Wrangler: Interactive Visual Specification of Data Transformation Scripts

Jacob Yim
Role: Paper Author

Original paper by Sean Kandel, Andreas Paepcke, Joseph Hellerstein and Jeffrey Heer (CHI 2011)

Problem Statement

- Data cleaning is hard!
- Restructuring data for analysis and improving data quality is time-consuming
  - Estimated to take up 80% of dev time and cost in data warehousing projects
- Transforms can be difficult to specify and evaluate
  - Discourages people from working with data
- Output of data wrangling must be editable and auditable, reusable and shareable
- Introducing Wrangler, an interactive system for data transformation

Related Work

- Many tools (SWYN, Potluck, Karma, Vegemite) use direct manipulation and programming by demonstration for specific cleaning tasks
  - Lack operations like reshaping, aggregation, and missing value imputation
  - Most do not generate scripts for further reuse
- Toped++ validates and transforms data
  - Does not support filtering, reshaping, or aggregation
- Potter's Wheel and Ajax for interactive data cleaning
  - Neither tool supports much direct manipulation
  - Wrangler's transformation language extends Potter's Wheel language

Transform Script Import Export

- Split **data repeatedly** on **newline** into **rows**
- Split **split repeatedly** on **','**
- Promote **row 0** to header

Text Columns Rows Table Clear

Delete **row 7**

Delete **empty rows**

Fill **row 7** by **copying** values from **above**

	Year	Property_crime_rate
0	Reported crime in Alabama	
1		
2	2004	4029.3
3	2005	3900
4	2006	3937
5	2007	3974.9
6	2008	4081.9
7		
8	Reported crime in Alaska	
9		
10	2004	3370.9
11	2005	3615
12	2006	3582

Usage Scenario

User uploads a CSV. Wrangler automatically identifies CSV format and applies transformations to tabularize the data.

Transform Script Import Export

- Split **data repeatedly** on **newline** into **rows**
- Split **split repeatedly** on **','**
- Promote **row 0** to header

Text Columns Rows Table Clear

Delete **row 7**

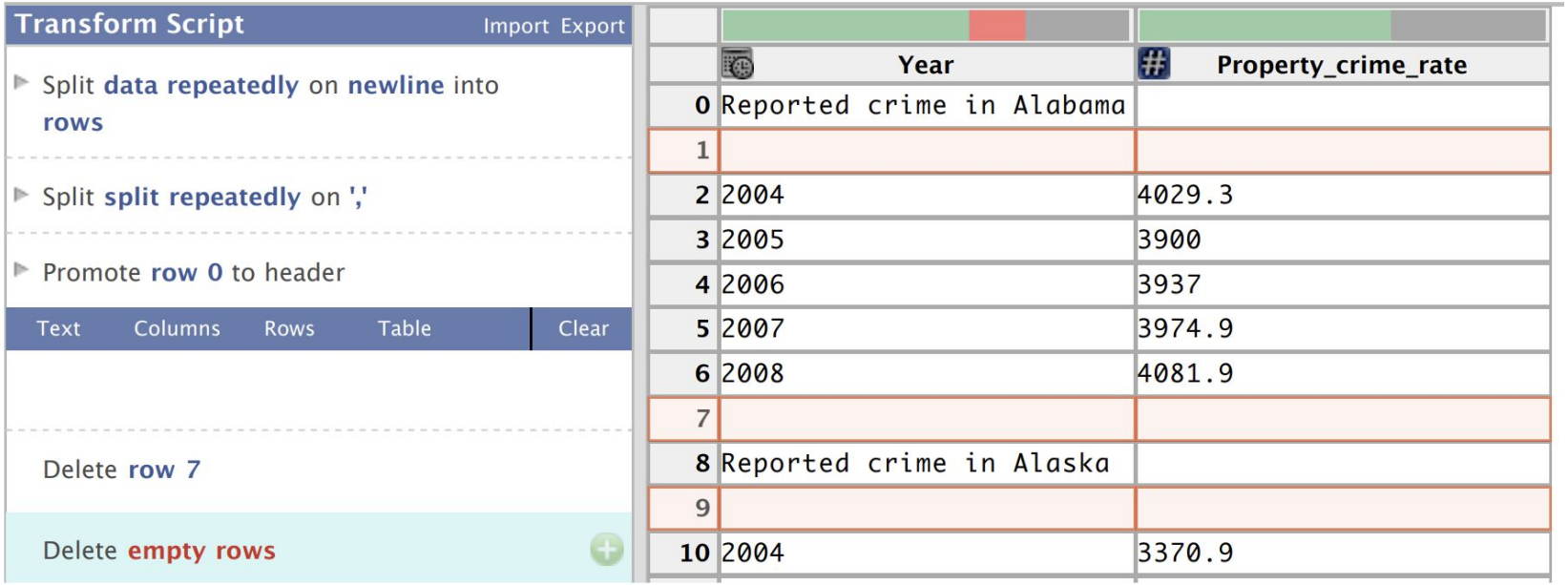
Delete **empty rows**

Fill **row 7** by **copying** values from **above**

	Year	Property_crime_rate
0	Reported crime in Alabama	
1		
2	2004	4029.3
3	2005	3900
4	2006	3937
5	2007	3974.9
6	2008	4081.9
7		
8	Reported crime in Alaska	
9		
10	2004	3370.9
11	2005	3615
12	2006	3582

Usage Scenario

User clicks an empty row to select it, and Wrangler suggests to delete empty rows.



Usage Scenario

User selects state names to extract them into a new column, previewed in yellow.

Transform Script Import Export

- Split **data repeatedly** on **newline** into **rows**
- Split **split repeatedly** on **','**
- Promote **row 0** to header
- Delete **empty rows**

Text Columns Rows Table Clear

Extract from **Year** after **'in '**

Extract from **Year** after **' in '**

Cut from **Year** after **'in '**

	Year	extract	Property_
0	Reported crime in Alabama	Alabama	
1	2004		4029.3
2	2005		3900
3	2006		3937
4	2007		3974.9
5	2008		4081.9
6	Reported crime in Alaska	Alaska	
7	2004		3370.9
8	2005		3615
9	2006		3582
10	2007		3373.9
11	2008		2928.3
12	Reported crime in Arizona	Arizona	
13	2004		5073.3
14	2005		4837

Usage Scenario

User clicks the gray bar to fill missing values, and chooses from the suggestions.

Transform Script Import Export

- Split **data repeatedly** on **newline** into **rows**
- Split **split repeatedly** on **','**
- Promote **row 0** to header
- Delete **empty rows**

Text Columns Rows Table Clear

Extract from **Year** after **'in '**

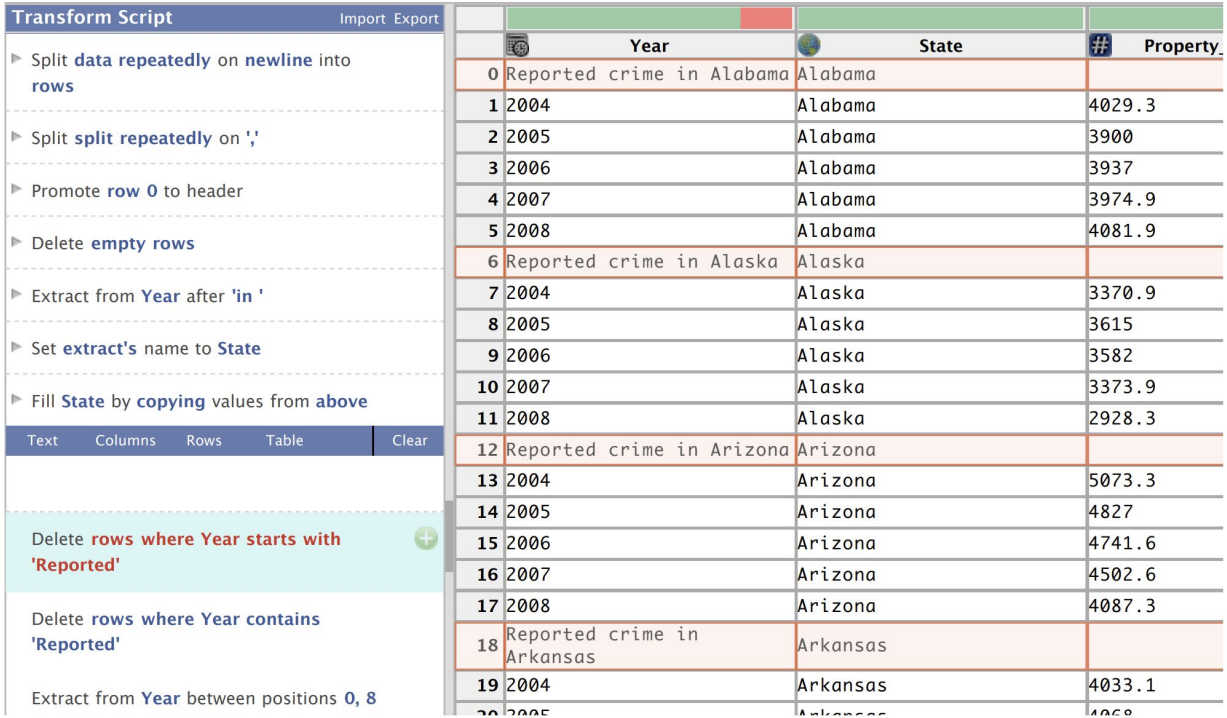
Extract from **Year** after **' in '**

Cut from **Year** after **'in '**

	Year	extract	Property_
0	Reported crime in Alabama	Alabama	
1	2004		4029.3
2	2005		3900
3	2006		3937
4	2007		3974.9
5	2008		4081.9
6	Reported crime in Alaska	Alaska	
7	2004		3370.9
8	2005		3615
9	2006		3582
10	2007		3373.9
11	2008		2928.3
12	Reported crime in Arizona	Arizona	
13	2004		5073.3
14	2005		4837

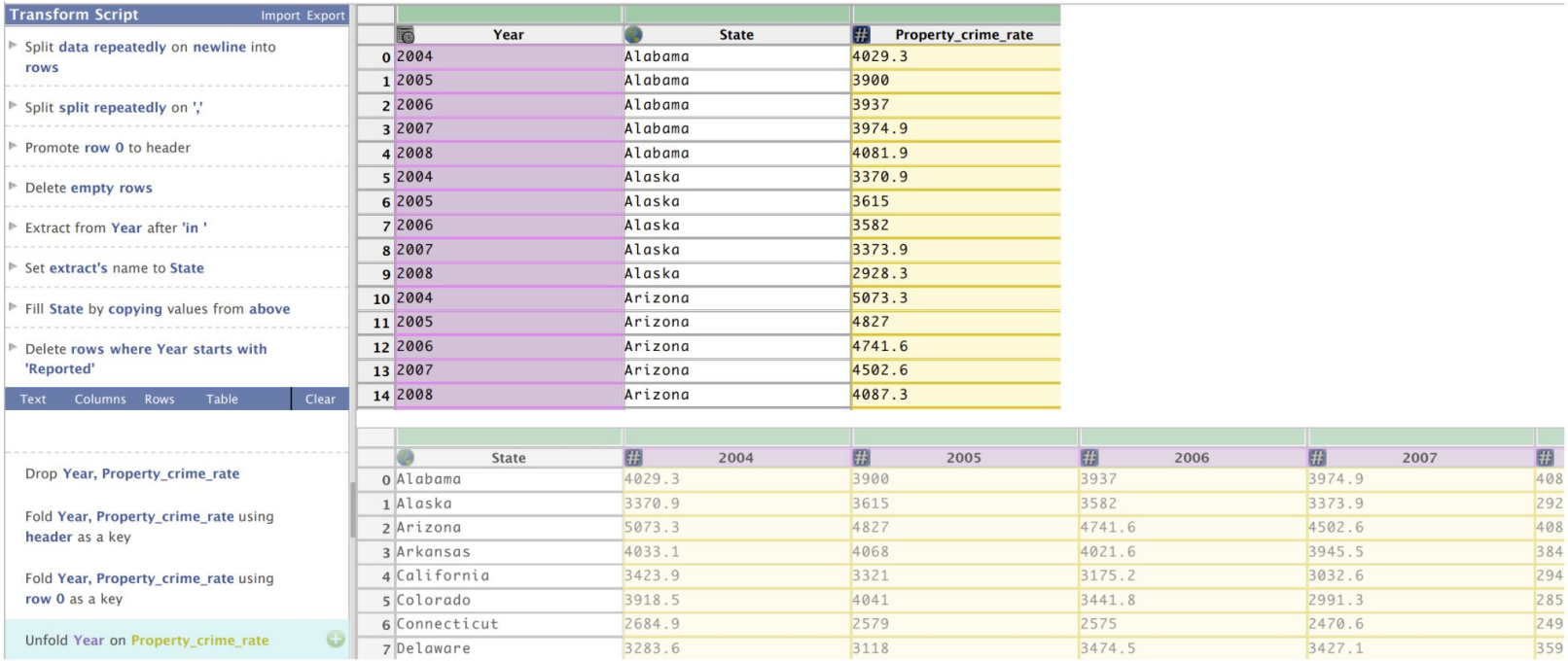
Usage Scenario

User sees the red bar above Year column, and selects a row to delete. Wrangler infers a pattern.



Usage Scenario

User selects Year and Property_crime_rate columns. Wrangler suggests an unfold operation.



	Year	State	Property_crime_rate
0	2004	Alabama	4029.3
1	2005	Alabama	3900
2	2006	Alabama	3937
3	2007	Alabama	3974.9
4	2008	Alabama	4081.9
5	2004	Alaska	3370.9
6	2005	Alaska	3615
7	2006	Alaska	3582
8	2007	Alaska	3373.9
9	2008	Alaska	2928.3
10	2004	Arizona	5073.3
11	2005	Arizona	4827
12	2006	Arizona	4741.6
13	2007	Arizona	4502.6
14	2008	Arizona	4087.3

	State	2004	2005	2006	2007	
0	Alabama	4029.3	3900	3937	3974.9	408
1	Alaska	3370.9	3615	3582	3373.9	292
2	Arizona	5073.3	4827	4741.6	4502.6	408
3	Arkansas	4033.1	4068	4021.6	3945.5	384
4	California	3423.9	3321	3175.2	3032.6	294
5	Colorado	3918.5	4041	3441.8	2991.3	285
6	Connecticut	2684.9	2579	2575	2470.6	249
7	Delaware	3283.6	3118	3474.5	3427.1	359

Wrangler Transformation Language

- Declarative language supports 8 classes of transforms
- **Map** transforms one input row to zero, one, or multiple output rows
- **Lookups** incorporate data from external tables
  - Some lookup tables built in (e.g. zip codes to state names)
- **Reshape** enables folding and unfolding (pivot tables)
- **Positional** transforms allow for fill operations and shifting columns up/down
- Also includes **sorting**, **aggregation**, **key generation**, and **schema** transforms
  - Schema transforms set column names, data types, and semantic roles

Data Types and Semantic Roles

- Wrangler defines a type or role for each column
  - Inferred based on data in columns, but can be specified by user
- Wrangler automatically validates data to create a “data quality meter” for each column
  - Counts values that don’t match type/role (red bar)
  - Also counts missing values (gray bar)
- Includes standard data types (e.g. integers, strings) and also higher-level semantic roles (e.g. zip code)
- A set of common semantic roles is baked in
  - Can be extended

Interface Design

- Basic interactions
  - Users can select rows/columns, click data quality meter, select text within cells, edit data values, and assign column names/data types/semantic roles
- As user interacts with data, Wrangler generates suggested transforms
  - Wrangler generates natural language descriptions of transforms
- User can select transforms to “preview” effects on data in real time
- Wrangler stores transformation history, which can be exported as a script
- Data quality meter constantly validates data against inferred types

Interface Design

Transform Script Import Export

- Split **data repeatedly** on **newline** into **rows**
- Split **split repeatedly** on **','**
- Promote **row 0** to header
- Delete **empty rows**
- Extract from **Year** after **'in '**
- Set **extract's** name to **State**
- Fill **State** by **copying** values from **above**
- Delete **rows where Year starts with 'Reported'**

Text Columns Rows Table Clear

	Year	State	Property_crime_rate
0	2004	Alabama	4029.3
1	2005	Alabama	3900
2	2006	Alabama	3937
3	2007	Alabama	3974.9
4	2008	Alabama	4081.9
5	2004	Alaska	3370.9
6	2005	Alaska	3615
7	2006	Alaska	3582
8	2007	Alaska	3373.9
9	2008	Alaska	2928.3
10	2004	Arizona	5073.3
11	2005	Arizona	4827
12	2006	Arizona	4741.6
13	2007	Arizona	4502.6
14	2008	Arizona	4087.3

- Drop **Year, Property_crime_rate**
- Fold **Year, Property_crime_rate** using **header** as a key
- Fold **Year, Property_crime_rate** using **row 0** as a key
- Unfold **Year** on **Property_crime_rate**

	State	2004	2005	2006	2007	
0	Alabama	4029.3	3900	3937	3974.9	408
1	Alaska	3370.9	3615	3582	3373.9	292
2	Arizona	5073.3	4827	4741.6	4502.6	408
3	Arkansas	4033.1	4068	4021.6	3945.5	384
4	California	3423.9	3321	3175.2	3032.6	294
5	Colorado	3918.5	4041	3441.8	2991.3	285
6	Connecticut	2684.9	2579	2575	2470.6	249
7	Delaware	3283.6	3118	3474.5	3427.1	359

Inference Engine

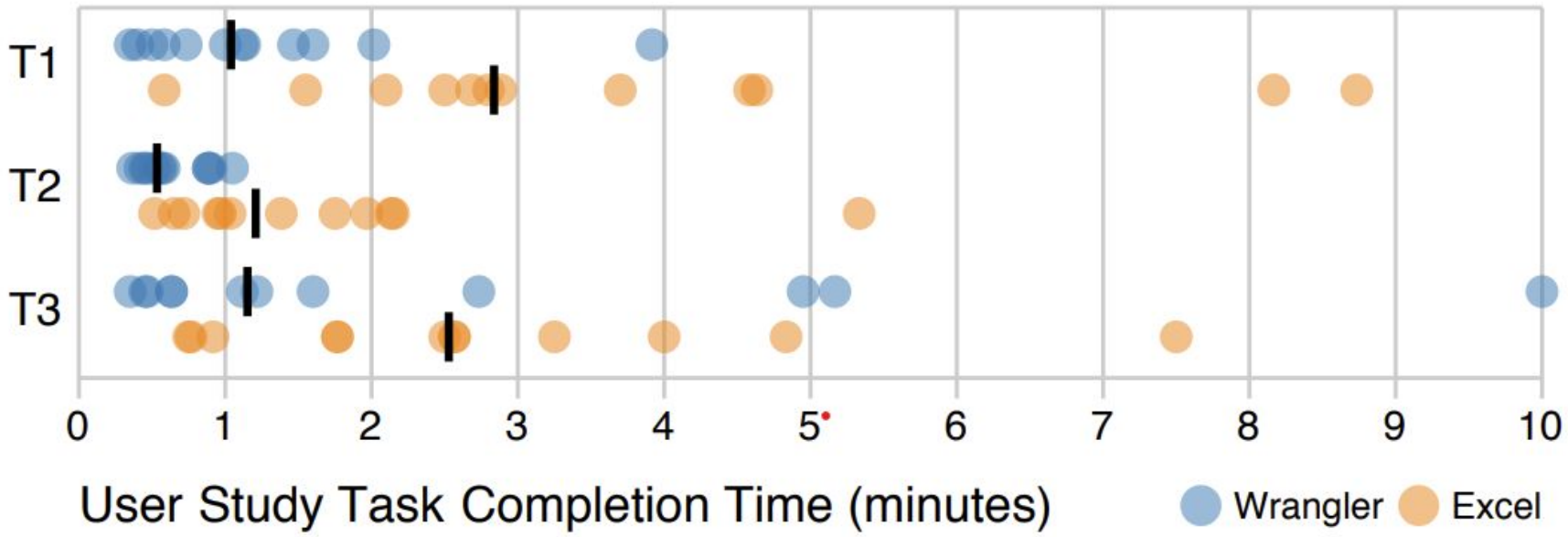
- Wrangler enumerates potential transformations using selected inputs, then ranks them based on a number of factors:
  - User input (user can select transform to use ahead of time)
  - Specification difficulty (row and text selections are “hard”, want to make these specifications less tedious)
  - Corpus frequency (frequency of transformations used by other users)
  - Complexity (prefer simple transforms)
  - Diversity (no one type accounts for more than ⅓ of suggestions)

Evaluation

- Ran a comparative evaluation with Excel
- Users assigned tasks and timed, filled out questionnaire
- 12 participants, analysts/students working with data

Evaluation Results

- Across all tasks, Wrangler over 2x as fast



Evaluation Results

- Users rated previews and suggestions more useful than direct editing
- While faster overall, users who got stuck worked slightly faster in Excel
  - Thanks to direct editing
- Some users got stuck by unknowingly filtering suggestions