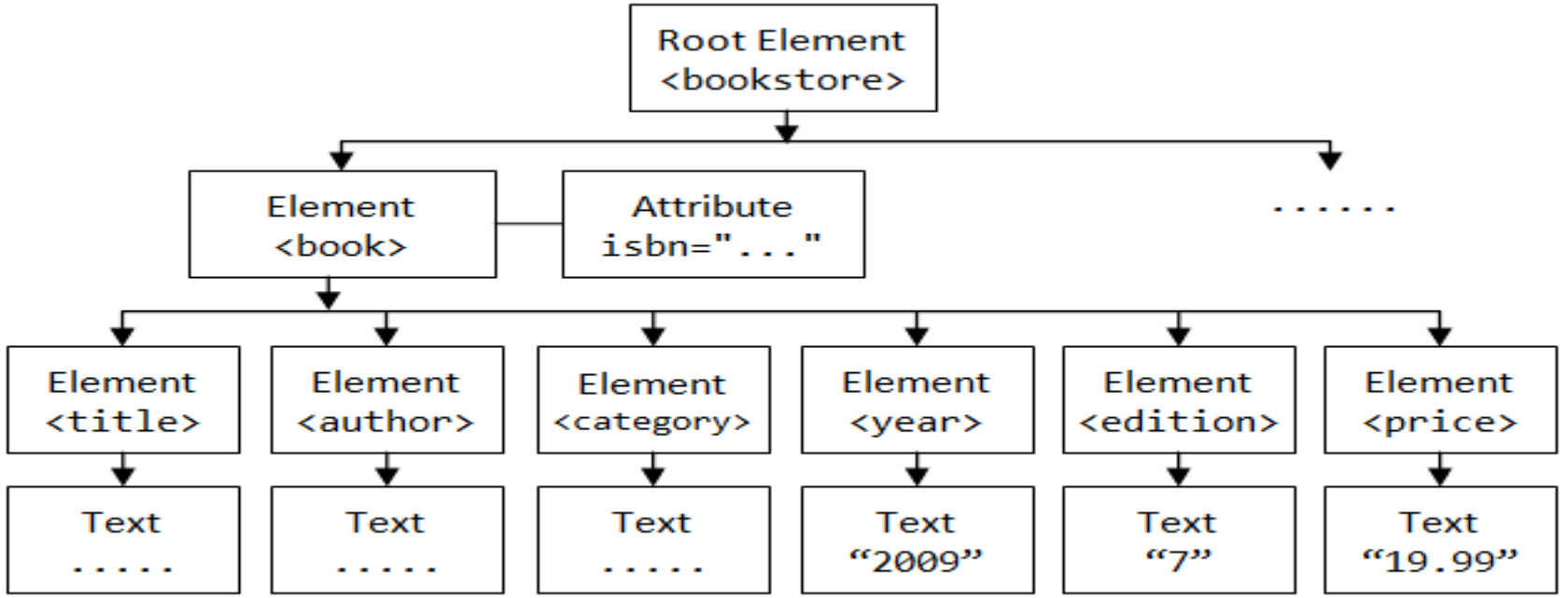
XML, DOM and SAX

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- bookstore.xml -->
<bookstore>
  <book ISBN="0123456001">
    <title>Java For Dummies</title>
    <author>Tan Ah Teck</author>
    <category>Programming</category>
    <year>2009</year>
    <edition>7</edition>
    <price>19.99</price>
  </book>
```

XML, DOM and SAX

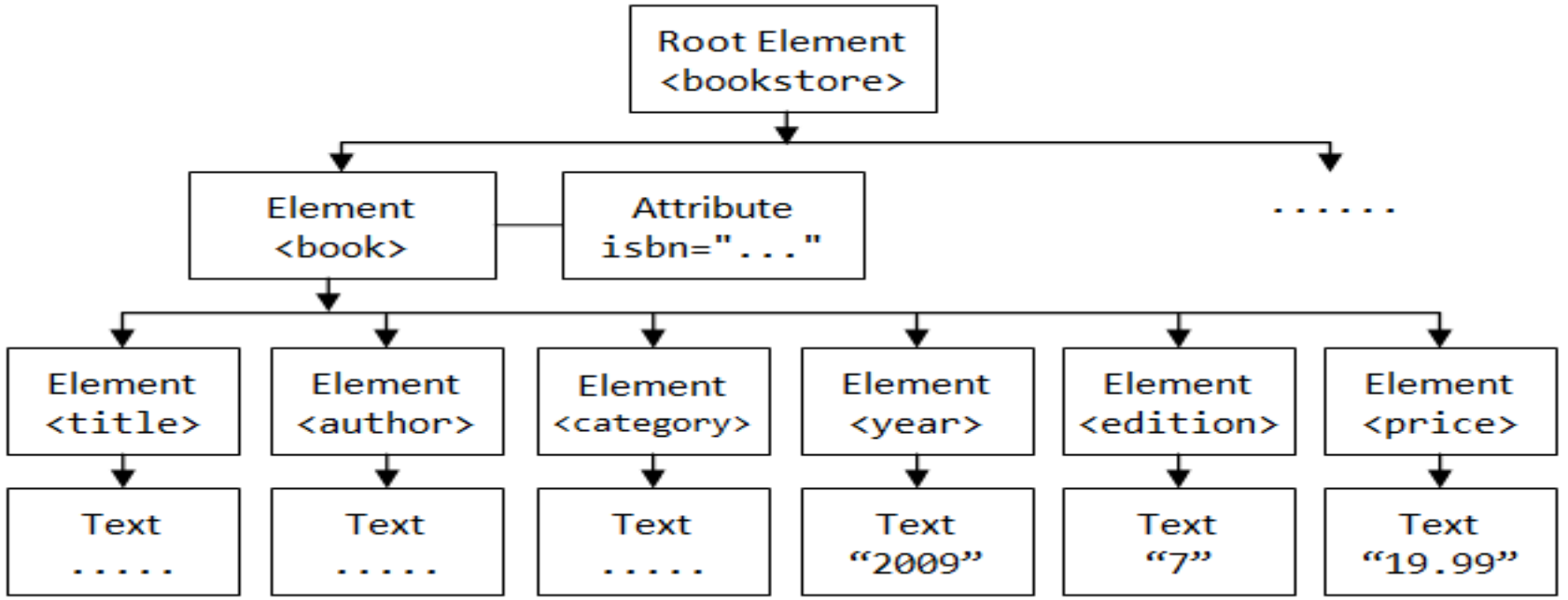
XML is a text format that encodes DOM (Document-Object Models) which is a data structure e.g. for Web pages.

The DOM is tree-structured:



XML, DOM and SAX

- The DOM is an easy object to work with: all the data in the object is accessible by links.
- The problem is that I might not care about most of the data, and I might not be able to fit the DOM for a large object in RAM.



SAX

SAX (Simple API for XML) is an alternative “lightweight” way to process XML.

A SAX parser generates a stream of events as it parses the XML file. The programmer registers handlers for each one.

It allows a programmer to handle only parts of the data structure.

SAX

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- bookstore.xml -->
<bookstore>
  <book ISBN="0123456001">
    <title>Java For Dummies</title>
    <author>Tan Ah Teck</author>
    <category>Programming</category>
    <year>2009</year>
    <edition>7</edition>
    <price>19.99</price>
  </book>
```

→ Document Header

→ Comment

→ Start-element “bookstore”

→ Start-element “book”

→ Start-element “title”

End-element “title”

→ End-element “book”

What about JSON?

Most JSON parsers construct the “DOM” directly.

But there are a few SAX-style parsers:

- Jackson
- JSON-simple

What about HTML?

- Common Crawl, about 5 billion web pages, between 0.2-0.5% of Google's web crawl.
- 60 TB, hosted on Amazon S3, also available for download.
- Includes link data, page rank.
- In ARC (Internet Archive) File format.

- So there's plenty of data, and there are many crawlers for targeted exploration…
  - HTTrack, …

HTML Tag Soup


```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head><!-- types/widgets/pages/common/page.tmpl home/index_v3.html generated by index_v3 on Wed 29 Feb 2012 11:04:41 PM PST -->
<title>San Francisco Bay Area &mdash; News, Sports, Business, Entertainment, Classifieds: SFGate</title>
<meta http-equiv="content-type" content="text/html; charset=iso-8859-1" />
<meta name="description" content="Find local news &amp;amp; information, updated weather, traffic, classifieds, sports scores, real estate, jobs, cars, food &amp;amp; wine, travel, entertainment, events and more on SFGate.com. Connect to the Bay Area community." />
<meta name="keywords" content="San Francisco, San Francisco Bay Area, news, local events, breaking news, world news, San Francisco Chronicle, SFGate" />
<meta property="fb:page_id" content="105702905593" />
<meta property="fb:admins" content="653226748,658759748" />
<!-- /widgets/sitewide/css/all/inc.html widgets/pages/common/post_write_mtime/css_inc.tmpl -->
<!-- generated by sitewidecss on Thu 16 Feb 2012 10:41:53 AM PST -->
<link rel="stylesheet" type="text/css" title="SFGate" media="all" href="http://imgs.sfgate.com/css1329417713/sitewide/css/sitewide.css" />
<!-- sitewide/css/all/inc.html end css_inc.tmpl -->
```

HTML Tools - Parsing

- "Beautiful Soup" http://www.crummy.com/software/BeautifulSoup/
  a Python API for handling real HTML. DOM or SAX interfaces.

- "TagSoup"
  http://ccil.org/~cowan/XML/tagsoup/
  provides a Sax interface, i.e. a streaming parse, to Java applications. Can transform to a format you want using XSLT.

- Taggle, part of the Arabica toolset
  http://www.jezuk.co.uk/cgi-bin/view/arabica/code
  is a version of TagSoup written in C++. You probably want to use this if you have a lot of data.

Web Services

Most large web sites today actively discourage screen-scraping to get their content, and provide Web Service APIs instead.

This is the “right” way to get data from online sources.

Web Services

W3C definition:

a "Web service" as "a software system designed to support interoperable machine-to-machine interaction over a network".

Two kinds:

- XML-based RPC-style messages: WSDL and SOAP
- REST-style stateless interactions, URLs encode state

Can run over different transports, but usually HTTP

Examples

Twitter: REST API and streaming API with JSON content. Provides sampling, searching and filtering capabilities.

Amazon: has a “product advertising API” in XML with a WSDL spec. Includes product search, reviews etc.

Livejournal: RSS/Atom + custom XML/RPC. Search by keyword, topic, follow friend links.

Netflix: Javascript, Atom and REST interfaces.

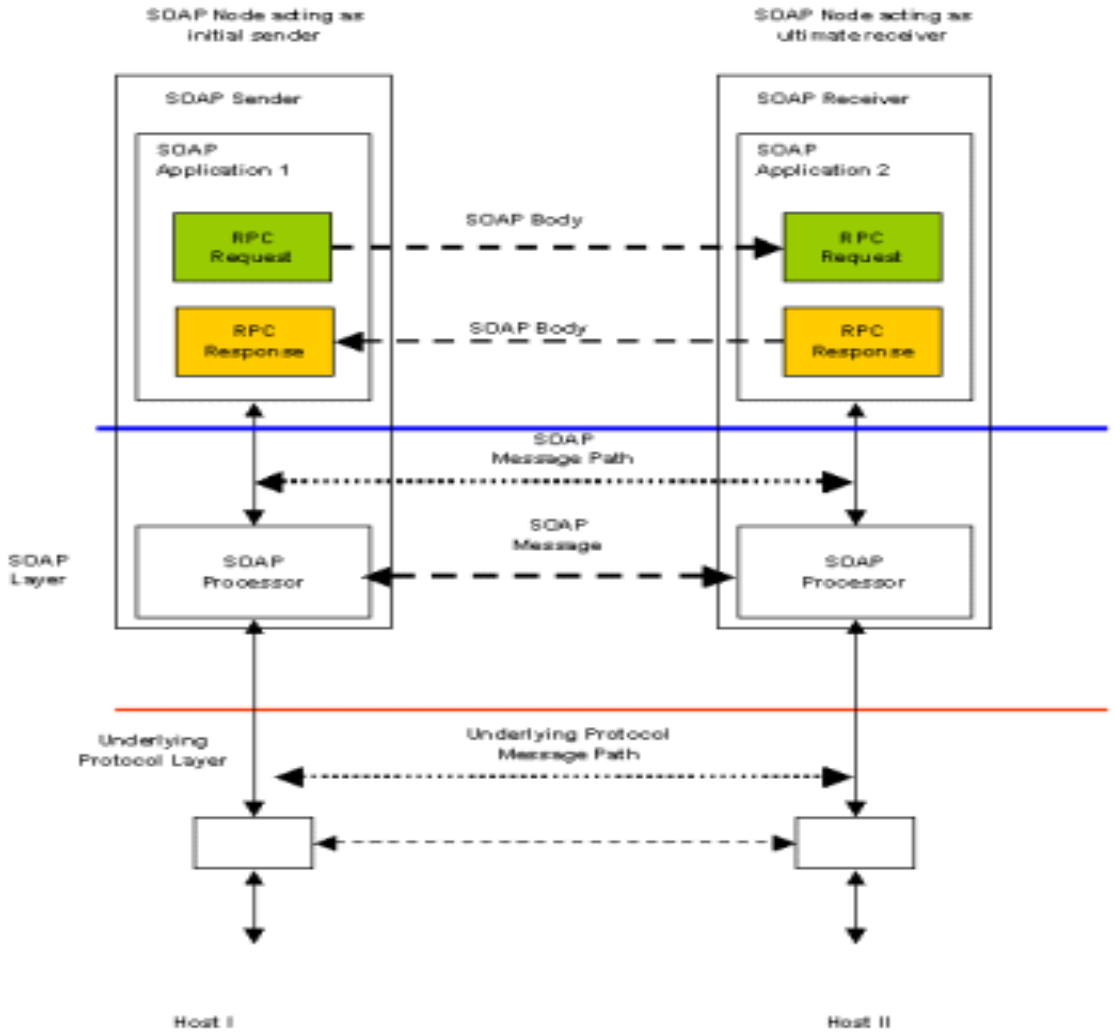
Ebay: Many APIs for searching, buying and posting. WSDL descriptions, client code in Java and .NET

Flickr: Comprehensive API set, free for non-commercial use. REST, XML-RPC, SOAP, with client code in many languages.

vBulletin: REST interface, most actions supported

Web Services

XML-RPC, requires a request-response cycle. Often longer “conversations.”



WSDL and SOAP

- Conceptually a Remote-Procedure-Call system, like CORBA, Java RMI etc.
- But RPC often has to pass through multiple layers of firewalls, causing many problems as security increased in the 1990s.
- Web services typically use HTTP as a transport, which runs almost anywhere, provides security and simple GET-POST methods. So HTTP-based RPC was a natural choice.
- SOAP uses XML messages, is human-readable, easy to process from any programming language, and relatively robust to version slippage.

SOAP is

SOAP covers the following four main areas:

- **A message format** for one-way communication describing how a message can be packed into an XML document.
- A **description** of how a SOAP message should be transported using HTTP (for Web-based interaction) or SMTP (for e-mail-based interaction).
- A **set of rules** that must be followed when processing a SOAP message and a simple classification of the entities involved in processing a SOAP message.
- A **set of conventions** on how to turn an RPC call into a SOAP message and back.

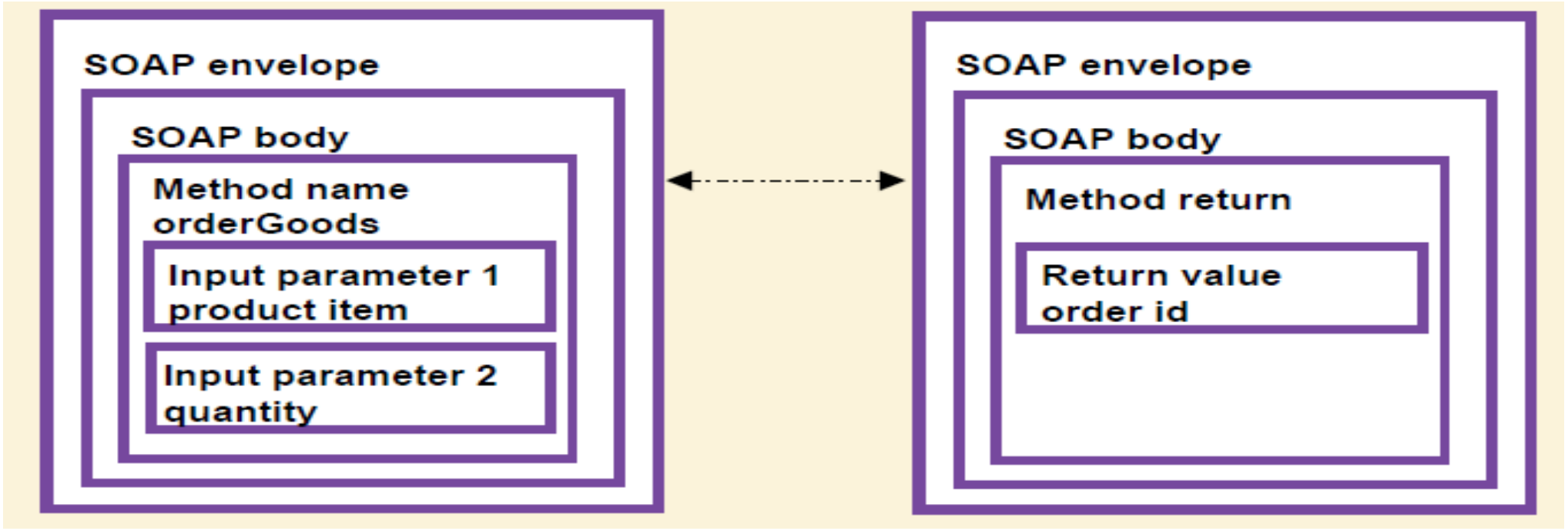
Soap Message

Typically an XML element containing header and body elements

```
<SOAP:Envelope xmlns:SOAP=
      "http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP:Header>
      <!– content of header goes here –>
   </SOAP:Header>
   <SOAP:Body>
      <!– content of body goes here –>
   </SOAP:Body>
</SOAP:Envelope>
```

SOAP RPC

SOAP RPC messages typically encode arguments that are presented to the calling program as parameters and return values:



Soap RPC

```
POST /travelservice
SOAPAction: "http://www.acme-travel.com/flightinfo"
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn

<SOAP:Envelope xmlns:SOAP=
    "http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP:Body>
    <m:GetFlightInfo
      xmlns:m="http://www.acme-travel.com/flightinfo"
      SOAP:encodingStyle=
        "http://schemas.xmlsoap.org/soap/encoding/"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema"
      xmlns:xsi=
        "http://www.w3.org/2001/XMLSchema-instance">
      <airlineName xsi:type="xsd:string">UL
      </airlineName>
        <flightNumber xsi:type="xsd:int">506
        </flightNumber>
    </m:GetFlightInfo>
  </SOAP:Body>
</SOAP:Envelope>
```

Soap Response

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn

<SOAP:Envelope xmlns:SOAP=
    "http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP:Body>
    <m:GetFlightInfoResponse
        xmlns:m="http://www.acme-travel.com/flightinfo"
        SOAP:encodingStyle=
          "http://schemas.xmlsoap.org/soap/encoding/"
        xmlns:xsd="http://www.w3.org/2001/XMLSchema"
        xmlns:xsi=
          "http://www.w3.org/2001/XMLSchema-instance">
      <flightInfo>
        <gate xsi:type="xsd:int">10</gate>
        <status xsi:type="xsd:string">ON TIME</status>
      </flightInfo>
    </m:GetFlightInfoResponse>
  </SOAP:Body>
</SOAP:Envelope>
```

REST

REpresentation State Transfer

Stateless Client/Server Protocol: Principles

1. Each message in the protocol contains all the information needed by the receiver to understand and/or process it. This constraint attempts to "keep things simple" and avoid needless complexity

2. Set of Uniquely Addressable Resources
   - "Everything is a Resource" in a RESTful system
   - Requires universal syntax for resource identification (e.g. URI)

REST

3. Set of Well-Defined Operations that can be applied to all resources
   - In context of HTTP, the primary methods are
   - POST, GET, PUT, DELETE
   - these are similar (but not exactly) to the database notion of
   - CRUD (Create, Read, Update, Delete)

4. The use of Hypermedia both for Application Information and State Transitions
   - Resources are typically stored in a structured data format that supports hypermedia links, such as XHTML or XML

REST example

```
<user>
    <name>Jane</name>
    <gender>female</gender>
    <location href="http://www.example.org/us/ny/new_york">
       New York City, NY, USA</location>
</user>
```

This documentation is a representation used for the User resource

It might live at http://www.example.org/users/jane/

- If a user needs information about Jane, they GET this resource
- If they need to modify it, they GET it, modify it, and PUT it back
- The href to the Location resource allows savvy clients to gain access to its information with another simple GET request

Implication: Clients cannot be “thin”; need to understand resource formats

REST vs. RPC

In RPC systems, the design emphasis is on **verbs**

- What operations can I invoke on a system?
- getUser(), addUser(), removeUser(), updateUser(), getLocation(), updateLocation(), listUsers(), listLocations(), etc.

In REST systems, the design emphasis is on **nouns**

- User, Location
- In REST, you would define XML representations for these resources and then apply the standard methods to them

Outline for this Evening

- Some Ideas from Kandel et al. Paper
- Data Types and Sources
- Data Preparation and Manipulation
- Files

Files

- What is a file?
  - A **file** is a named sequence of **bytes**
    - Typically stored as a collection of pages (or blocks)
  - A **filesystem** is a collection of files organized within an hierarchical namespace
    - Responsible for laying out those bytes on physical media
    - Stores file metadata
    - Provides an API for interaction with files
  - Standard operations
    - open()/close()
    - seek()
    - read()/write()

Files

- Hierarchical namespace
  - / is known as the root of a filesystem
  - On Linux, the Filesystem Hierarchy Standard specifies which files live where
    - System executables in /usr/bin
    - Log files in /var/log
  - Permissions can be applied to all files beneath a directory
  - Files are not always arranged in a hierarchical namespace
    - Content-addressable storage (CAS)
    - Often used for large multimedia collections

File Formats

- Considerations for a file format
  - Data model: tabular, hierarchical, array
  - Physical layout
  - Field units and validation
  - Metadata: header, side file, specification, other?
  - Plain text or binary
  - Encoding: ASCII, UTF-8, other?
  - Delimiters and escaping
  - Compression, encryption, checksums?
  - Schema evolution

File Performance

Read/Write time (180 MB tabular file)

	Read Time (Text)	Write Time (Text)	Read Time (Binary)	Write Time (Binary)
Pandas (Python)	88 secs	107 secs	**	**
Scala/Java	12 secs	30 secs	0.5-2* secs	0.5-2* secs

** Pandas doesn't have a default binary file I/O library –
you can use Python, but performance depends on what you pick.

* 2 seconds is the time for sustainable read/write.
May be faster due to caching

File Performance - Compression

Read/Write time (180 MB tabular file, Scala/Java)

Binary File	Read Time	Write Time	File Size
Gzip level 6 (Java default)	1-2 secs	11 secs	35 MB
Gzip level 3	1-2 secs	4 secs	37 MB
Gzip level 1	1-2 secs	2 secs	38 MB
LZ4 fast	1-2 secs	1-2 secs	63 MB

Text File	Read Time	Write Time	File Size
Gzip level 6 (default)	1-2 secs	22 secs	32 MB
Gzip level 3	1-2 secs	14 secs	35 MB
Gzip level 1	1-2 secs	12 secs	39 MB
LZ4 fast	1-2 secs	11 secs	63 MB