Advanced Data Management (CSCI 640/490)

Data Wrangling

Dr. David Koop
Data Terminology

• Items
  - An item is an individual discrete entity
  - e.g., a row in a table

• Attributes
  - An attribute is some specific property that can be measured, observed, or logged
  - a.k.a. variable, (data) dimension
  - e.g., a column in a table
### Tables

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>S</th>
<th>T</th>
<th>U</th>
</tr>
</thead>
<tbody>
<tr>
<td>Order ID</td>
<td>Order Date</td>
<td>Order Priority</td>
<td>Product Container</td>
<td>Product Base Margin</td>
<td>Ship Date</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>10/14/06</td>
<td>5-Low</td>
<td>Large Container</td>
<td>0.8</td>
<td>10/21/06</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>2/21/08</td>
<td>4-Not Specified</td>
<td>Small Pack</td>
<td>0.55</td>
<td>2/22/08</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>7/16/07</td>
<td>2-High</td>
<td>Small Pack</td>
<td>0.79</td>
<td>7/17/07</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>7/16/07</td>
<td>2-High</td>
<td>Jumbo Box</td>
<td>0.03</td>
<td>7/17/07</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>7/16/07</td>
<td>2-High</td>
<td>Medium Box</td>
<td>0.00</td>
<td>7/18/07</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>7/16/07</td>
<td>2-High</td>
<td>Medium Box</td>
<td>0.03</td>
<td>7/18/07</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>10/23/07</td>
<td>4-Not Specified</td>
<td>Wrap Bag</td>
<td>0.52</td>
<td>10/24/07</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>10/23/07</td>
<td>4-Not Specified</td>
<td>Small Box</td>
<td>0.58</td>
<td>10/25/07</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>11/3/07</td>
<td>1-Urgent</td>
<td>Small Box</td>
<td>0.55</td>
<td>11/3/07</td>
<td></td>
</tr>
<tr>
<td>65</td>
<td>3/18/07</td>
<td>1-Urgent</td>
<td>Small Pack</td>
<td>0.49</td>
<td>3/19/07</td>
<td></td>
</tr>
<tr>
<td>66</td>
<td>3/18/07</td>
<td>1-Urgent</td>
<td>Wrap Bag</td>
<td>0.56</td>
<td>3/20/07</td>
<td></td>
</tr>
<tr>
<td>69</td>
<td>12/18/06</td>
<td>5-Low</td>
<td>Small Box</td>
<td>0.59</td>
<td>12/23/06</td>
<td></td>
</tr>
<tr>
<td>69</td>
<td>12/18/06</td>
<td>5-Low</td>
<td>Wrap Bag</td>
<td>0.6</td>
<td>12/23/06</td>
<td></td>
</tr>
<tr>
<td>96</td>
<td>4/17/05</td>
<td>2-High</td>
<td>Small Box</td>
<td>0.55</td>
<td>4/19/05</td>
<td></td>
</tr>
<tr>
<td>97</td>
<td>1/29/06</td>
<td>3-Medium</td>
<td>Small Box</td>
<td>0.38</td>
<td>1/30/06</td>
<td></td>
</tr>
<tr>
<td>129</td>
<td>11/19/08</td>
<td>5-Low</td>
<td>Small Box</td>
<td>0.37</td>
<td>11/28/08</td>
<td></td>
</tr>
<tr>
<td>130</td>
<td>5/8/08</td>
<td>2-High</td>
<td>Small Box</td>
<td>0.37</td>
<td>5/9/08</td>
<td></td>
</tr>
<tr>
<td>130</td>
<td>5/8/08</td>
<td>2-High</td>
<td>Medium Box</td>
<td>0.38</td>
<td>5/10/08</td>
<td></td>
</tr>
<tr>
<td>130</td>
<td>5/8/08</td>
<td>2-High</td>
<td>Small Box</td>
<td>0.6</td>
<td>5/11/08</td>
<td></td>
</tr>
<tr>
<td>132</td>
<td>6/11/06</td>
<td>3-Medium</td>
<td>Medium Box</td>
<td>0.6</td>
<td>6/12/06</td>
<td></td>
</tr>
<tr>
<td>132</td>
<td>6/11/06</td>
<td>3-Medium</td>
<td>Jumbo Box</td>
<td>0.69</td>
<td>6/14/06</td>
<td></td>
</tr>
<tr>
<td>134</td>
<td>5/1/08</td>
<td>4-Not Specified</td>
<td>Large Box</td>
<td>0.82</td>
<td>5/3/08</td>
<td></td>
</tr>
<tr>
<td>135</td>
<td>10/21/07</td>
<td>4-Not Specified</td>
<td>Small Pack</td>
<td>0.64</td>
<td>10/23/07</td>
<td></td>
</tr>
<tr>
<td>166</td>
<td>9/12/07</td>
<td>2-High</td>
<td>Small Box</td>
<td>0.55</td>
<td>9/14/07</td>
<td></td>
</tr>
<tr>
<td>193</td>
<td>8/8/06</td>
<td>1-Urgent</td>
<td>Medium Box</td>
<td>0.57</td>
<td>8/10/06</td>
<td></td>
</tr>
<tr>
<td>194</td>
<td>4/5/08</td>
<td>3-Medium</td>
<td>Wrap Bag</td>
<td>0.42</td>
<td>4/7/08</td>
<td></td>
</tr>
</tbody>
</table>
Categorial, Ordinal, and Quantitative

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>S</th>
<th>T</th>
<th>U</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Order ID</td>
<td>Order Date</td>
<td>Order Priority</td>
<td>Product Container</td>
<td>Product Base Margin</td>
</tr>
<tr>
<td>----</td>
<td>----------</td>
<td>-------------</td>
<td>----------------</td>
<td>------------------</td>
<td>---------------------</td>
</tr>
<tr>
<td>3</td>
<td>10/14/06</td>
<td>5-Low</td>
<td>Large Box</td>
<td>0.8</td>
<td>10/21/06</td>
</tr>
<tr>
<td>6</td>
<td>2/21/08</td>
<td>4-Not Specified</td>
<td>Small Pack</td>
<td>0.55</td>
<td>2/22/08</td>
</tr>
<tr>
<td>32</td>
<td>7/16/07</td>
<td>2-High</td>
<td>Small Pack</td>
<td>0.79</td>
<td>7/17/07</td>
</tr>
<tr>
<td>32</td>
<td>7/16/07</td>
<td>2-High</td>
<td>Jumbo Box</td>
<td>0.72</td>
<td>7/17/07</td>
</tr>
<tr>
<td>32</td>
<td>7/16/07</td>
<td>2-High</td>
<td>Medium Box</td>
<td>0.6</td>
<td>7/18/07</td>
</tr>
<tr>
<td>32</td>
<td>7/16/07</td>
<td>2-High</td>
<td>Medium Box</td>
<td>0.65</td>
<td>7/18/07</td>
</tr>
<tr>
<td>35</td>
<td>10/23/07</td>
<td>4-Not Specified</td>
<td>Wrap Bag</td>
<td>0.52</td>
<td>10/24/07</td>
</tr>
<tr>
<td>35</td>
<td>10/23/07</td>
<td>4-Not Specified</td>
<td>Small Box</td>
<td>0.58</td>
<td>10/25/07</td>
</tr>
<tr>
<td>36</td>
<td>11/3/07</td>
<td>1-Urgent</td>
<td>Small Box</td>
<td>0.55</td>
<td>11/3/07</td>
</tr>
<tr>
<td>65</td>
<td>3/18/07</td>
<td>1-Urgent</td>
<td>Small Pack</td>
<td>0.49</td>
<td>3/19/07</td>
</tr>
<tr>
<td>66</td>
<td>1/20/05</td>
<td>5-Low</td>
<td>Wrap Bag</td>
<td>0.56</td>
<td>1/20/05</td>
</tr>
<tr>
<td>69</td>
<td>6/4/05</td>
<td>4-Not Specified</td>
<td>Small Pack</td>
<td>0.44</td>
<td>6/6/05</td>
</tr>
<tr>
<td>69</td>
<td>6/4/05</td>
<td>4-Not Specified</td>
<td>Small Pack</td>
<td>0.6</td>
<td>6/6/05</td>
</tr>
<tr>
<td>70</td>
<td>12/18/06</td>
<td>5-Low</td>
<td>Small Box</td>
<td>0.59</td>
<td>12/23/06</td>
</tr>
<tr>
<td>70</td>
<td>12/18/06</td>
<td>5-Low</td>
<td>Small Box</td>
<td>0.59</td>
<td>12/23/06</td>
</tr>
<tr>
<td>96</td>
<td>4/17/05</td>
<td>2-High</td>
<td>Medium Box</td>
<td>0.55</td>
<td>4/19/05</td>
</tr>
<tr>
<td>97</td>
<td>1/29/06</td>
<td>3-Medium</td>
<td>Small Box</td>
<td>0.58</td>
<td>1/30/06</td>
</tr>
<tr>
<td>129</td>
<td>11/19/08</td>
<td>5-Low</td>
<td>Small Box</td>
<td>0.37</td>
<td>11/28/08</td>
</tr>
<tr>
<td>130</td>
<td>5/8/08</td>
<td>2-High</td>
<td>Medium Box</td>
<td>0.38</td>
<td>5/9/08</td>
</tr>
<tr>
<td>130</td>
<td>5/8/08</td>
<td>2-High</td>
<td>Small Box</td>
<td>0.38</td>
<td>5/10/08</td>
</tr>
<tr>
<td>130</td>
<td>5/8/08</td>
<td>2-High</td>
<td>Small Box</td>
<td>0.6</td>
<td>5/11/08</td>
</tr>
<tr>
<td>132</td>
<td>6/11/06</td>
<td>3-Medium</td>
<td>Medium Box</td>
<td>0.6</td>
<td>6/12/06</td>
</tr>
<tr>
<td>132</td>
<td>6/11/06</td>
<td>3-Medium</td>
<td>Jumbo Box</td>
<td>0.69</td>
<td>6/14/06</td>
</tr>
<tr>
<td>134</td>
<td>5/1/08</td>
<td>4-Not Specified</td>
<td>Large Box</td>
<td>0.82</td>
<td>5/3/08</td>
</tr>
<tr>
<td>135</td>
<td>10/21/07</td>
<td>4-Not Specified</td>
<td>Small Pack</td>
<td>0.64</td>
<td>10/23/07</td>
</tr>
<tr>
<td>166</td>
<td>9/12/07</td>
<td>2-High</td>
<td>Small Box</td>
<td>0.55</td>
<td>9/14/07</td>
</tr>
<tr>
<td>193</td>
<td>8/8/06</td>
<td>1-Urgent</td>
<td>Medium Box</td>
<td>0.57</td>
<td>8/10/06</td>
</tr>
<tr>
<td>194</td>
<td>4/5/08</td>
<td>3-Medium</td>
<td>Wrap Bag</td>
<td>0.42</td>
<td>4/7/08</td>
</tr>
</tbody>
</table>

quantitative
ordinal
categorical
Sequential and Diverging Data

- **Sequential:** homogenous range from a minimum to a maximum
  - Examples: Land elevations, ocean depths

- **Diverging:** can be deconstructed into two sequences pointing in opposite directions
  - Has a **zero point** (not necessary 0)
  - Example: Map of both land elevation and ocean depth

[Rogowitz & Treinish, 1998]
Sequential and Diverging Data

- **Sequential**: homogenous range from a minimum to a maximum
  - Examples: Land elevations, ocean depths

- **Diverging**: can be deconstructed into two sequences pointing in opposite directions
  - Has a **zero point** (not necessary 0)
  - Example: Map of both land elevation and ocean depth

[Rogowitz & Treinish, 1998]
Semantics

• The meaning of the data
• Example: 94023, 90210, 02747, 60115
Semantics

• The meaning of the data
• Example: 94023, 90210, 02747, 60115
  - Attendance at college football games?
Semantics

• The meaning of the data
• Example: 94023, 90210, 02747, 60115
  - Attendance at college football games?
  - Salaries?
Semantics

• The meaning of the data
• Example: 94023, 90210, 02747, 60115
  - Attendance at college football games?
  - Salaries?
  - Zip codes?
• Cannot always infer based on what the data looks like
• Often require semantics to better understand data, column names help
• May also include rules about data: a zip code is part of an address that uniquely identifies a residence
• Useful for asking good questions about the data
Data Model vs. Conceptual Model

• Data Model: raw data that has a specific data type (e.g. floats):
  - Temperature Example: [32.5, 54.0, -17.3] (floats)

• Conceptual Model: how we think about the data
  - Includes semantics, reasoning
  - Temperature Example:
    • Quantitative: [32.50, 54.00, -17.30]
Data Model vs. Conceptual Model

• Data Model: raw data that has a specific data type (e.g. floats):
  - Temperature Example: [32.5, 54.0, -17.3] (floats)

• Conceptual Model: how we think about the data
  - Includes semantics, reasoning
  - Temperature Example:
    • Quantitative: [32.50, 54.00, -17.30]
    • Ordered: [warm, hot, cold]

[via A. Lex, 2015]
Data Model vs. Conceptual Model

• Data Model: raw data that has a specific data type (e.g. floats):
  - Temperature Example: [32.5, 54.0, -17.3] (floats)

• Conceptual Model: how we think about the data
  - Includes semantics, reasoning
  - Temperature Example:
    • Quantitative: [32.50, 54.00, -17.30]
    • Ordered: [warm, hot, cold]
    • Categorical: [not burned, burned, not burned]
Derived Data
Derived Data

- Often, data in its original form isn't as useful as we would like
- Examples: Data about a basketball team's games
Derived Data

- Often, data in its original form isn't as useful as we would like
- Examples: Data about a basketball team's games
- Example 1: 1stHalfPoints, 2ndHalfPoints
  - More useful to know total number of points
  - Points = 1stHalfPoints + 2ndHalfPoints
Derived Data

- Often, data in its original form isn't as useful as we would like.
- Examples: Data about a basketball team's games.
- Example 1: 1stHalfPoints, 2ndHalfPoints
  - More useful to know total number of points.
  - Points = 1stHalfPoints + 2ndHalfPoints.
- Example 2: Points, OpponentPoints
  - Want to have a column indicating win/loss.
  - Win = True if (Points > OpponentPoints) else False.
Derived Data

• Often, data in its original form isn't as useful as we would like

• Examples: Data about a basketball team's games

• Example 1: 1stHalfPoints, 2ndHalfPoints
  - More useful to know total number of points
  - Points = 1stHalfPoints + 2ndHalfPoints

• Example 2: Points, OpponentPoints
  - Want to have a column indicating win/loss
  - Win = True if (Points > OpponentPoints) else False

• Example 3: Points
  - Want to have a column indicating how that point total ranks
  - Rank = index in sorted list of all Point values
Assignment 2

• Assignment 1 Questions with pandas, DuckDB, and polars
• CS 640 students do all, CS 490 do pandas & DuckDB (polars is EC)
• Can work by framework or by query
• Most questions can be answered with a single statement… but that statement can take a while to write
  - Read documentation
  - Check hints
Next Week

- No in-person lectures
- You will work through courselets on data wrangling and data cleaning
Test 1

• Move to Wednesday, Feb. 28?
  - No one has contacted me so I plan to move to Feb. 28

• Will cover topics through the courselets

• Format:
  - Multiple Choice
  - Free Response: longer-form questions that involve multiple steps, responding to readings
  - CSCI 640 students have an extra two pages
What if data isn't correct/trustworthy/in the right format?
Dirty Data
Geolocation Errors

- Maxmind helps companies determine where users are located based on IP address
- "How a quiet Kansas home wound up with 600 million IP addresses and a world of trouble" [Washington Post, 2016]
Numeric Outliers

ages of employees (US)

- median 37
- mean 58.52632
- variance 9252.041
FINDINGS

we got about the future of the data science, the most salient takeaway was how excited our respondents were about the evolution of the field. They cited things in their own practice, how they saw their jobs getting more interesting and less repetitive, all while expressing a real and broad enthusiasm about the value of the work in their organization.

As data science becomes more commonplace and simultaneously a bit demystified, we expect this trend to continue as well. After all, last year's respondents were just as excited about their work (about 79% were "satisfied" or better).

How a Data Scientist Spends Their Day

Here's where the popular view of data scientists diverges pretty significantly from reality. Generally, we think of data scientists building algorithms, exploring data, and doing predictive analysis. That's actually not what they spend most of their time doing, however.

As you can see from the chart above, 3 out of every 5 data scientists we surveyed actually spend the most time cleaning and organizing data. You may have heard this referred to as "data wrangling" or compared to digital janitor work. Everything from list verification to removing commas to debugging databases—that time adds up and it adds up immensely. Messy data is by far the more time-consuming aspect of the typical data scientist's work flow. And nearly 60% said they simply spent too much time doing it.

Data scientist job satisfaction

60%

19%

9%

4%

3%

Building training sets: 3%

Cleaning and organizing data: 60%

Collecting data sets: 19%

Mining data for patterns: 9%

Refining algorithms: 4%

Other: 5%

This takes a lot of time!

[CrowdFlower Data Science Report, 2016]
Why That's a Problem

Simply put, data wrangling isn't fun. It takes forever. In fact, a few years back, the New York Times estimated that up to 80% of a data scientist's time is spent doing this sort of work. Here, it's necessary to point out that data cleaning is incredibly important. You can't do the sort of work data scientists truly enjoy doing with messy data. It needs to be cleaned, labeled, and enriched before you can trust the output.

The problem here is two-fold. One: data scientists simply don't like doing this kind of work, and, as mentioned, this kind of work takes up most of their time. We asked our respondents what was the least enjoyable part of their job. They had this to say:

Note how those last two charts mirror each other. The things data scientists do most are the things they enjoy least. Last year, we found that respondents far prefer doing the more creative, interesting parts of their job, things like predictive analysis and mining data for patterns. That's where the real value comes. But again, you simply can't do that work unless the data is properly labeled. And nobody likes labeling data.

Do Data Scientists Have What They Need?

With a shortage of data scientists out there in the world, we wanted to find out if they thought they were properly supported in their job. After all, when you need more data scientists, you'll often find a single person doing the work of several.

What's the least enjoyable part of data science?

- Building training sets: 10%
- Cleaning and organizing data: 57%
- Collecting data sets: 21%
- Mining data for patterns: 3%
- Refining algorithms: 4%
- Other: 5%

[CrowdFlower Data Science Report, 2016]
Dirty Data: Statistician's View

• Some process produces the data
• Want a model but have non-ideal samples:
  - Distortion: some samples corrupted by a process
  - Selection bias: likelihood of a sample depends on its value
  - Left and right censorship: users come and go from scrutiny
  - Dependence: samples are not independent (e.g. social networks)
• You can add/augment models for different problems, but cannot model everything
• Trade-off between accuracy and simplicity

[J. Canny et al.]
Dirty Data: Database Expert's View

- Got a dataset
- Some values are missing, corrupted, wrong, duplicated
- Results are absolute (relational model)
- Better answers come from improving the quality of values in the dataset

[J. Canny et al.]
Dirty Data: Domain Expert's View

- Data doesn't look right
- Answer doesn't look right
- What happened?
- Domain experts carry an implicit model of the data they test against
- You don't always need to be a domain expert to do this
  - Can a person run 50 miles an hour?
  - Can a mountain on Earth be 50,000 feet above sea level?
  - Use common sense
Dirty Data: Data Scientist's View

• Combination of the previous three views
• All of the views present problems with the data
• The goal may dictate the solutions:
  - Median value: don't worry too much about crazy outliers
  - Generally, aggregation is less susceptible by numeric errors
  - Be careful, the data may be correct…
Be careful how you detect dirty data

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

  – National Center for Atmospheric Research
Where does dirty data originate?

- Source data is bad, e.g. person entered it incorrectly
- Transformations corrupt the data, e.g. certain values processed incorrectly due to a software bug
- Integration of different datasets causes problems
- Error propagation: one error is magnified
Types of Dirty Data Problems

- **Separator Issues**: e.g. CSV without respecting double quotes
  - 12, 13, "Doe, John", 45

- **Naming Conventions**: NYC vs. New York

- **Missing required fields**, e.g. key

- **Different representations**: 2 vs. two

- **Truncated data**: "Janice Keihanaikukauakahihuliheekahaunaele" becomes "Janice Keihanaikukauakahihuliheek" on Hawaii license

- **Redundant records**: may be exactly the same or have some overlap

- **Formatting issues**: 2017-11-07 vs. 07/11/2017 vs. 11/07/2017

[J. Canny et al.]
Data Wrangling

- Data wrangling: transform raw data to a more meaningful format that can be better analyzed
- Data cleaning: getting rid of inaccurate data
- Data transformations: changing the data from one representation to another
- Data reshaping: reorganizing the data
- Data merging: combining two datasets
Data Cleaning
Wrangler: Interactive Visual Specification of Data Transformation Scripts

S. Kandel, A. Paepcke, J. Hellerstein, J. Heer
Wrangler

- Data cleaning takes a lot of **time** and **human effort**
- "Tedium is the message"
- Repeating this process on multiple data sets is even worse!
- Solution:
  - interactive interface (mixed-initiative)
  - transformation language with natural language "translations"
  - suggestions + "programming by demonstration"
Your Critique/Questions
Example Critique

• Summary: Wrangler tackles data wrangling tasks by combining a language for specifying operations with an interface allowing users to specify the types of changes they are interested; the system can then generate suggested operations and demonstrates them on demand.

• Critique: The suggestions may lead to states that a user cannot recover from easily. Suppose a suggestion looks like it works well, but a user later realizes was incorrect. They can backtrack, but it's often unclear where to and which other path to take. In addition, a user has to have some idea of the constructs of the language in order to edit parameters. Without a good idea of the impact of the parameters, the work may become as tedious as manual correction. Perhaps a more example-based strategy could help.
Previous Work: Potter's Wheel

• V. Raman and J. Hellerstein, 2001
• Defines structure extractions for identifying fields
• Defines transformations on the data
• Allows user interaction
## Potter's Wheel: Structure Extraction

<table>
<thead>
<tr>
<th>Example Column Value (Example erroneous values)</th>
<th># Structures Enumerated</th>
<th>Final Structure Chosen (Punc = Punctuation)</th>
</tr>
</thead>
<tbody>
<tr>
<td>-60</td>
<td>5</td>
<td>Integer</td>
</tr>
<tr>
<td>UNITED, DELTA, AMERICAN etc.</td>
<td>5</td>
<td>IspellWord</td>
</tr>
<tr>
<td>SFO, LAX etc. (JFK to OAK)</td>
<td>12</td>
<td>AllCapsWord</td>
</tr>
<tr>
<td>1998/01/12</td>
<td>9</td>
<td>Int Punc(/) Int Punc(/) Int</td>
</tr>
<tr>
<td>M, Tu, Thu etc.</td>
<td>5</td>
<td>Capitalized Word</td>
</tr>
<tr>
<td>06:22</td>
<td>5</td>
<td>Int(len 2) Punc(:) Int(len 2)</td>
</tr>
<tr>
<td>12.8.15.147 (ferret03.webtop.com)</td>
<td>9</td>
<td>Double Punc(‘.’) Double</td>
</tr>
<tr>
<td>”GET\b (\b)</td>
<td>5</td>
<td>Punc(””) IspellWord Punc()</td>
</tr>
<tr>
<td>/postmodern/lecs/xia/sld013.htm</td>
<td>4</td>
<td>ξ*</td>
</tr>
<tr>
<td>HTTP</td>
<td>3</td>
<td>AllCapsWord(HTTP)</td>
</tr>
<tr>
<td>/1.0</td>
<td>6</td>
<td>Punc(/) Double(1.0)</td>
</tr>
</tbody>
</table>

[\[V. Raman and J. Hellerstein, 2001\]

---

D. Koop, CSCI 640/490, Spring 2023
Potter's Wheel: Transforms

<table>
<thead>
<tr>
<th>Transform</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Format</td>
<td>$\phi(R, i, f) = {(a_1, \ldots, a_{i-1}, a_i, \ldots, a_n, f(a_i)) \mid (a_1, \ldots, a_n) \in R}$</td>
</tr>
<tr>
<td>Add</td>
<td>$\alpha(R, x) = {(a_1, \ldots, a_n, x) \mid (a_1, \ldots, a_n) \in R}$</td>
</tr>
<tr>
<td>Drop</td>
<td>$\pi(R, i) = {(a_1, \ldots, a_{i-1}, a_i, \ldots, a_n) \mid (a_1, \ldots, a_n) \in R}$</td>
</tr>
<tr>
<td>Copy</td>
<td>$\kappa((a_1, \ldots, a_n), i) = {(a_1, \ldots, a_n, a_i) \mid (a_1, \ldots, a_n) \in R}$</td>
</tr>
<tr>
<td>Merge</td>
<td>$\mu((a_1, \ldots, a_n), i, j, \text{glue}) = {(a_1, \ldots, a_{i-1}, a_i, a_{i+1}, \ldots, a_j, a_{j+1}, \ldots, a_n, a_i \oplus \text{glue} \oplus a_j) \mid (a_1, \ldots, a_n) \in R}$</td>
</tr>
<tr>
<td>Split</td>
<td>$\omega((a_1, \ldots, a_n), i, \text{splitter}) = {(a_1, \ldots, a_{i-1}, a_i, a_{i+1}, \ldots, a_n, a_i, \text{null}) \mid (a_1, \ldots, a_n) \in R \land \text{pred}(a_i)} \cup {(a_1, \ldots, a_{i-1}, a_i, a_{i+1}, \ldots, a_n, \text{null}, a_i) \mid (a_1, \ldots, a_n) \in R \land \neg\text{pred}(a_i)}$</td>
</tr>
<tr>
<td>Divide</td>
<td>$\delta((a_1, \ldots, a_n), i, \text{pred}) = {(a_1, \ldots, a_{i-1}, a_i, a_{i+1}, \ldots, a_n, \text{null}) \mid (a_1, \ldots, a_n) \in R \land \text{pred}(a_i)}$</td>
</tr>
<tr>
<td>Fold</td>
<td>$\lambda(R, i_1, i_2, \ldots, i_k) = {(a_1, \ldots, a_{i_1-1}, a_{i_1}, a_{i_1+1}, \ldots, a_{i_2-1}, a_{i_2}, a_{i_2+1}, \ldots, a_{i_k-1}, a_{i_k}, a_{i_k+1}, \ldots, a_n, a_{i_1}) \mid (a_1, \ldots, a_n) \in R \land 1 \leq l \leq k}$</td>
</tr>
<tr>
<td>Select</td>
<td>$\sigma(R, \text{pred}) = {(a_1, \ldots, a_n) \mid (a_1, \ldots, a_n) \in R \land \text{pred}(a_1, \ldots, a_n)}$</td>
</tr>
</tbody>
</table>

**Notation:** $R$ is a relation with $n$ columns. $i, j$ are column indices and $a_i$ represents the value of a column in a row. $x$ and glue are values. $f$ is a function mapping values to values. $x \oplus y$ concatenates $x$ and $y$. splitter is a position in a string or a regular expression, left$(x, \text{splitter})$ is the left part of $x$ after splitting by splitter. pred is a function returning a boolean.

[V. Raman and J. Hellerstein, 2001]
Potter's Wheel: Example

<table>
<thead>
<tr>
<th>Anna</th>
<th>Davis</th>
<th>Stewart, Bob</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Format

'(.*)', '(.*>' to "2 \1"

Anna Davis

Jerry Dole

Joan Marsh

Bob Stewart

Split at '

<table>
<thead>
<tr>
<th>Bob</th>
<th>Stewart</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anna</td>
<td>Davis</td>
</tr>
<tr>
<td>Jerry</td>
<td>Dole</td>
</tr>
<tr>
<td>Joan</td>
<td>Marsh</td>
</tr>
</tbody>
</table>

2 Merges

Anna Davis

Jerry Dole

Joan Marsh

Bob Stewart

[V. Raman and J. Hellerstein, 2001]
To study the effect of parsing according to specificity and consider structures starting with the least specific, we have decomposed datasets into much smaller substrings, and the discrepancies after each transform. This approach also solves the ambiguous data lineage problem of whether a discrepancy is due to an error in the interface or because of a poor transform. If the user wishes to know the lineage of a particular discrepancy, the system only needs to apply the transforms one after another, checking for undos as described in the introduction. [6, 9]

4.4 Undoing Transforms and Tracking Data Lineage

Undoing these requires physical undo, which is user specified split position). The ability to undo incorrect transforms is an important requirement for interactive transformation. However, if the system only considers structures starting with the least specific, as does Potter's Wheel, it never changes the actual data. Instead, Potter's Wheel only collects transforms as the user adds them, and applies them only on the records displayed on the screen, and auditing tools, as described in the introduction. [6, 9]

5 Related Work

Extracting structure from poorly structured data is increasingly important for "wrapping" data from web pages, and many tools exist in both the research and commercial world. (e.g., Clio [19]). As discussed in Section 4.3, these methods are useful as default algorithms for Potter's Wheel's discrepancy detector. However we believe that a semi-automatic, interactive approach is robust to structural data errors. Also, for detecting discrepancies it is important to infer structures in terms of generic user-defined domains, in a way that is user specified split position). Such general purpose algorithms are useful as default algorithms for Potter's Wheel's discrepancy detector. However we believe that in many contexts such as data warehousing and data integration of schemas from various data sources. We intend to extend our focus is on the ease of specification and incremental application, and not merely on expressive power.

We have run experiments comparing the throughput at which one can split values using these methods. We see that one; it illustrates how crucial the choice of starting specificity is. Figure 12 compares the throughput at which one can split values using these methods. We see that performs much better than the others, with the improvement being dramatic at splits involving many structures.

Figure 10: Parse structures inferred from various split-by-examples

| Example Values Split By User (| is user specified split position) | Inferred Structure | Comments |
|---------------------------------|--------------------|-----------|
| Taylor, Jane |, $52,072 | Blair, John |, $73,238 | Tony Smith |, $1,00,533 |
| MAA |to| SIN | JFK |to| SFO | LAX |–| ORD | SEA |/| OAK |
| 321 Blake #7 |, Berkeley |, CA 94720 | 719 MLK Road |, Fremont |, CA 95743 |
| (<len 3 identifier> < ξ* > < len 3 identifier>) |
| (<ξ* < ‘’, Money>) |
| Parsing is doable despite no good delimiter. A regular expression domain can infer a structure of $[0-9,* for last component. |
| Parsing is possible despite multiple delimiters. |
| Parsing is easy because of consistent delimiter. |

[V. Raman and J. Hellerstein, 2001]
Wrangler Transformation Language

- Based on Potter's Wheel
- Map: Delete, Extract, Cut, Split, Update
- Lookup/join: Use external data (e.g. from zipcode → state)
- Reshape: Fold and Unfold (aka pivot)
- Positional: Fill and lag
- Sorting, aggregation, key generation, schema transforms
Interface

- Automated Transformation Suggestions
- Editable Natural Language Explanations
- Fill Bangladesh by copying values from above
- Fill Bangladesh by interpolating the 5 values from above
- Fill Bangladesh by averaging values from above
- Visual Transformation Previews
- Transformation History

[S. Kandel et al., 2011]
Automation from past actions

- Infer parameter sets from user interaction
- Generating transforms
- Ranking and ordering transformations:
  - Based on user preferences, difficulty, and corpus frequency
  - Sort transforms by type and diversify suggestions

[S. Kandel et al., 2011]
Data Wrangler Demo

- [http://vis.stanford.edu/wrangler/app/](http://vis.stanford.edu/wrangler/app/)

<table>
<thead>
<tr>
<th>Transform Script</th>
<th>Import Export</th>
</tr>
</thead>
<tbody>
<tr>
<td>▶ Split data repeatedly on newline into rows</td>
<td></td>
</tr>
<tr>
<td>▶ Split split repeatedly on ','</td>
<td></td>
</tr>
<tr>
<td>▶ Promote row 0 to header</td>
<td></td>
</tr>
<tr>
<td>Text</td>
<td>Columns</td>
</tr>
</tbody>
</table>

Delete row 7

Delete empty rows

Fill row 7 by copying values from above

<table>
<thead>
<tr>
<th>Year</th>
<th>Property_crime_rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>2004</td>
<td>4029.3</td>
</tr>
<tr>
<td>2005</td>
<td>3900</td>
</tr>
<tr>
<td>2006</td>
<td>3937</td>
</tr>
<tr>
<td>2007</td>
<td>3974.9</td>
</tr>
<tr>
<td>2008</td>
<td>4081.9</td>
</tr>
<tr>
<td>2009</td>
<td>4029.3</td>
</tr>
<tr>
<td>2010</td>
<td>3370.9</td>
</tr>
<tr>
<td>2011</td>
<td>3615</td>
</tr>
<tr>
<td>2012</td>
<td>3582</td>
</tr>
</tbody>
</table>
Evaluation

• Compare with Excel
• Tests:
  - Extract text from a single string entry
  - Fill in missing values with estimates
  - Reshape tables
• Allowed users to ask questions about Excel, not Wrangler
• Found significant effect of tool and users found previews and suggestions helpful
• Complaint: No manual fallback, make implications of user choices more obvious for users
Task Completion Times

We performed a repeated-measures ANOVA of completion times with task, tool, and Excel novice/expert performance by experts (regardless of tool) in the reshaping task (T3). No other interactions were significant. We divided subjects into “novices” and “experts” according to their prior experience with Excel on a 10-point scale (1 being less experienced than experts). We found a significant interaction effect of task (T1, T2, T3) and tool (Wrangler, Excel), Wilks’ λ = 0.01, F(2, 54) = 17.33, p<0.001, partial η² = 0.596. We found a significant interaction effect of task (T1, T2, T3) and Excel novice/expert, Wilks’ λ = 0.30, F(2, 54) = 11.10, p<0.001, partial η² = 0.54. We also found a significant interaction effect of tool (Wrangler, Excel) and Excel novice/expert, Wilks’ λ = 0.943, F(1, 54) = 4.3, p<0.05, partial η² = 0.05.

Users rated previews for comparison: all subjects use it regularly and half self-reported their median self-reported expertise rating (5). We performed a repeated-measures ANOVA of ratings (1 = not useful, 5 = most useful) with task, tool, and Excel novice/expert. We found a significant interaction effect of task (T1, T2, T3) and tool (Wrangler, Excel), Wilks’ λ = 0.596, F(2, 54) = 4.8, p<0.05, partial η² = 0.22. We found a significant interaction effect of task (T1, T2, T3) and Excel novice/expert, Wilks’ λ = 3.3, F(2, 54) = 2.5, p<0.05, partial η² = 0.07. We also found a significant interaction effect of tool (Wrangler, Excel) and Excel novice/expert, Wilks’ λ = 0.47, F(1, 54) = 3.3, p<0.05, partial η² = 0.06.

Figure 11. Task completion times. Black bars indicate median values.

[T. Kandel et al., 2011]
Improvements in Prediction

Update suggestions when given more information

[Heer et al., 2015]
Data Wrangling Tasks

- Unboxing: Discovery & Assessment: What's in there? (types, distribution)
- Structuring: Restructure data (table, nested data, pivot tables)
- Cleaning: does data match expectations (often involves user)
- Enriching & Blending: Adding new data
- Optimizing & Publishing: Structure for storage or visualization

[J. M. Hellerstein et al., 2018]
Differences with Extract-Transform-Load (ETL)

- **ETL:**
  - Who: IT Professionals
  - Why: Create static data pipeline
  - What: Structured data
  - Where: Data centers

- "Modern Data Preparation":
  - Who: Analysts
  - Why: Solve problems by designing recipes to use data
  - What: Original, custom data blended with other data
  - Where: Cloud, desktop

[J. M. Hellerstein et al., 2018]
Evolution of Wrangler

• Authors started a company, Trifacta
• Eventually bought by Alteryx
• Now known as Alteryx Designer Cloud
• Offer Free Student Licenses: Link