中国R会
The China-R Conference

中国人民大學 | 统计学院
RENMIN UNIVERSITY OF CHINA | SCHOOL OF STATISTICS

中国人民大学应用统计科学研究中心
Center for Applied Statistics of Renmin University of China

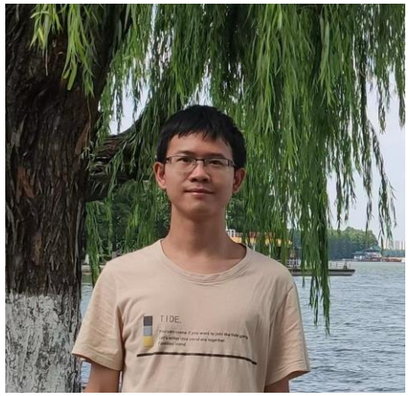
CAPITAL OF STATISTICS
PROFESSION, HUMANITY & INTEGRITY

Understanding the Semantics of Data Wrangling Scripts

Kai Xiong

2022.11.22

Introduce Myself



Kai Xiong (熊凯)

kaixiong@zju.edu.cn

ZJU·IDG

Zhejiang University

Tutor: Yingcai Wu (巫英才)

Homepage: xkkevin.github.io

Sept. 2020 - Present ● Hangzhou, China

Ph.D. Student

College of Computer Science and Technology, Zhejiang University

Apr. 2020 - Nov. 2022 ● Hangzhou, China

Research Intern

Jianwei Group, **Zhejiang Lab**

Sept. 2019 - Jun. 2020 ● Ningbo, China

Postgraduate Student

School of Software Technology, Zhejiang University

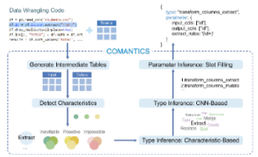
Aug. 2015 - Jun. 2019 ● Xi'an, China

Undergraduate Student

School of Computer Science and Technology, Xidian University

Research Interests:

- data wrangling and visual analytics
- how to assist data workers in understanding the process of data transformations



Revealing the Semantics of Data Wrangling Scripts With COMANTICS

Kai Xiong, Zhongsu Luo, Siwei Fu, Yongheng Wang, Mingliang Xu, Yingcai Wu

IEEE Transactions on Visualization and Computer Graphics (IEEE VIS), 2022

Visualizing the Semantics of Data Wrangling Scripts

Zhongsu Luo, **Kai Xiong**, Siwei Fu, Yongheng Wang, Hujun Bao, Yingcai Wu

Journal of Computer-Aided Design & Computer Graphics (ChinaVIS), 2022

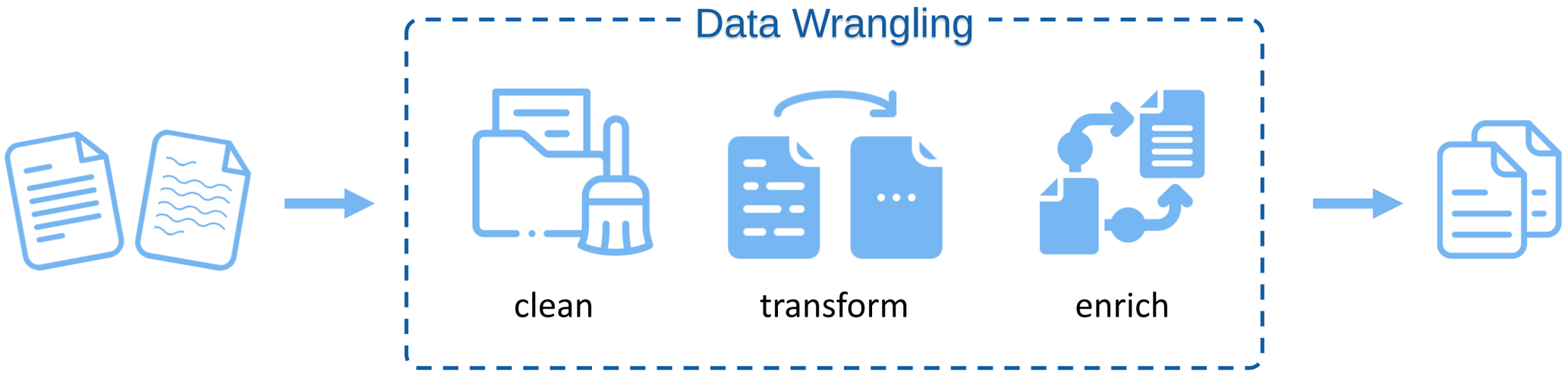
Visualizing the Scripts of Data Wrangling with SOMNUS

Kai Xiong, Siwei Fu, Guoming Ding, Zhongsu Luo, Rong Yu, Wei Chen, Hujun Bao, Yingcai Wu

IEEE Transactions on Visualization and Computer Graphics, 2022

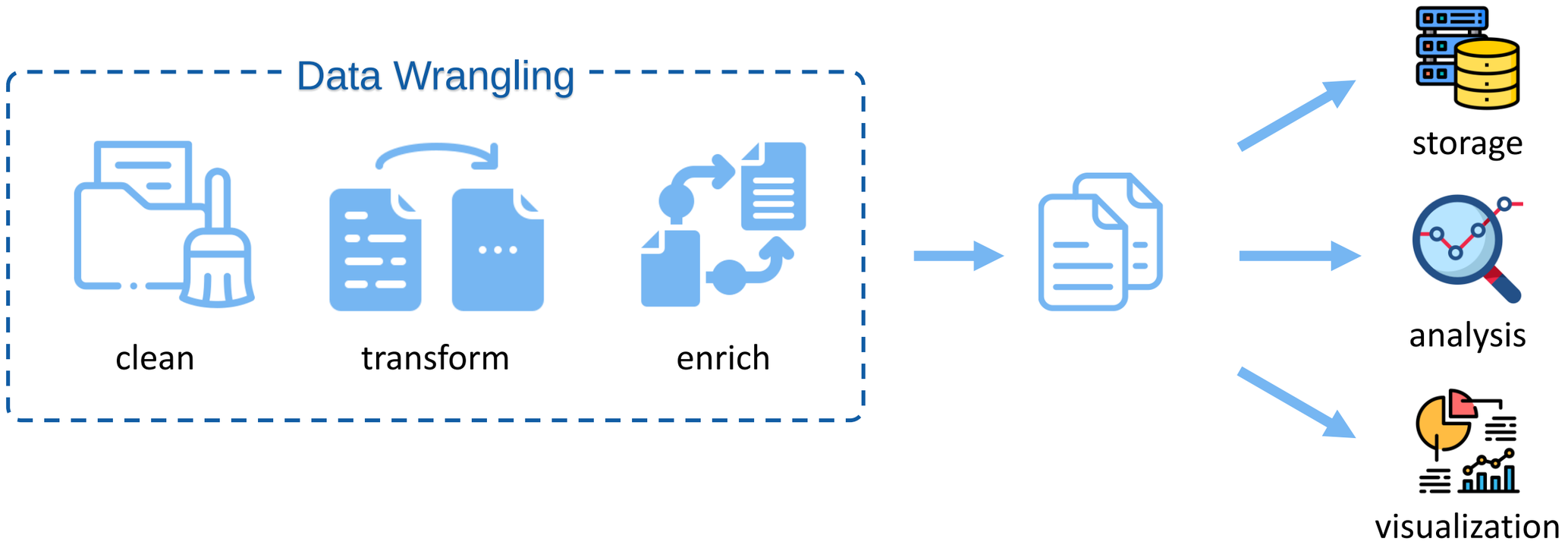
Data Wrangling

- Data wrangling is an arduous process of cleaning, transforming, and enriching messy and complex data into a desired format

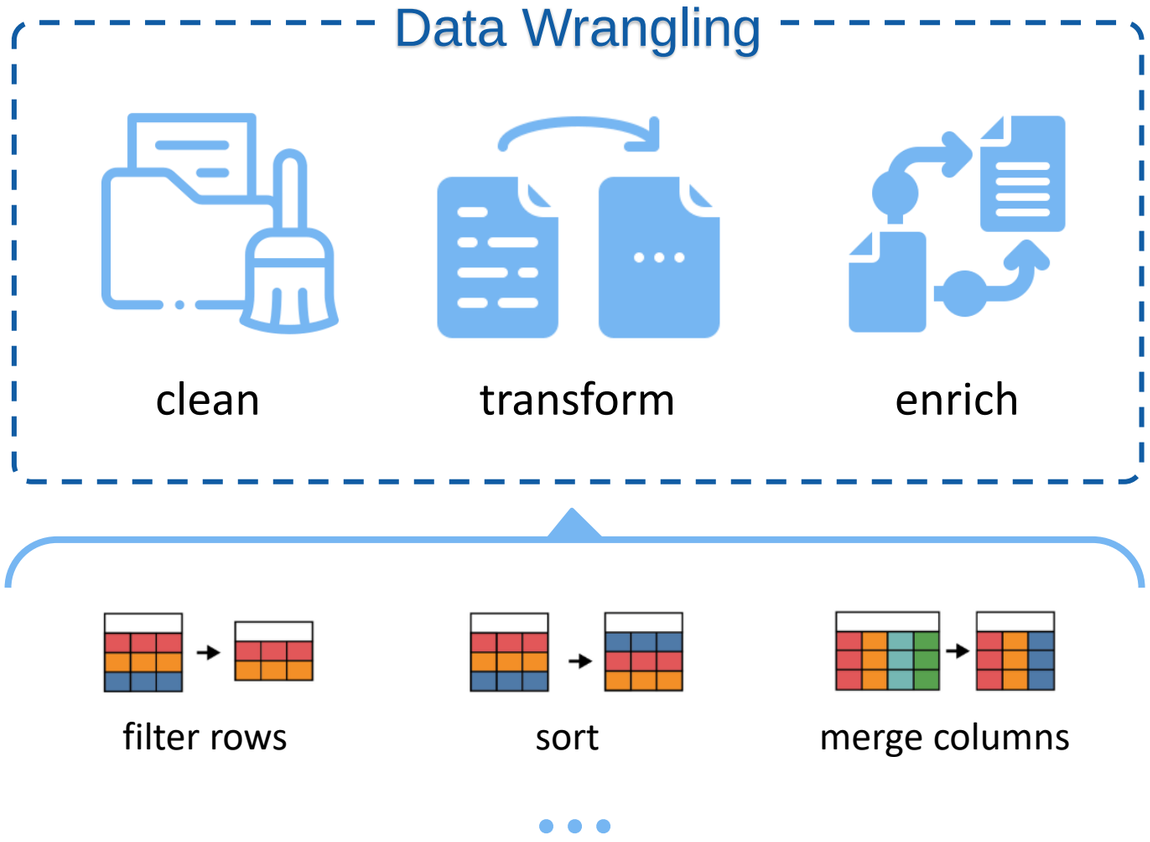


Data Wrangling

- Data wrangling is an arduous process of cleaning, transforming, and enriching messy and complex data into a desired format



Data Transformation



Kasica et al. 2021.

A multi-table framework for data wrangling

Operation categories

1. Create
2. Delete
3. Transform
4. Separate
5. Combine

×

Data object types

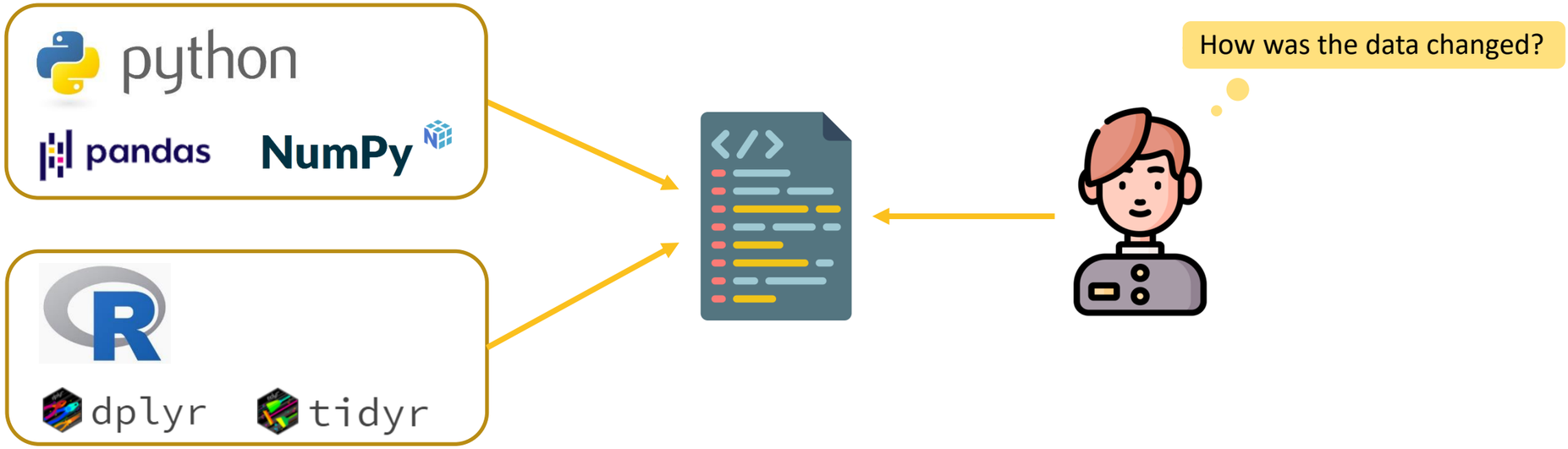
1. Tables
2. Columns
3. Rows

↓

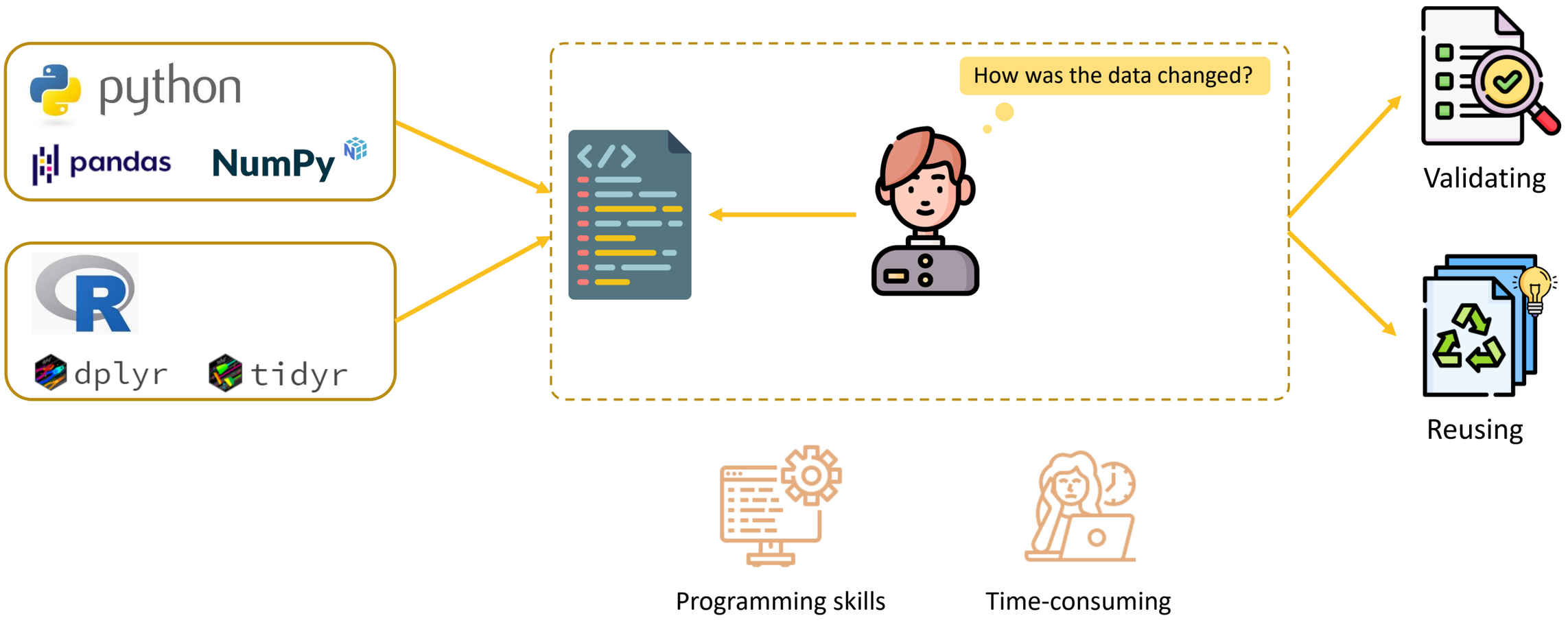
15 high-level transformations

S. Kasica, C. Berret, and T. Munzner. Table Scraps: An actionable framework for multi-table data wrangling from an artifact study of computational journalism. IEEE TVCG, 2021.

Understanding the Semantics of Scripts



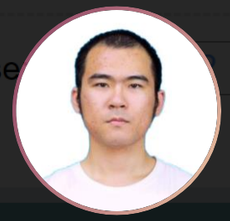
Understanding the Semantics of Scripts



INTERACTIVE DATA GROUP

Visualizing the Scripts of Data Wrangling with SOMNUS






Kai Xiong[1, 2] Siwei Fu[2] Guoming Ding[1, 2] Zhongsu Luo[3, 2]



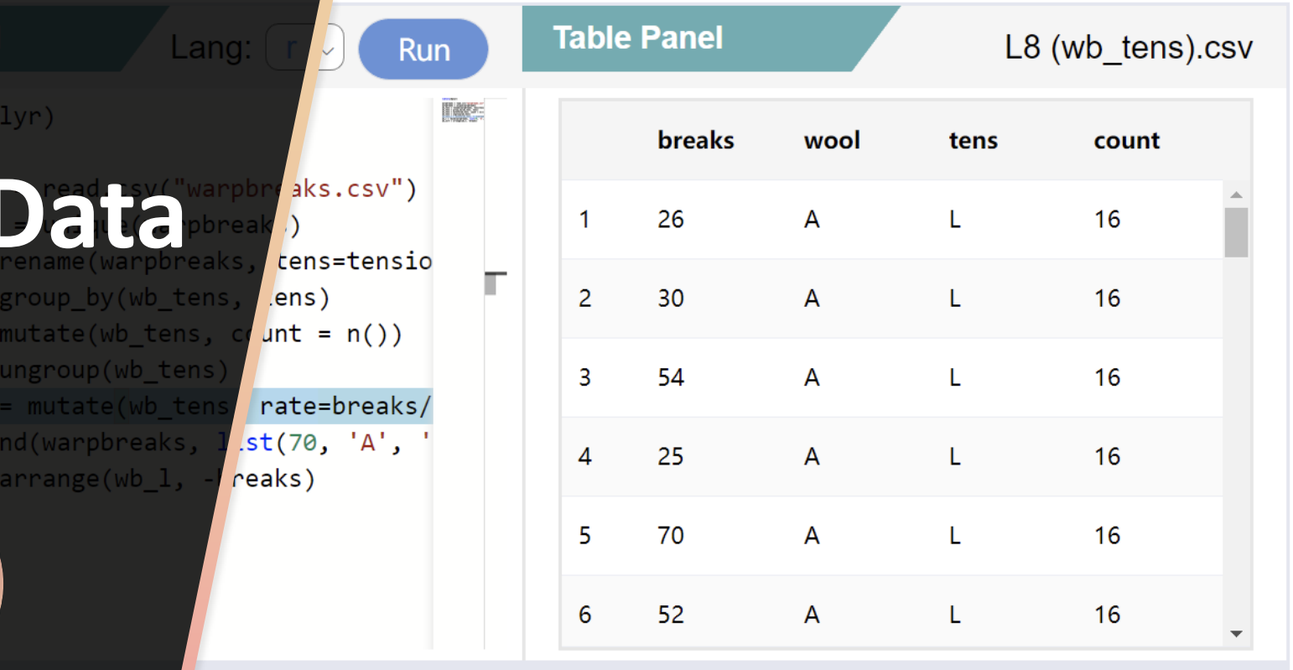

Rong Yu[2] Wei Chen[1, 2] Hujun Bao[1, 2] Yingcai Wu[1, 2]

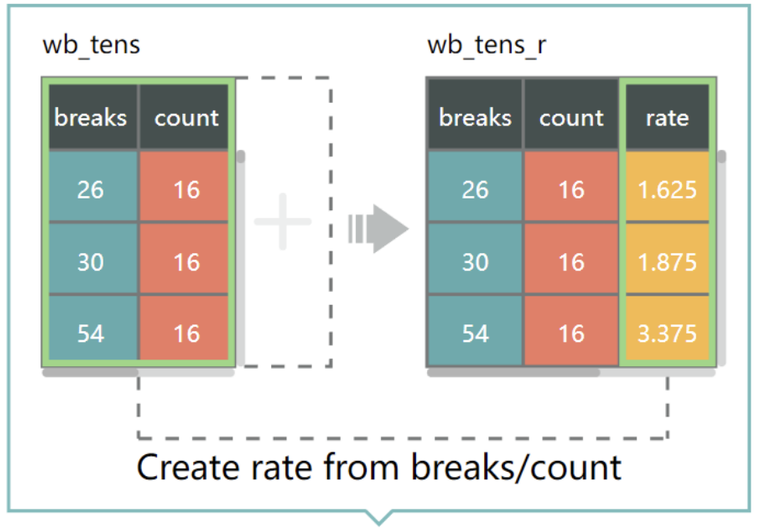
[1] State Key Lab of CAD&CG, Zhejiang University, Hangzhou, China

[2] Zhejiang Lab, Hangzhou, China

[3] Zhejiang University of Technology, Hangzhou, China

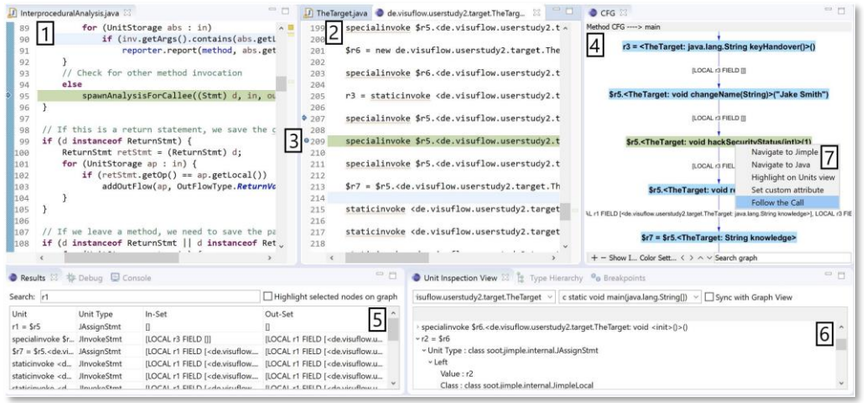
VIS 2022





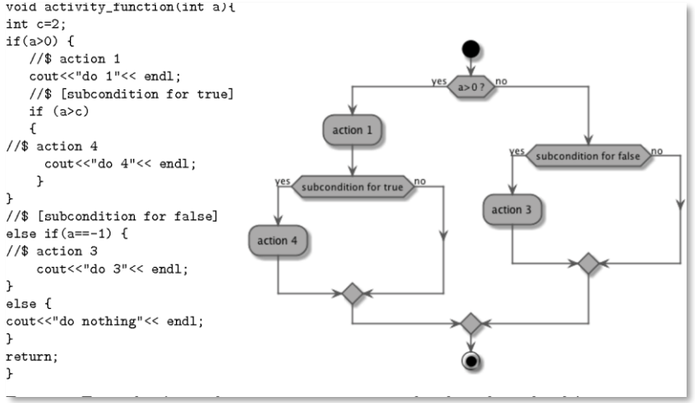
Program Visualization

Nguyen et al. 2018



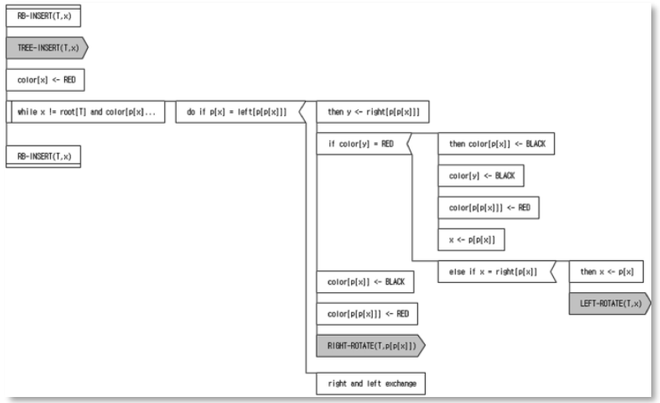
VisuFlow

Kosower et al. 2014



Flowgen

Cheon et al. 2019



VizMe

Not in the context of **data transformation**

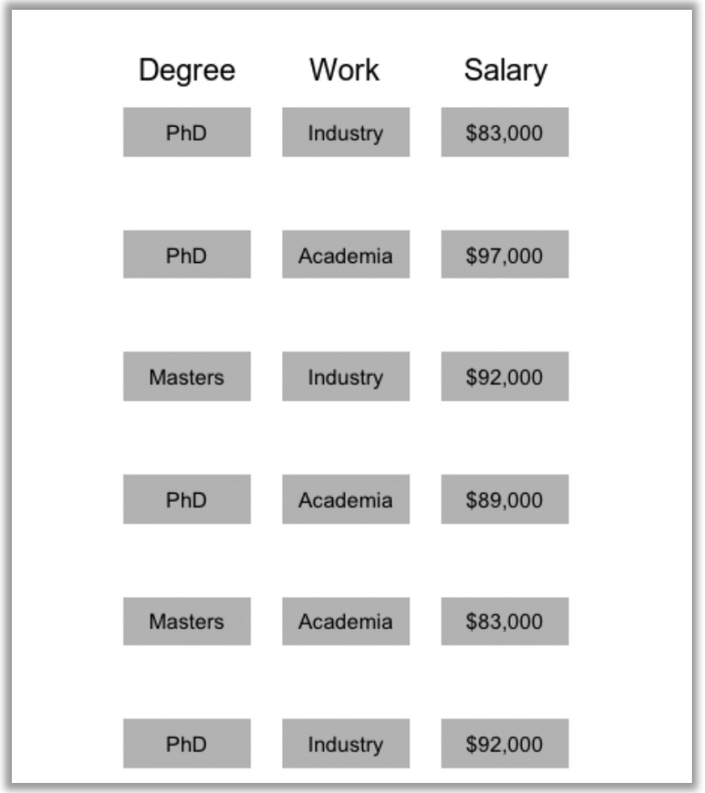
Data Provenance

Khan et al. 2017

Student ID	Last Name	Year of Enrollment	Course	GPA
1	Johnson	2012	CSE	3.47
4	Hayes	2012	CSE	3.82
2	Turner	2012	ECE	3.34
3	Jones	2012	ECE	3.54
5	Lindsay	2012	ECE	2.99
6	Smith	2013	CSE	3.12
10	Frick	2013	CSE	3.53
11	Jones	2013	CSE	3.78
12	Zelaya	2013	CSE	3.94
7	Wallace	2013	ECE	4.00
8	Jabbar	2013	ECE	3.65
9	Borg	2013	ECE	3.32

Data Tweening

Pu et al. 2021



Datamations

hard to **explore** and only for **a single table**

Somnus

Data Panel

Source

- input1.csv
- input2.csv

Upload

Case: r_case1

Script Panel

Language: r

Run

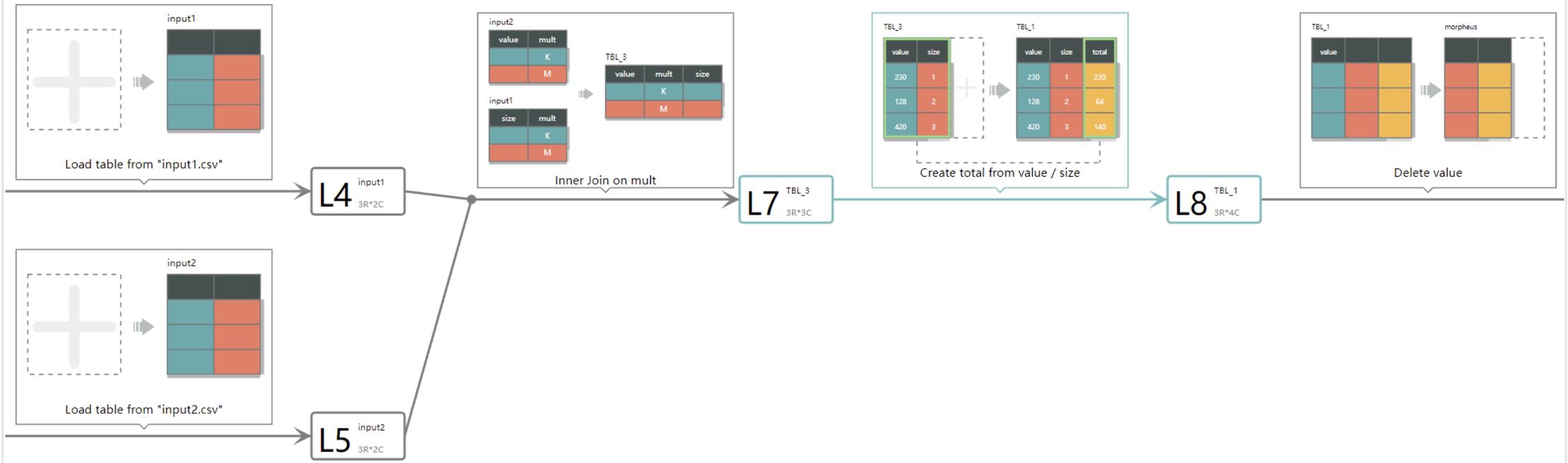
```r
library(tidyr)
library(dplyr)

input1 = read.csv("input1.csv")
input2 = read.csv("input2.csv")

TBL_3=inner_join(input2, input1)
TBL_1=mutate(TBL_3,total=value / size)
morpheus=select(TBL_1,-`value`)
morpheus=as.data.frame(morpheus)
morpheus=select(morpheus,2,1,3)
morpheus
```

Table Panel

L8 (TBL_1).csv

	value	mult	size	total
1	230	K	1	230
2	128	M	2	64
3	420	G	3	140

Graph Panel



SOMNUS Overview

Data Tables

Data Wrangling Script

```
… …
s_sum = s_sum.reset_index(drop=True)
r_sum = r_sum.reset_index(drop=True)

rev_sales = pd.concat([s_sum, r_sum])
rev_sales.loc[len(rev_sales)] = rev_sales.mean()
… …
```

Input

Provenance Graph

Output

SOMNUS Overview

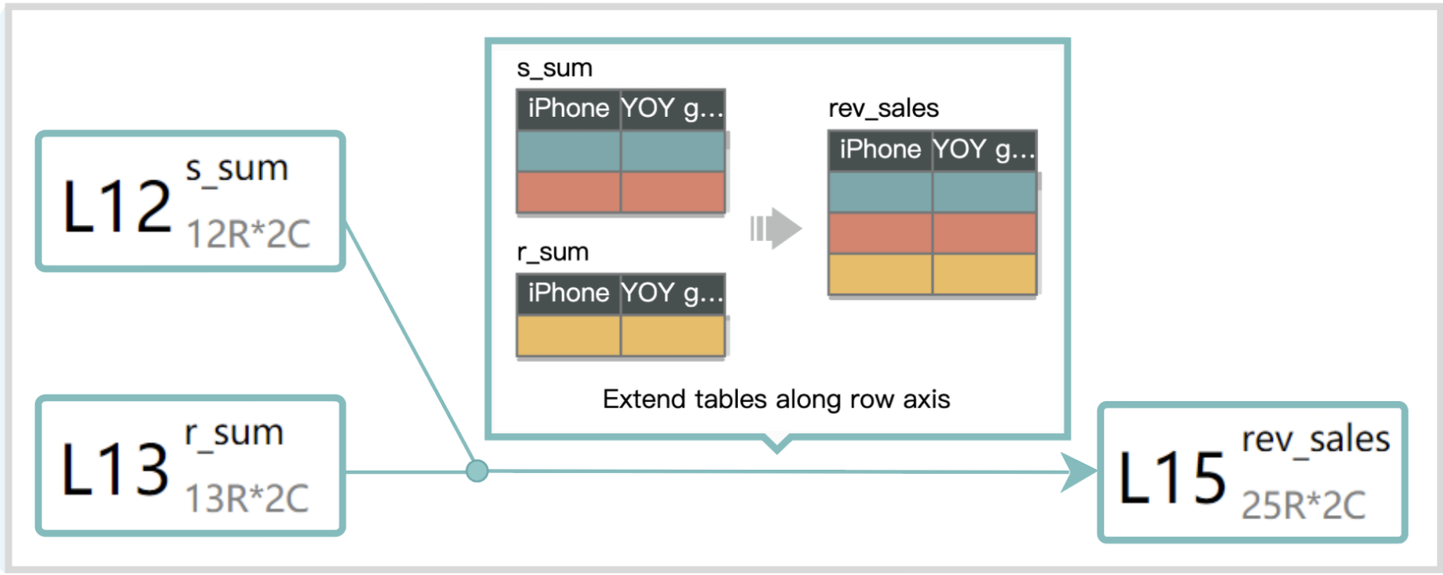
Data Tables

Data Wrangling Script

```
… …
s_sum = s_sum.reset_index(drop=True)
r_sum = r_sum.reset_index(drop=True)

rev_sales = pd.concat([s_sum, r_sum])
rev_sales.loc[len(rev_sales)] = rev_sales.mean()
… …
```

Input

Provenance Graph



Output

Design Requirement

R1: Present the Semantics

R2: Link with Data

R3: Depict Necessary Information

R4: Keep Encoding Consistent

R5: Reveal Table Provenance

R6: Dig into Details

R7: Independent of Programming Languages

Design of Glyphs

Design space of glyphs

depict

Kasica et al. 2021

Op Class	Sets	Expanded Operations
Create	0:1	T/C/R: `Create`
Delete	1:0	T/C/R: `Delete`
Transform	1:1	T: `Rearrange`, `Reshape`; C/R: `Transform`
Separate	1:N	T: `Subset`, `Decompose`, `Split`; C/R: `Separate`
Combine	N:1	T: `Extend`, `Supplement`, `Match`; C: `Combine`; R: `Summarize`, `Interpolate`

Data transformations

Parameter Space

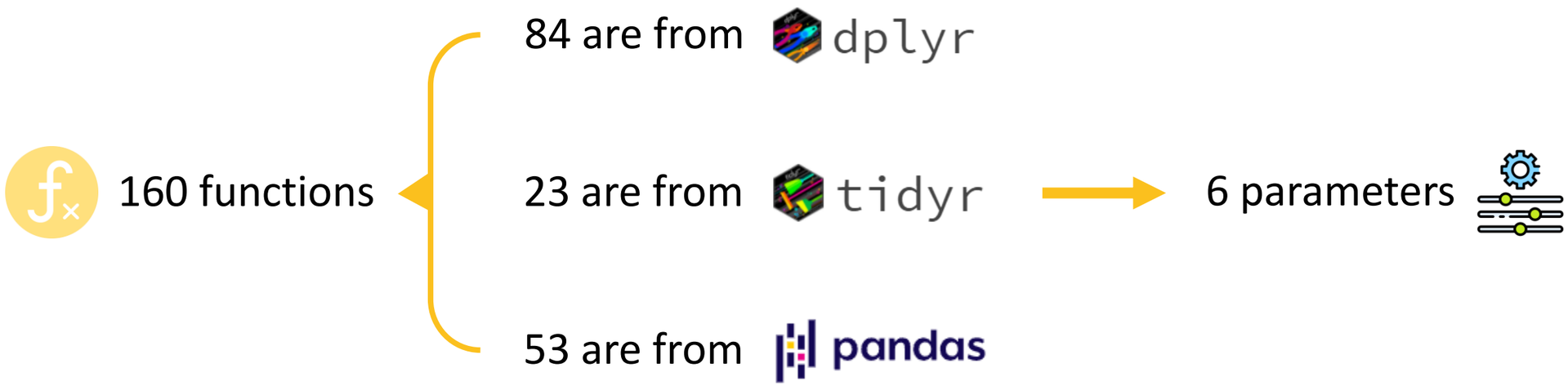
which information shall be encoded

Visual Channels

which visual channels can be mapped

Design of Glyphs

Parameter Space → Visual Channels



Design of Glyphs

Parameter Space → Visual Channels

5 Contextual Columns/Rows

col1	col2	price
a	1	18$
b	1	25$
c	2	15$

store

col1	col2	price
a	1	18
b	1	25
c	2	15

store_extract

1 Function Name

4 Transformation Parameters

```
store_extract = extract(store, price, “price”, “(.)$”) # transform price
```

2 Data Tables

3 Explicit Columns/Rows

col1	col2	col3
A	1	m
B	2	t
A	1	m

tbl3

```
tbl4 = unique(tbl3) # delete duplicate rows
```

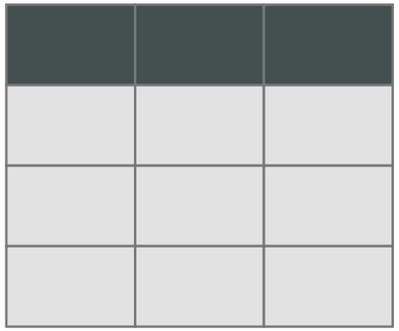
2 1 2

6 Implicit Columns/Rows

Design of Glyphs

Parameter Space → Visual Channels

```
store_extract = extract(store, price, “price”, “(.)$”) # transform price
```



Design of Glyphs

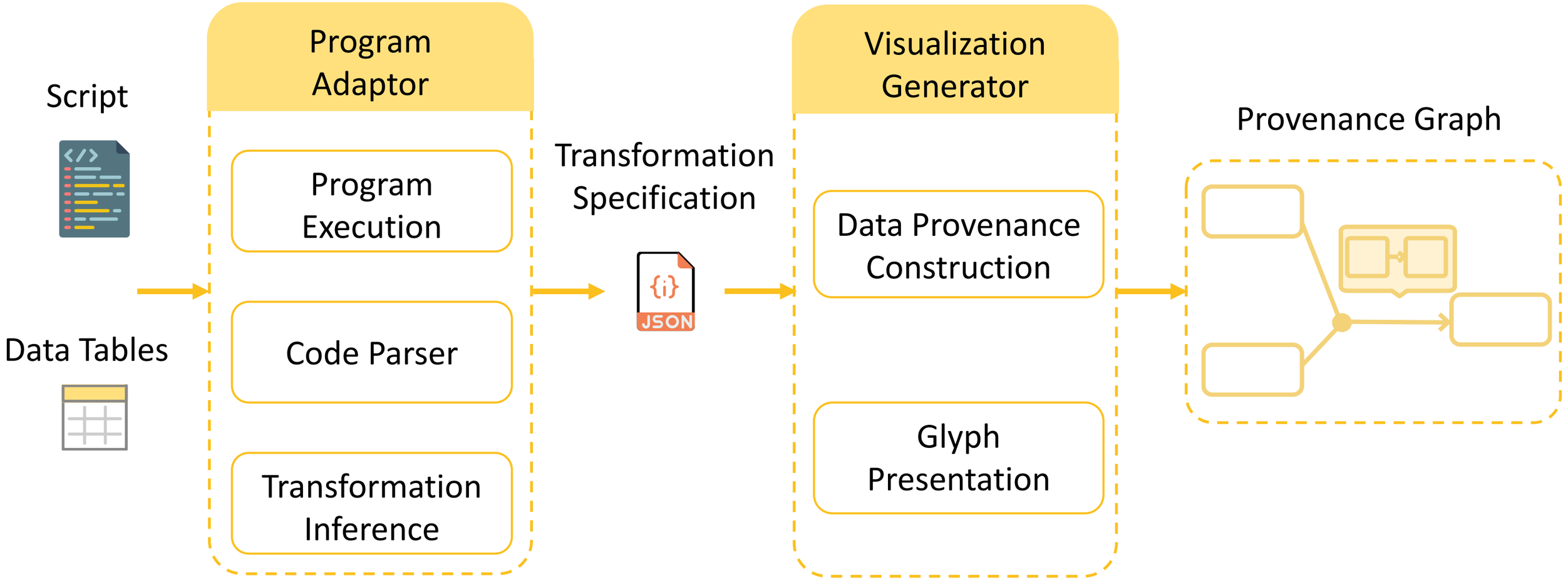
Parameter Space → Visual Channels

```
store_extract = extract(store, price, “price”, “(.)$”) # transform price
```

Please refer to our paper for the detailed design rationale.

Design of Glyphs

SOMNUS Architecture



SOMNUS

Program Adaptor → Visualization Generator

Program Execution

Code Parser

Transformation Inference

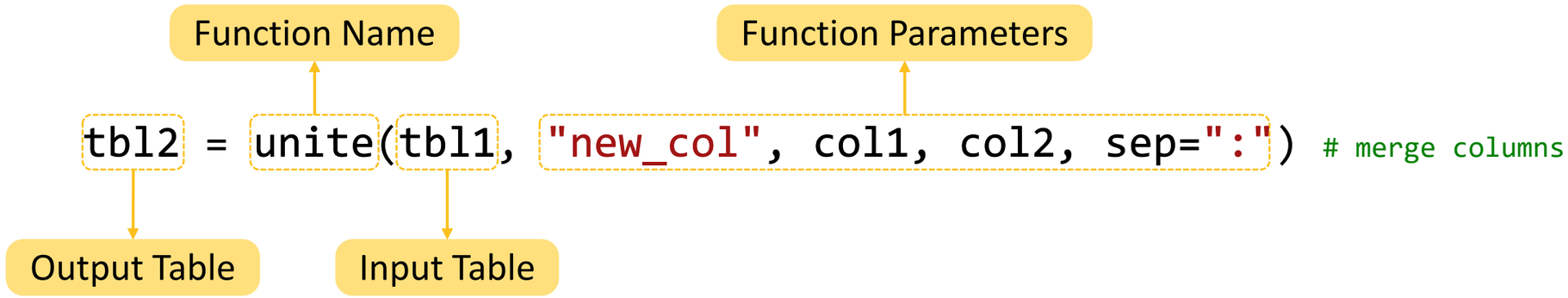
Line	Input Table	Output Table
1		
2		
...		

SOMNUS

Program Adaptor → Visualization Generator

Program Execution

Code Parser



Transformation Inference

SOMNUS

Program Adaptor → Visualization Generator

Program Execution

Code Parser

Transformation Inference

Input Table: tbl1

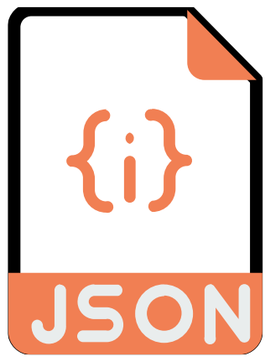
Output Table: tbl2

Function Name: `unite`

Function Parameters:

```
"new_col", col1, col2, sep=":"
```

Transformation Type



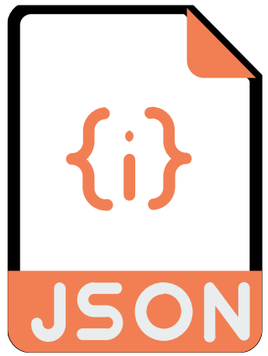
SOMNUS

Program Adaptor → Visualization Generator

Program Execution

Code Parser

Transformation Inference




```
transformation specifications:
▼ (10) [{…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}]
  ▶ 0: {operation_rule: 'Load table from "apple-iphone-revenue.csv"', output_t
  ▶ 1: {operation_rule: 'Load table from "apple-iphone-unit-sales.csv"', outpu
  ▶ 2: {input_explicit_col: Array(1), input_table_file: 'L4 (sales).csv', inpu
  ▶ 3: {input_explicit_col: Array(1), input_table_file: 'L3 (revenue).csv', in
  ▼ 4:
    ▶ input_explicit_col: (2) ['YOY growth', 'iPhone']
    ▶ input_implicit_col: ['year']
      input_table_file: "L6 (sales).csv"
      input_table_name: "sales"
      operation_rule: "Group by year, Summarize columns by sum"
    ▶ output_explicit_col: (2) ['YOY growth', 'iPhone']
      output_table_file: "L9 (s_sum).csv"
      output_table_name: "s_sum"
      type: "combine_rows_summarize"
    ▶ [[Prototype]]: Object
  ▶ 5: {input_explicit_col: Array(2), input_implicit_col: Array(1), input_tabl
  ▶ 6: {input_explicit_col: Array(1), input_table_file: 'L9 (s_sum).csv', inpu
  ▶ 7: {input_explicit_col: Array(1), input_table_file: 'L10 (r_sum).csv', inp
  ▶ 8: {input_table_file: Array(2), input_table_name: Array(2), operation_rule
  ▶ 9: {input_table_file: 'L15 (rev_sales).csv', input_table_name: 'rev_sales'
    length: 10
```

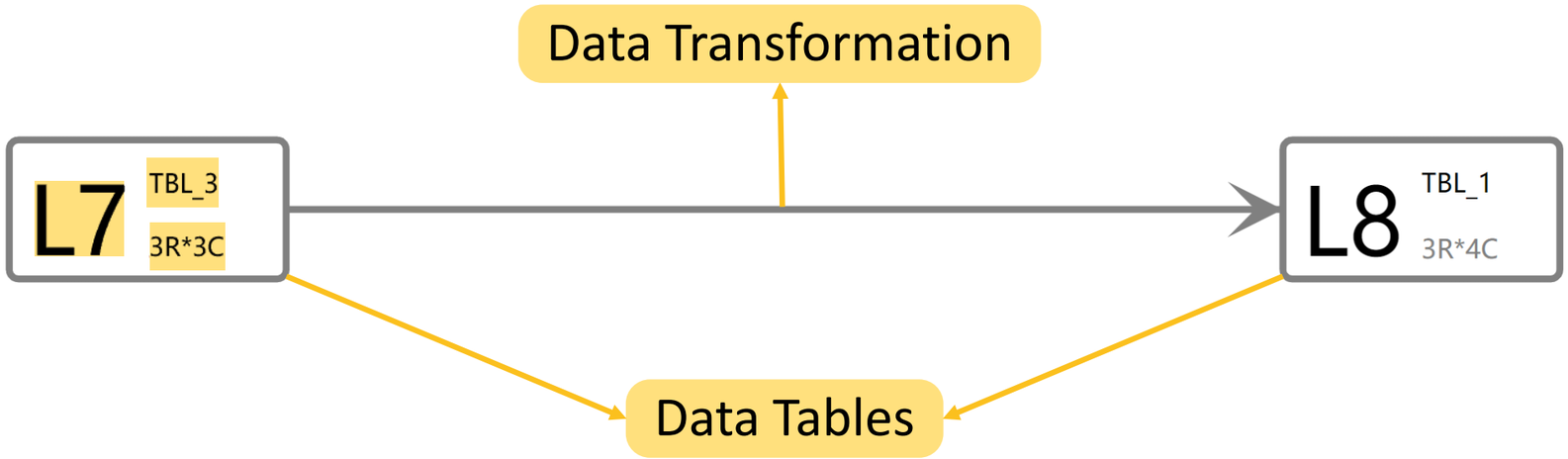
SOMNUS

Program Adaptor

Visualization Generator

Data Provenance Construction

Glyph Presentation



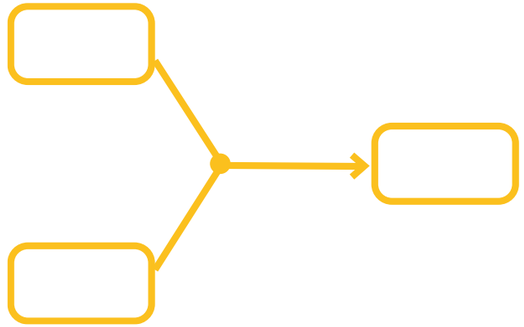
SOMNUS

Program Adaptor → Visualization Generator

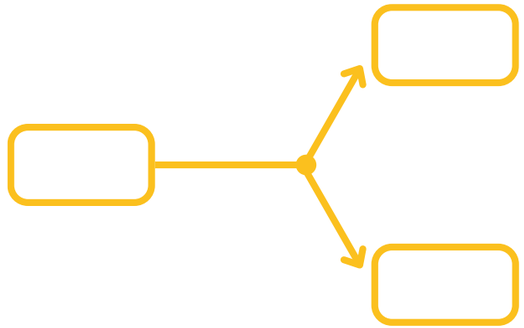
Data Provenance Construction

Glyph Presentation

Convergence edges:
(many-to-one)



Divergence edges:
(one-to-many)



Directed edges:
(one-to-one, create/delete tables)

SOMNUS

Program Adaptor → Visualization Generator

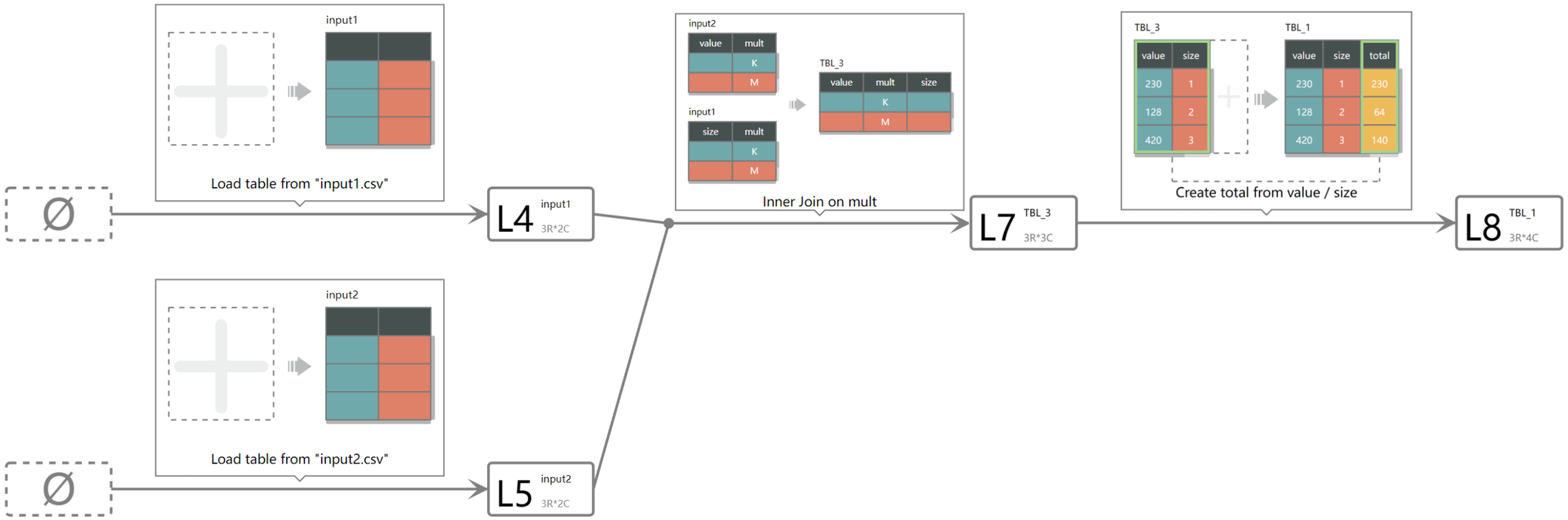
Data Provenance Construction

Glyph Presentation

SOMNUS

Program Adaptor

Visualization Generator



User Study

- Q1: Does the **glyph design** improve user efficiency in comprehending **the semantics of data transformation**?

5 **function** understanding tasks (*Func*)

- Q2: Does the **provenance graph** facilitate the understanding of **data dependencies**?

5 **script** understanding tasks (*Script*)

User Study

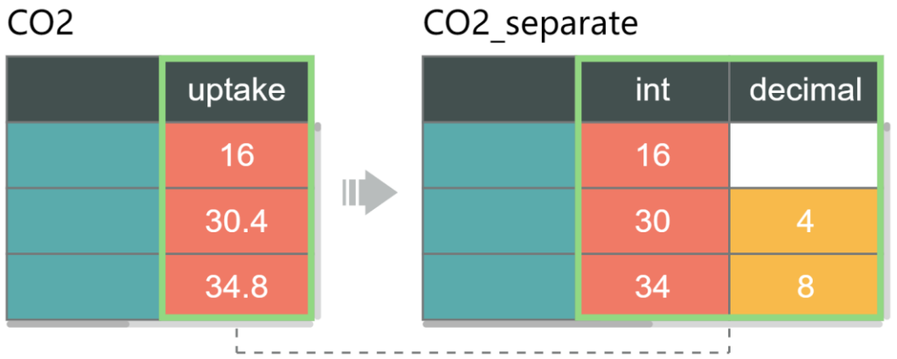
- Techniques:

Code:
```
CO2_separate = separate(CO2, uptake, into=c("int", "decimal"), sep="[.]")
```

Text: Split uptake on delimiters matching '.' into "int" and "decimal" columns

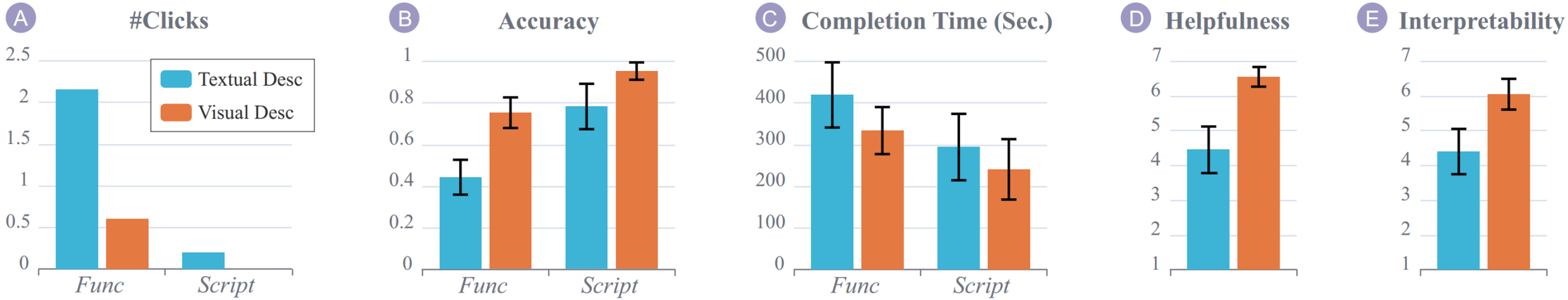
TRIFACTA

Visualization:



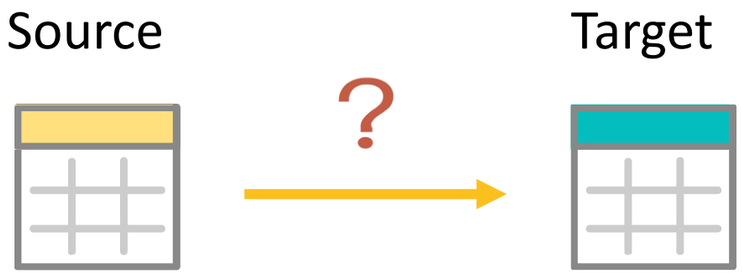
User Study

- Participants:
  - 20 subjects (4 females, 16 males, age: 22-35)
- Results:



Example Applications

1. Double-checking:
   - help data scientists validate the procedure of data transformation
2. MORPHEUS Revisited:
   - reveal intermediate data transformations given source and target tables



Y. Feng, R. Martins, J. Van Geffen, I. Dillig, and S. Chaudhuri. Component-based synthesis of table consolidation and transformation tasks from examples. ACM SIGPLAN Notices, 2017.

Somnus

Data Panel

Source

Upload

Case: r_case1

Script Panel

Language: r Run

```r
library(tidyr)
library(dplyr)

input1 = read.csv("input1.csv")
input2 = read.csv("input2.csv")

TBL_3=inner_join(input2, input1)
TBL_1=mutate(TBL_3,total=value / size)
morpheus=select(TBL_1,-`value`)
morpheus=as.data.frame(morpheus)
morpheus=select(morpheus,2,1,3)
morpheus
```

Table Panel

input1.csv

	size	mult
1	1	K
2	2	M
3	3	G

Graph Panel

Conclusion

Contributions

- a design space consisting of two dimensions that guide the design of a collection of 23 glyphs.
- a pipeline, SOMNUS, that visualizes the creation and evolution of data tables across a series of transformations.
- a controlled study that evaluates how users perform with visualization and text using comparison tasks.
- two example applications that showcase how SOMNUS can benefit different usage scenarios.

Future Work

- enhance SOMNUS by supporting a large number of functions and parameters.
- explore how to visualize conditional statements and loops in the provenance graph.

INTERACTIVE DATA GROUP

Revealing the Semantics of Data Wrangling Scripts With COMANTICS


Kai Xiong[1, 2]


Zhongsu Luo[3, 2]


Siwei Fu[2]

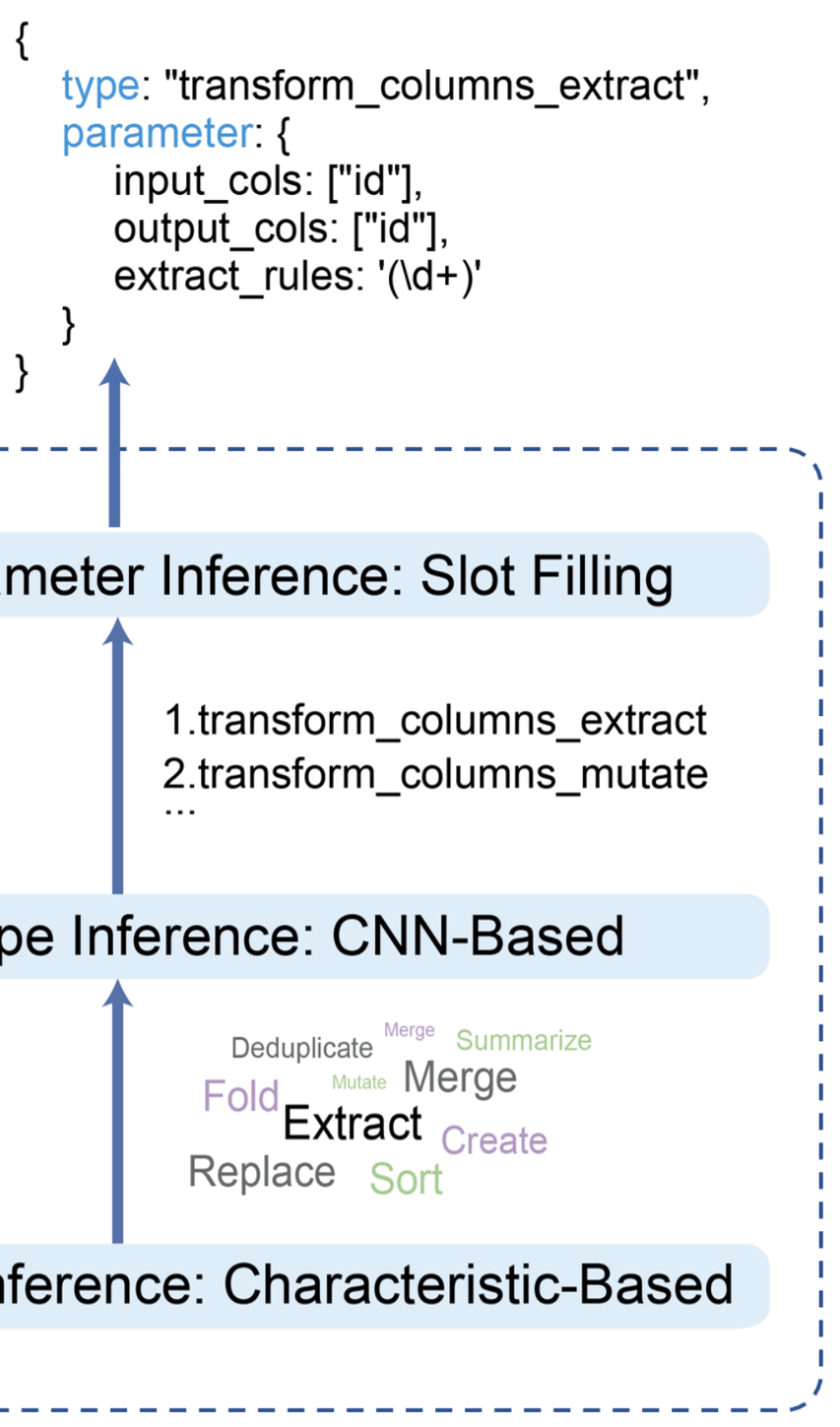

Yongheng Wang[2]


Mingliang Xu[4]

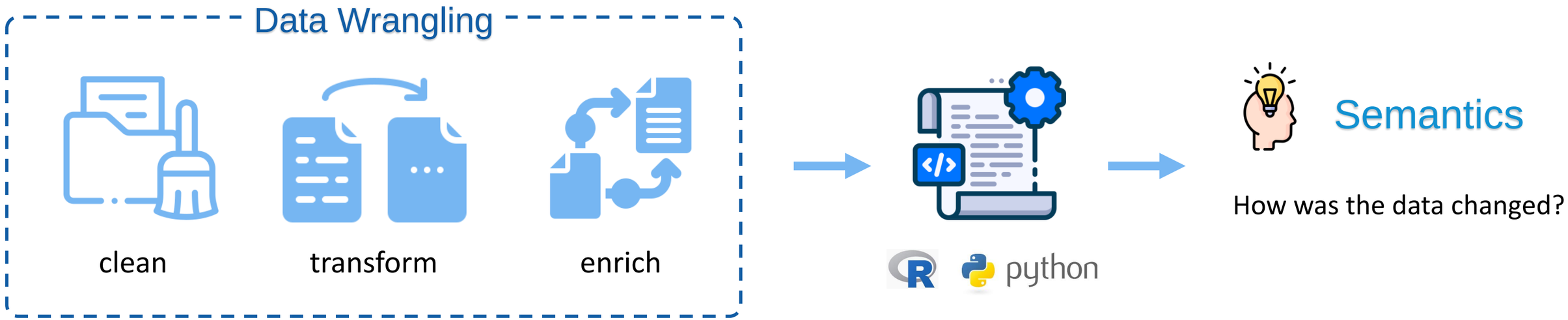

Yingcai Wu[1, 2]

[1] State Key Lab of CAD&CG, Zhejiang University, Hangzhou, China
[2] Zhejiang Lab, Hangzhou, China
[3] Zhejiang University of Technology, Hangzhou, China
[4] School of Computer and Artificial Intelligence, Zhengzhou University, Zhengzhou, China

VIS 2022



Semantics of Scripts



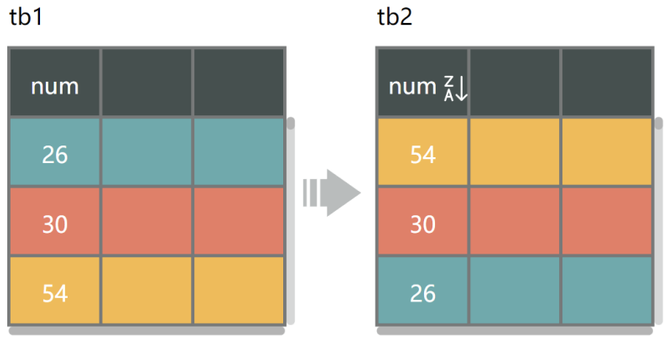
Semantics of Scripts

- The **semantics** of wrangling code is the **type of data transformation** and its **parameters**.

```
tb2 = arrange(tb1, -num)
```

Semantics

How was the data changed?

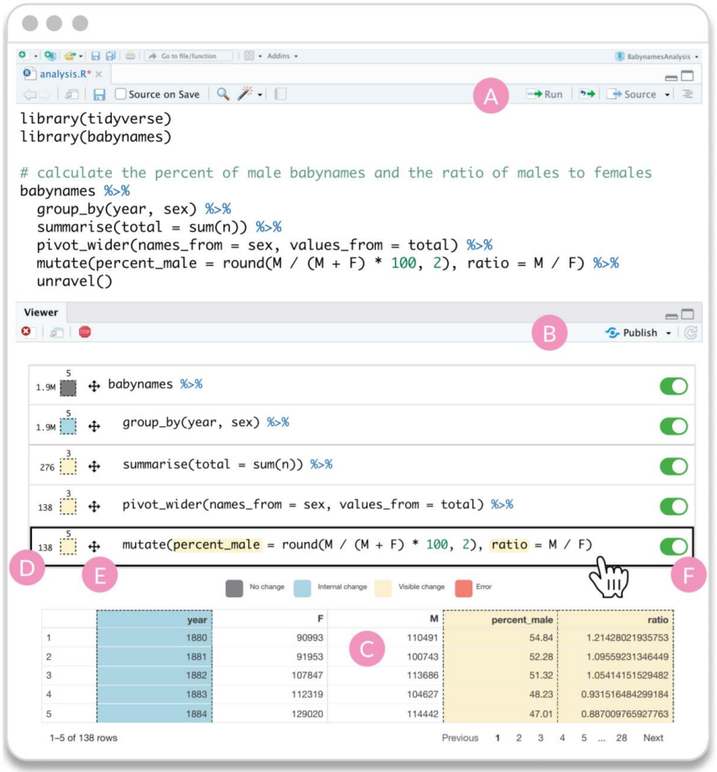


Transformation type: sort

Parameters: {
Column: num
Input Table: tb1
Order: descending
Output Table: tb2
}

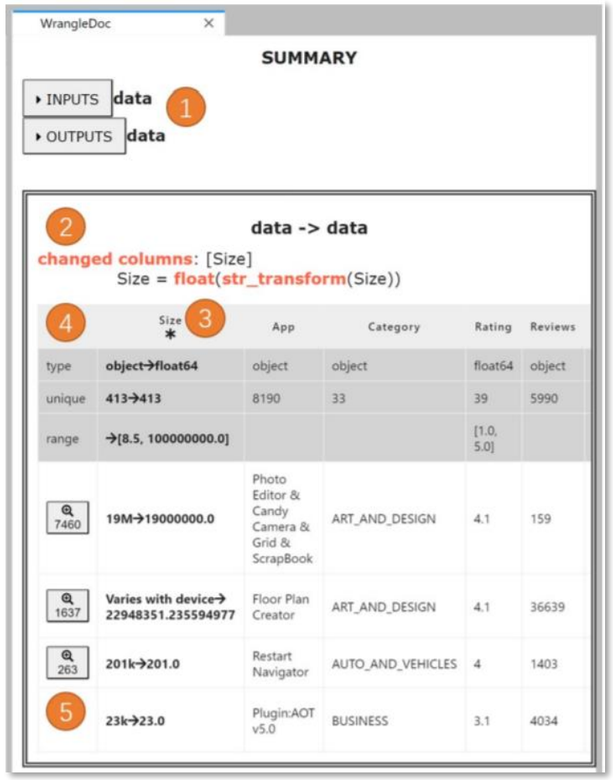
Program Understanding

Shrestha et al. 2021



Unravel

Yang et al. 2021



WrangleDoc

Cannot support inferring **data transformations**

Program Understanding

Khan et al. 2017

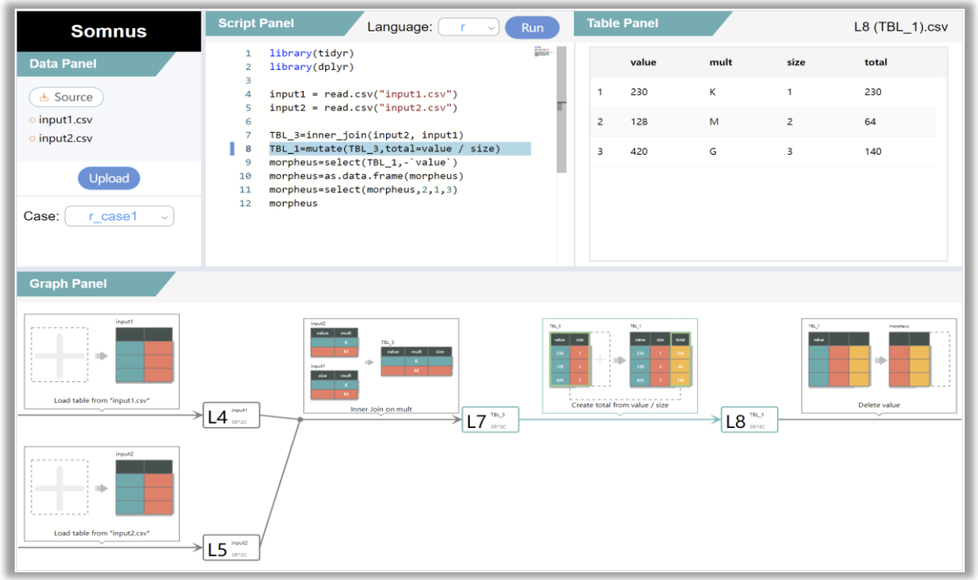
Student ID	Last Name	Year of Enrollment	Course	GPA
1	Johnson	2012	CSE	3.47
4	Hayes	2012	CSE	3.82
2	Turner	2012	ECE	3.34
3	Jones	2012	ECE	3.54
5	Lindsay	2012	ECE	2.99
6	Smith	2013	CSE	3.12
10	Frick	2013	CSE	3.53
11	Jones	2013	CSE	3.78
12	Zelaya	2013	CSE	3.94
7	Wallace	2013	ECE	4.00
8	Jabbar	2013	ECE	3.65
9	Borg	2013	ECE	3.32

Data Tweening

Pu et al. 2021

Datamations

Xiong et al. 2022



SOMNUS

Rely on hand-crafted rules, limited in **generalizability and scalability**

COMANTICS

Data Wrangling Script

```
1  import pandas as pd
2
3  df = pd.read_csv("students.csv")
4
5  df.id = df.id.str.extract("(\d+)")
6  df.drop_duplicates(inplace=True)
7  df.loc[:, "total"] = df.math + df.art
8  results = df.sort_values("total", ascending = False)
   ……
```

Semantics

```
{
   type: "transform_columns_extract",
   parameter: {
       input_cols: ["id"],
       output_cols: ["id"],
       extract_rules: '(\d+)'
   }
}
```

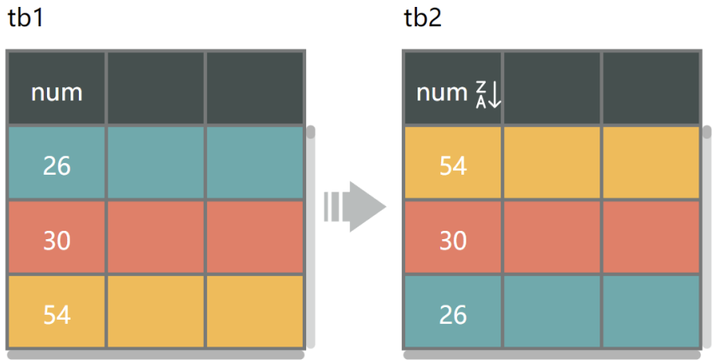
Challenges

Scripting Language Level → Function Level → Parameter Level

Challenges

Scripting Language Level | Function Level | Parameter Level

- Different scripting languages have diverse regulations in implementing data transformations



Sort

dplyr

```
tb2 = arrange(tb1, -num)
tb2 = arrange(tb1, desc(num))
```

python pandas

```
tb2 = tb1.sort_values('num', ascending=False)
tb2 = tb1.sort_values(by=['num'], ascending=False)
```

Challenges

Scripting Language Level | **Function Level** | Parameter Level

- There are different functions to implement the same data transformation

tb2

O	P	Q
10.2	5.5	4.5
10.4	5.7	3.2

tb1

K	V
O	10.2
O	10.4
P	5.5
P	5.7
Q	4.5
Q	3.2

Fold

tidyr

```
tb2 = gather(tb1, K, V, -`ID`, -`T`)
tb2 = pivot_longer(tb1, O:Q, names_to='K', values_to='V')
```

Challenges

Scripting Language Level | Function Level | **Parameter Level**

- Functions may omit important parameters for simplicity

store

Category
Q2/07
Q3/08
Q4/09

revenue

Category	Quarter	Year
Q2/07	Q2	07
Q3/07	Q3	08
Q4/07	Q4	09

Split Category on '/'

Split

tidyr

```
revenue = separate(store, Category, c("Quarter", "Year"), sep="/")
```

Challenges

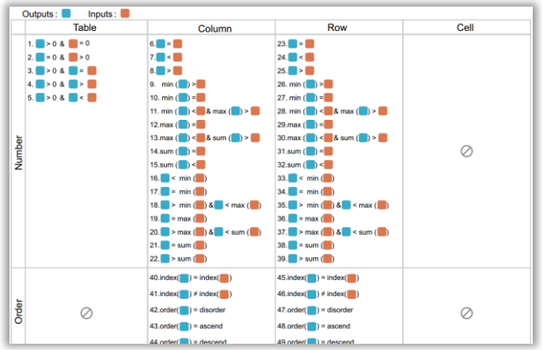
Scripting Language Level → **Function Level** → **Parameter Level**

- Different scripting languages have diverse regulations in implementing data transformations
- There are different functions to implement the same data transformation
- Functions may omit important parameters for simplicity

Our Solution

Qualitative research

A space of table changes



Infer transformation candidates

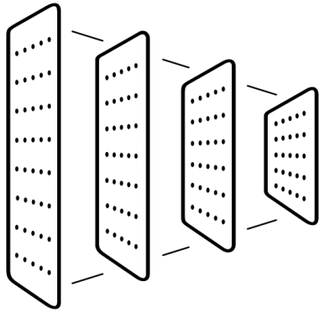
Gender	Age
Male	23
Female	
Male	31

Gender	Age
Male	23
Male	31

1. Keep Rows Where Gender is Male
2. Remove Rows with Missing Values

Machine learning

A CNN model



Resolve ambiguities

1. Keep Rows Where Gender is Male ✗
2. Remove Rows with Missing Values ✓

✗

Observation

- Differences between input and output tables can reveal the transformation type

Input

Gender	Age
Male	23
Female	
Male	31

Output

Gender	Age
Male	23
Male	31

High-level transformation

Create Columns

Create Rows

Delete Rows

Combine Columns

Low-level operation

1. **Filter:** Keep Rows Where Gender is Male
2. **DropNa:** Remove Rows with Missing Values

Observation

- Differences between input and output tables can reveal the transformation type

Input

Gender	Age
Male	23
Female	
Male	31
	26

Output

Gender	Age
Male	23
Male	31

High-level transformation

Create Columns

Create Rows

Delete Rows

Combine Columns

Low-level operation

1 **Filter:**
Keep Rows Where Gender is Male

2 **DropNa:**
Remove Rows with Missing Values

Space of Table Changes

- For different transformations, what changes can be made to tables?

python R

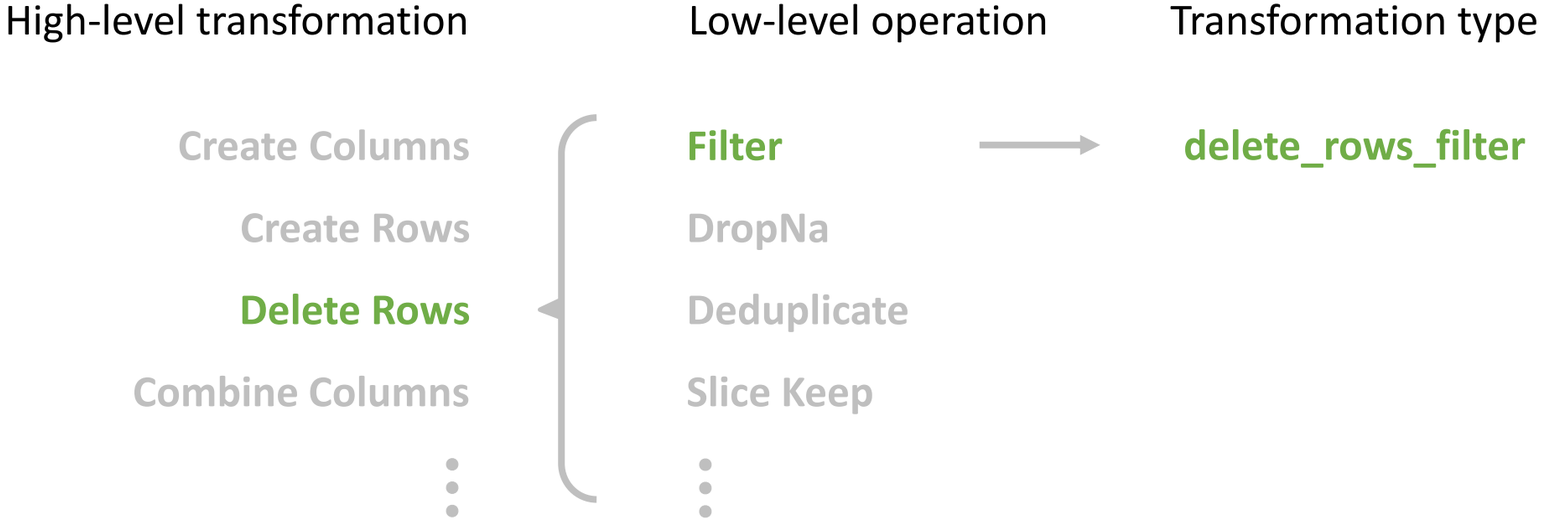
921 lines of wrangling code

Inst	Language	Case	Code	Input Table	Output Table	Transformation Type
1	Python	1	raw_full = pd.read_csv('full-downloaded.csv',index_col=None		L3 (raw_full).csv	create_tables_load
2	Python	1	raw_partial = pd.read_csv('partials-downloaded.csv',index_cc		L4 (raw_partial).csv	create_tables_load
3	Python	1	data_raw = pd.concat([raw_full,raw_partial])	L3 (raw_full).csv|L4 (raw_partial).c	L5 (data_raw).csv	combine_tables_extend
4	Python	1	data_typed = data_raw.assign(DemoType = data_raw.apply(	L5 (data_raw).csv	L8 (data_typed).csv	create_columns_mutate
5	Python	1	permit_status_filtered = data_typed[(data_typed['StatusCurre	L8 (data_typed).csv	L10 (permit_status_filterec	delete_rows_filter
6	Python	1	filtered_residential = permit_status_filtered[permit_status_filte	L10 (permit_status_filtered).csv	L11 (filtered_residential).cs	delete_rows_filter
7	Python	1	cutoff = filtered_residential[filtered_residential['IssuedDate'] <	L11 (filtered_residential).csv	L12 (cutoff).csv	delete_rows_filter
8	Python	1	demolitions_cut = filtered_residential	L11 (filtered_residential).csv	L13 (demolitions_cut).csv	identical_operation
9	Python	1	filtered_homes = demolitions_cut[(demolitions_cut['PermitCla	L13 (demolitions_cut).csv	L14 (filtered_homes).csv	delete_rows_filter
10	Python	1	demolitions_full = filtered_homes	L14 (filtered_homes).csv	L15 (demolitions_full).csv	identical_operation
914	R	32	p78_input1= read.csv("r78_input1.csv")		L4 (p78_input1).csv	create_tables_load
915	R	32	TBL_15=filter(p78_input1,`y` < 1.900000)	L4 (p78_input1).csv	L5 (TBL_15).csv	delete_rows_filter
916	R	32	TBL_7=group_by(TBL_15,`x`)	L5 (TBL_15).csv	L6 (TBL_7).csv	identical_operation
917	R	32	TBL_3=summarise(TBL_7,a=mean(`y`))	L6 (TBL_7).csv	L7 (TBL_3).csv	combine_rows_summarize
918	R	32	TBL_1=inner_join(TBL_3,p78_input1)	L7 (TBL_3).csv|L4 (p78_input1).cs	L8 (TBL_1).csv	combine_tables_inner_join
919	R	32	morpheus=mutate(TBL_1,z=y / a)	L8 (TBL_1).csv	L9 (morpheus).csv	create_columns_mutate
920	R	32	morpheus=as.data.frame(morpheus)	L9 (morpheus).csv	L10 (morpheus).csv	identical_operation
921	R	32	morpheus=select(morpheus,1,3,4,2,5)	L10 (morpheus).csv	L11 (morpheus).csv	transform_tables_rearrange

Space of Table Changes

- For different transformations, what changes can be made to tables?

Transformation Type
create_tables_load
create_tables_load
combine_tables_extend
create_columns_mutate
delete_rows_filter
delete_rows_filter
delete_rows_filter
identical_operation
delete_rows_filter



Space of Table Changes

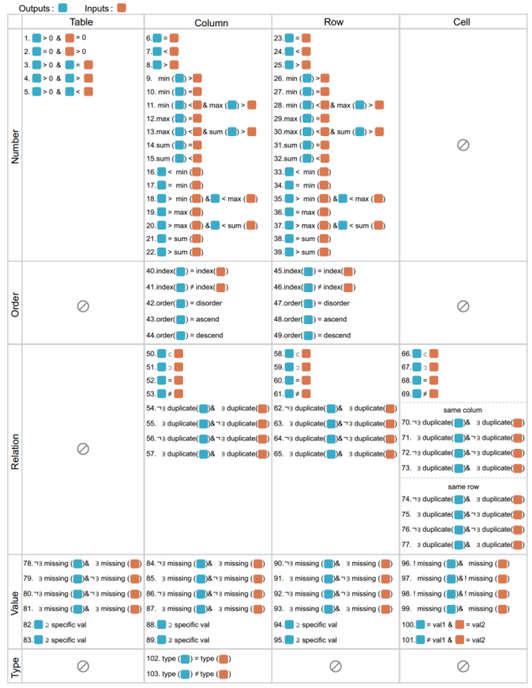
- For different transformations, what changes can be made to tables?

Input Table	Output Table
	L3 (raw_full).csv
	L4 (raw_partial).csv
L3 (raw_full).csv|L4 (raw_partial).c	L5 (data_raw).csv
L5 (data_raw).csv	L8 (data_typed).csv
L8 (data_typed).csv	L10 (permit_status_filterec
L10 (permit_status_filtered).csv	L11 (filtered_residential).cs
L11 (filtered_residential).csv	L12 (cutoff).csv
L11 (filtered_residential).csv	L13 (demolitions_cut).csv
L13 (demolitions_cut).csv	L14 (filtered_homes).csv

record

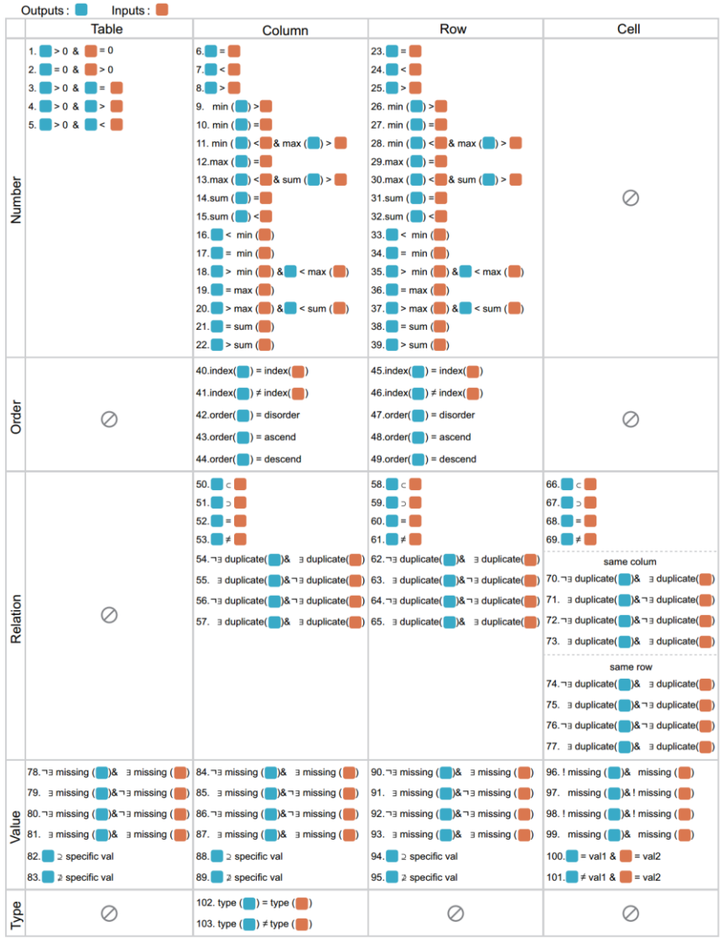
table differences

open and
axial coding



the space of table changes

Space of Table Changes



2 dimensions

Types of data objects

1. Tables
2. Columns
3. Rows
4. Cells

Properties of data changes

1. Number
2. Order
3. Relation
4. Value
5. Type

103 characteristics

$C_1 — C_{103}$

Number

Table

1. the number of output tables is greater than zero and the number of input tables is zero
2. the number of output tables is zero and the number of input tables is greater than zero
3. the number of output tables is greater than zero and the number of output tables is equal to the number of input tables
4. the number of output tables is greater than zero and the number of output tables is greater than the number of input tables
5. the number of output tables is greater than zero and the number of output tables is less than the number of input tables

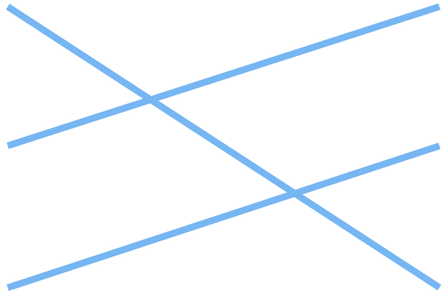
Column

6. the number of columns in the output table is equal to the number of columns in the input table
7. the number of columns in the output table is less than the number of columns in the input table

Build a Mapping

Transformations

- create_columns_mutate
- delete_rows_filter
- delete_columns_select_keep
- combine_rows_summarize
- transform_columns_rename
- transform_columns_mutate
- transform_tables_sort
- transform_columns_replace



Characteristics

6. ■ = ■
7. ■ < ■
8. ■ > ■
9. min (■) > ■
10. min (■) = ■
11. min (■) < ■ & max (■) > ■
12. max (■) = ■
13. max (■) < ■ & sum (■) > ■

■ Output

■ Input

Build a Mapping

Will the characteristic appear after performing a transformation?

Transformations	Inevitable	Impossible	Possible
• create_columns_mutate	• C_1 , C_3	• C_6 , C_9	• C_2 , …
• delete_rows_filter	• C_2 , C_5	• C_1 , C_7	• C_3 , …
• delete_columns_select_keep	• C_3 , C_5	• C_2 , C_8	• C_1 , …
• combine_rows_summarize			
• transform_columns_rename			
• transform_columns_mutate			
• transform_tables_sort			
• transform_columns_replace			
⋮	⋮	⋮	⋮

Build a Mapping

Will the characteristic appear after performing a transformation?

Transformations

- delete_rows_filter

Gender	Age
Male	23
Female	
Male	31

→

Gender	Age
Male	23
Male	31

Keep Rows Where Gender is Male

Inevitable ✓

- The number of columns remains unchanged
- . . .

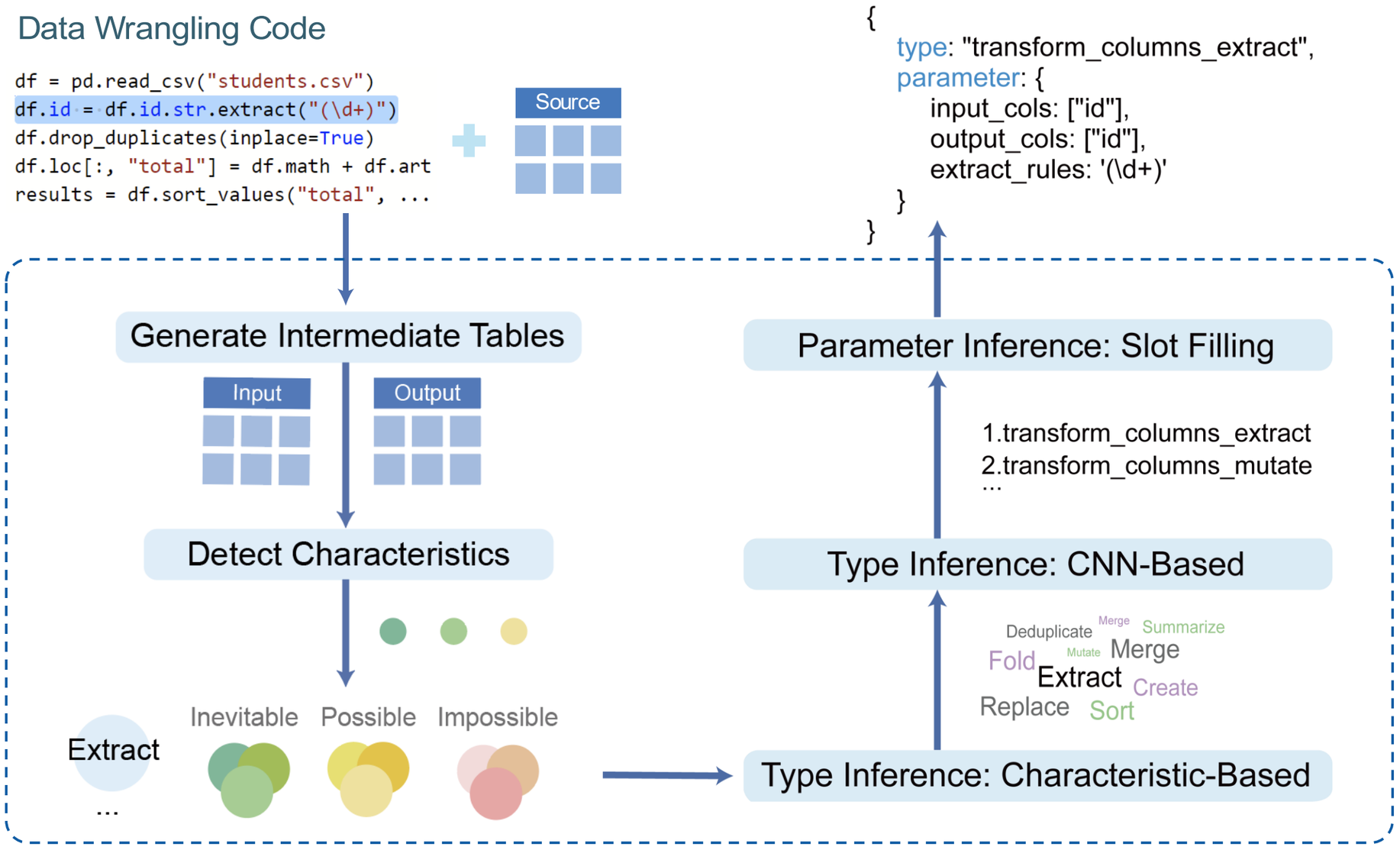
Impossible ✗

- The number of rows has increased
- . . .

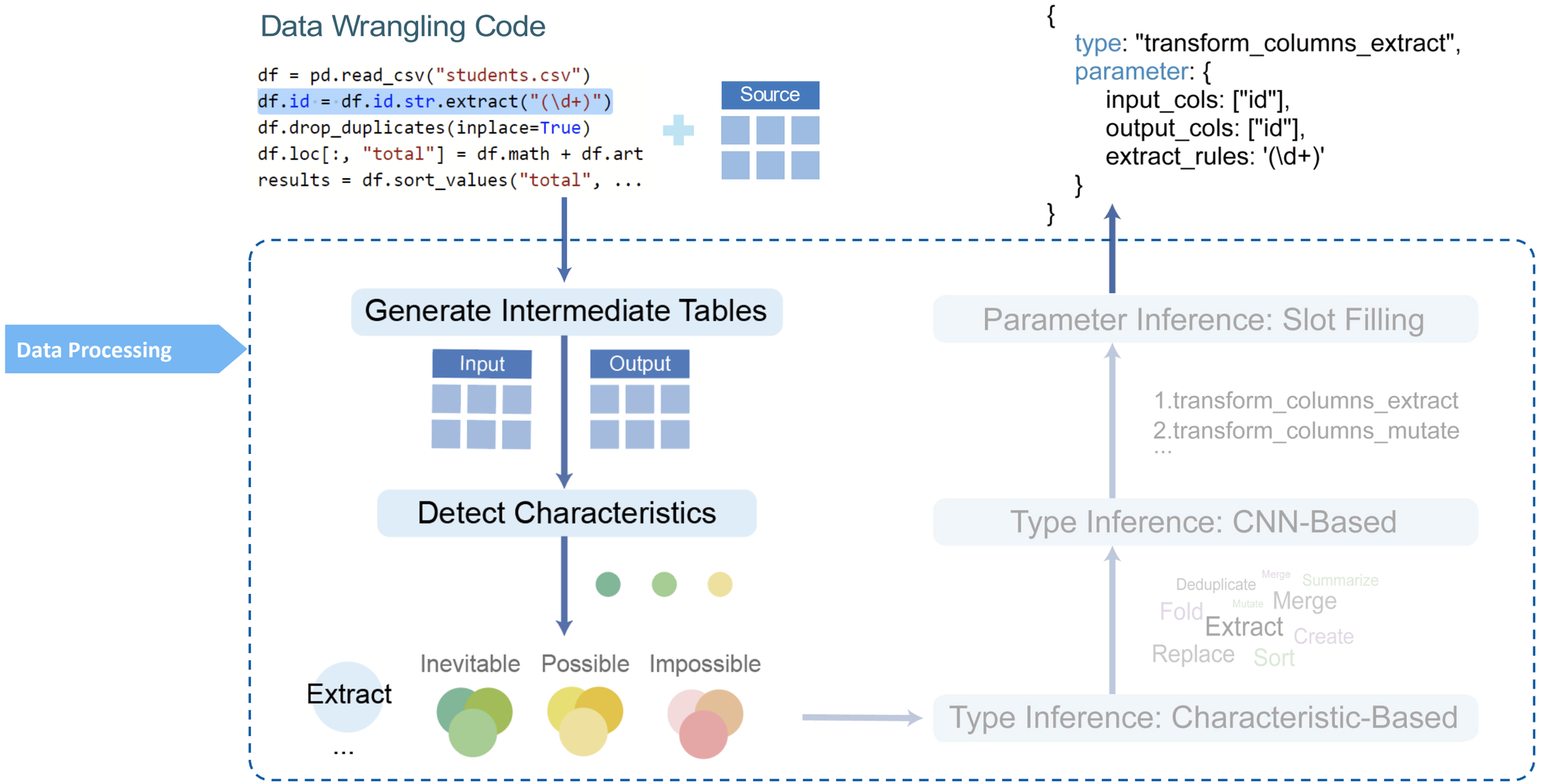
Possible ?

- The number of rows has decreased
- . . .

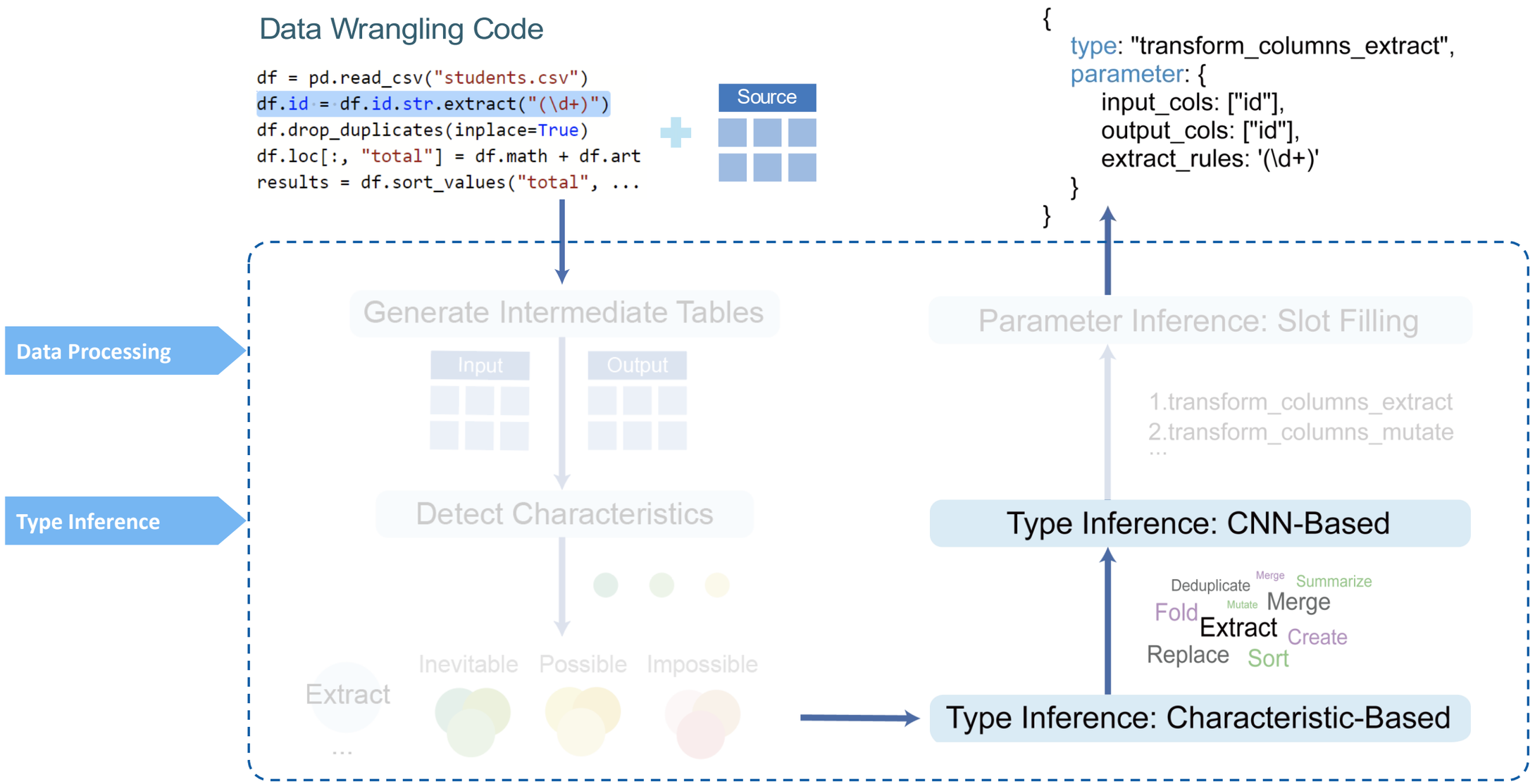
COMANTICS Overview



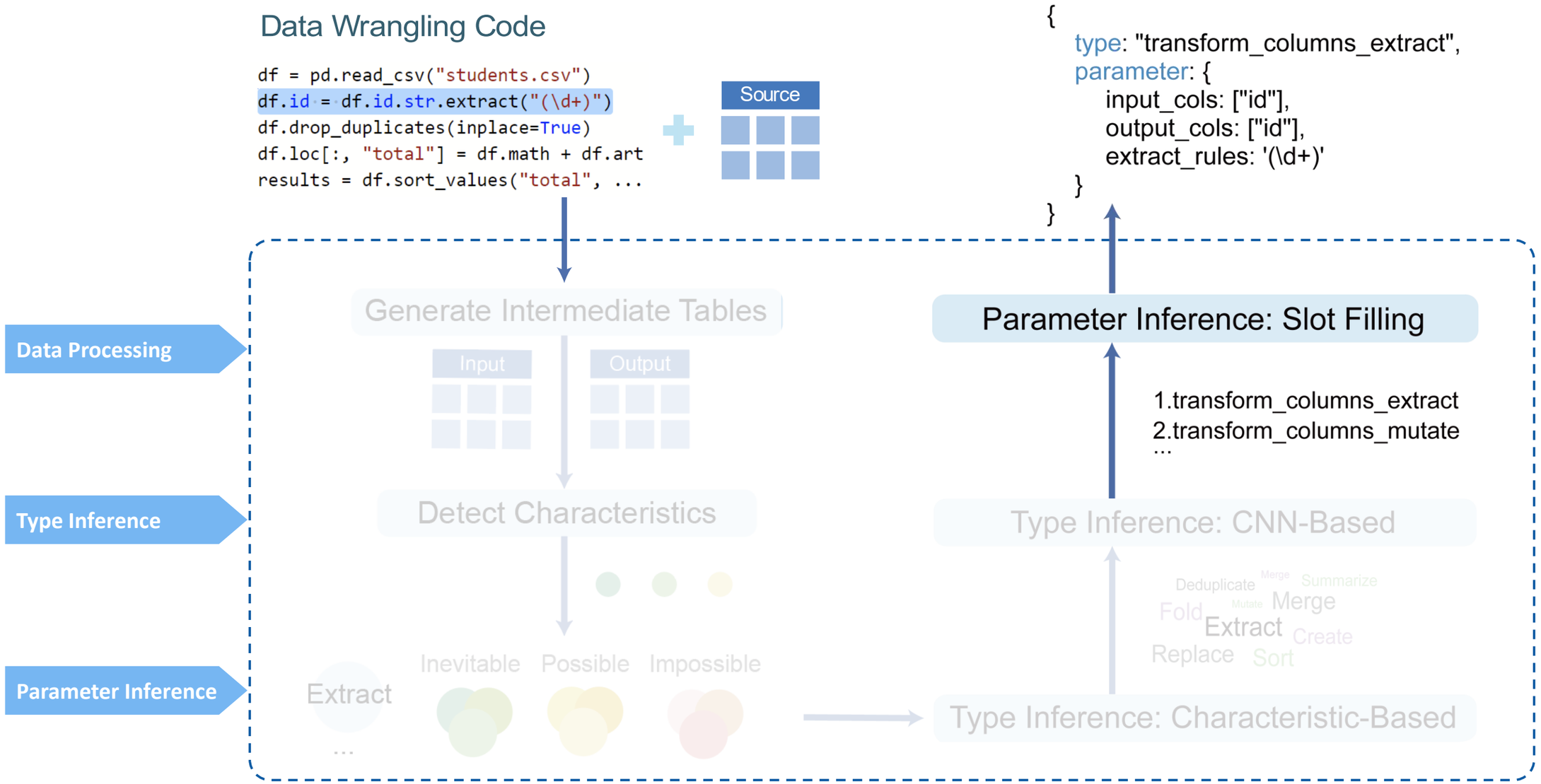
COMANTICS Overview



COMANTICS Overview



COMANTICS Overview



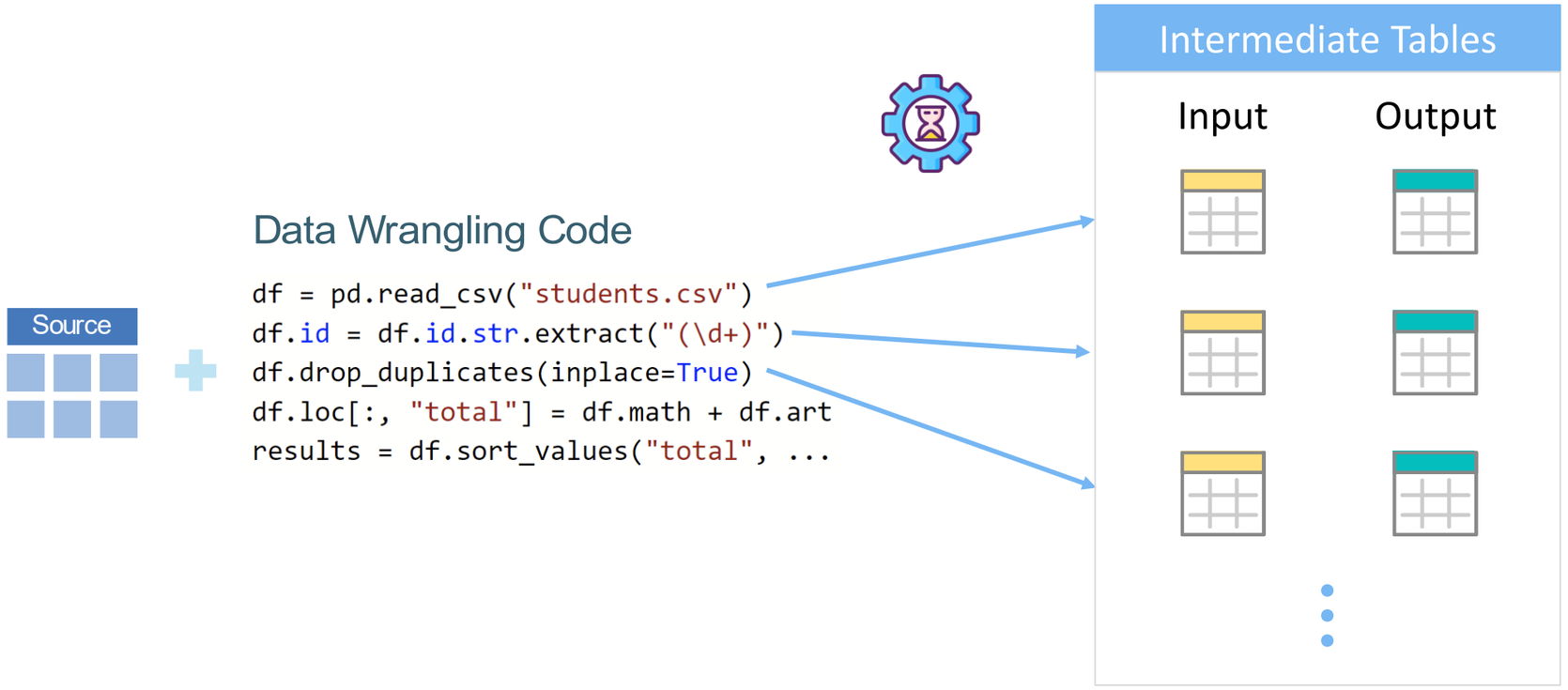
COMANTICS

Data Processing

- Generate Tables

Type Inference

Parameter Inference



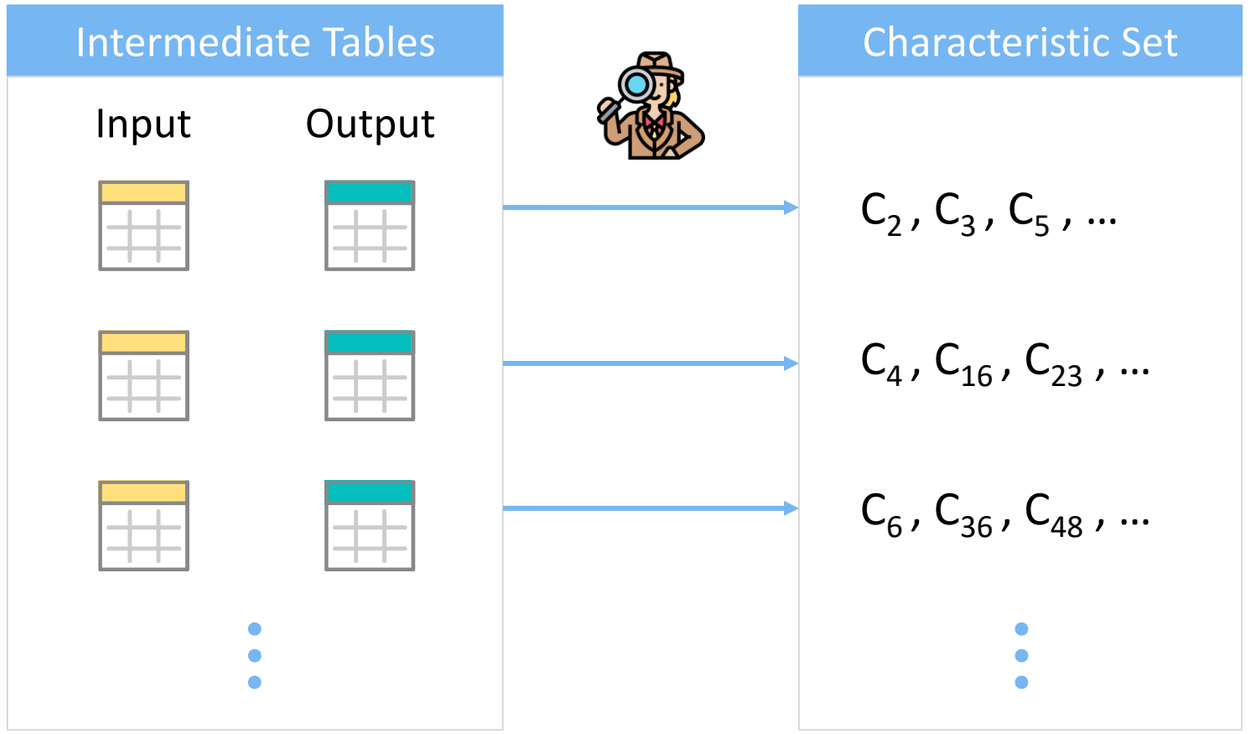
COMANTICS

Data Processing

- Generate Tables
- Detect Characteristics

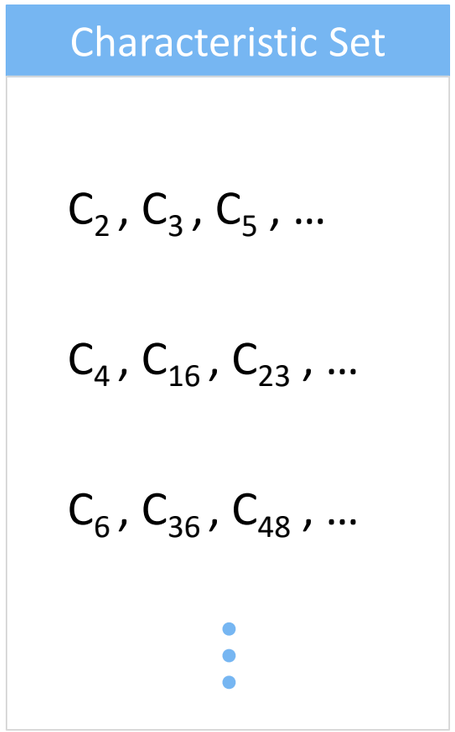
Type Inference

Parameter Inference



COMANTICS

- **Data Processing**
  - Generate Tables
  - Detect Characteristics
- **Type Inference**
  - Characteristic-Based
- **Parameter Inference**



Two Strategies: filter out transformations whose

1. inevitable set is not a subset of the characteristic set
2. impossible set intersects the characteristic set

COMANTICS

- **Data Processing**
  - Generate Tables
  - Detect Characteristics
- **Type Inference**
  - Characteristic-Based
- **Parameter Inference**

Characteristic Set

C_2 , C_3 , C_5 , …

C_4 , C_{16} , C_{23} , …

C_6 , C_{36} , C_{48} , …

Two Strategies: filter out transformations whose

1. inevitable set is not a subset of the characteristic set
2. impossible set intersects the characteristic set

Mapping

Transformation	Inevitable	Impossible	Possible
T_1	C_1, C_3	C_6, C_9	C_2, …
T_2	C_2, C_5	C_1, C_7	C_3, …
T_3	C_3, C_5	C_2, C_8	C_1, …
…	…	…	…

COMANTICS

Data Processing

- Generate Tables
- Detect Characteristics

Type Inference

- Characteristic-Based

Parameter Inference

Characteristic Set
C_2 , C_3 , C_5 , …
C_4 , C_{16} , C_{23} , …
C_6 , C_{36} , C_{48} , …
⋮

Two Strategies: filter out transformations whose

1. inevitable set is not a subset of the characteristic set
2. impossible set intersects the characteristic set

Mapping

Transformation	Inevitable	Impossible	Possible
✗ T_1	C_1, C_3	C_6, C_9	C_2, …
T_2	C_2, C_5	C_1, C_7	C_3, …
T_3	C_3, C_5	C_2, C_8	C_1, …
…	…	…	…

Strategy 1

COMANTICS

- **Data Processing**
  - Generate Tables
  - Detect Characteristics
- **Type Inference**
  - Characteristic-Based
- **Parameter Inference**

Characteristic Set

$C_2, C_3, C_5, \ldots$

$C_4, C_{16}, C_{23}, \ldots$

$C_6, C_{36}, C_{48}, \ldots$

Two Strategies: filter out transformations whose

1. inevitable set is not a subset of the characteristic set
2. impossible set intersects the characteristic set

Mapping

Transformation	Inevitable	Impossible	Possible	
✗ T_1	C_1, C_3	C_6, C_9	$C_2, \ldots$	Strategy 1
T_2	C_2, C_5	C_1, C_7	$C_3, \ldots$	
✗ T_3	C_3, C_5	C_2, C_8	$C_1, \ldots$	Strategy 2
…	…	…	…	

COMANTICS

Data Processing

- Generate Tables
- Detect Characteristics

Type Inference

- Characteristic-Based

Parameter Inference

Gender	Age
Male	23
Female	28
Male	31

Gender	Age
Male	23
Female	28
Male	31

Ambiguity 1

the output is identical to the input

1. Remove Duplicate Rows
2. Sort Age in Ascending Order

⋮

Gender	Age
Male	23
Female	
Male	31

Gender	Age
Male	23
Male	31

Ambiguity 2

more than two transformations lead to the same changes

1. Keep Rows Where Gender is Male
2. Remove Rows with Missing Values

⋮

Mapping

Transformation	Inevitable	Impossible	Possible
✗ T_1	C_1, C_3	C_6, C_9	C_2, …
T_2	C_2, C_5	C_1, C_7	C_3, …
✗ T_3	C_3, C_5	C_2, C_8	C_1, …
…	…	…	…

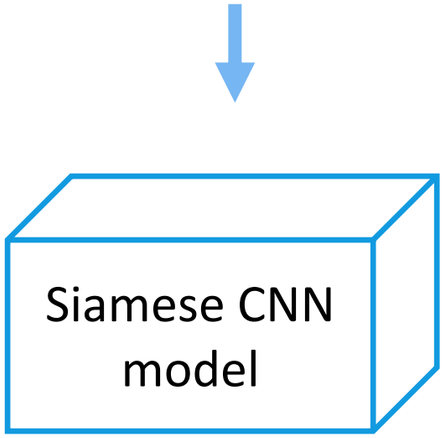
→ Transformation candidates:

T_2 , T_6 , T_{17} , T_{21}

COMANTICS

- Data Processing
  - Generate Tables
  - Detect Characteristics
- **Type Inference**
  - Characteristic-Based
  - **CNN-Based**
- Parameter Inference

Transformation candidates: T_2 , T_6 , T_{17} , T_{21}



suitable for

1. few shot problems
2. imbalanced data

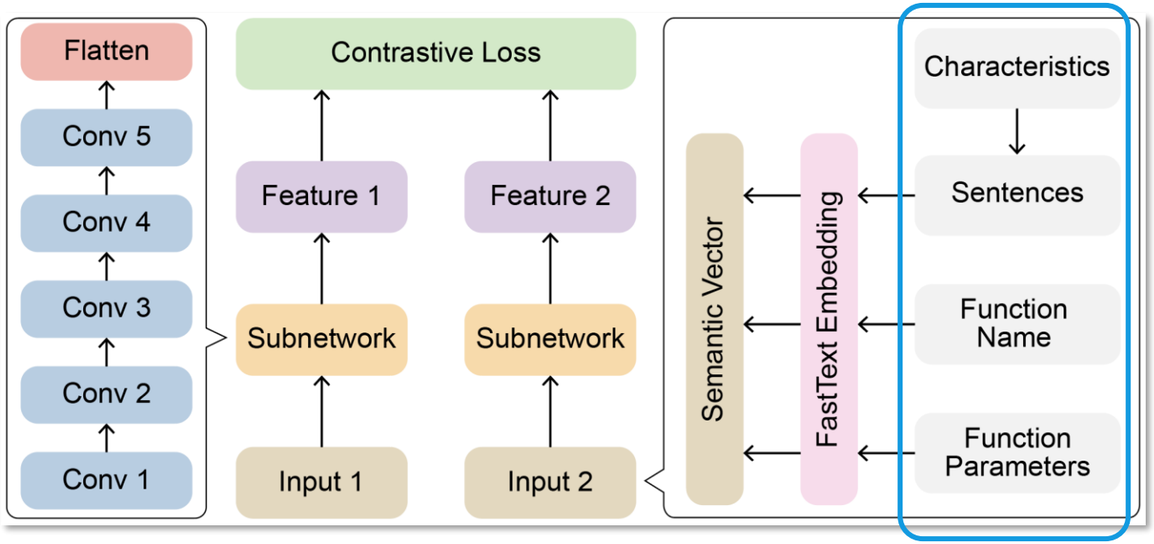
Transformation likelihood: $T_6 > T_2 > T_{21} > T_{17}$

COMANTICS

- **Data Processing**
  - Generate Tables
  - Detect Characteristics
- **Type Inference**
  - Characteristic-Based
  - CNN-Based
- **Parameter Inference**

Transformation candidates: T_2 , T_6 , T_{17} , T_{21}

Siamese CNN model



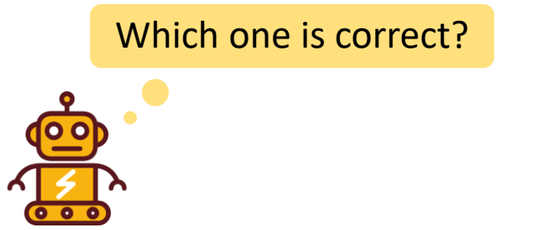
Please refer to the paper for details.

Transformation likelihood: $T_6 > T_2 > T_{21} > T_{17}$

COMANTICS

- **Data Processing**
  - Generate Tables
  - Detect Characteristics
- **Type Inference**
  - Characteristic-Based
  - CNN-Based
- **Parameter Inference**
  - Slot Filling

Transformation likelihood: $T_6 > T_2 > T_{21} > T_{17}$



T_6 ✗	T_2 ✓	T_{21}	T_{17}
• P1: ✓	• P1: ✓	• P1:	• P1:
• P2: ✓	• P2: ✓	• P2:	• P2:
• P3: ✗	• P3: ✓	• P3:	• P3:
• P4:		• P4:	• P4:
		• P5:	

slot

Transformation type: T_2

Parameters: {
P1: xxx ,
P2: xxx ,
P3: xxx
}

```
df.id = df.id.str.extract("(\d+)")
```

```
{
   type: "transform_columns_extract",
   parameter: {
      input_cols: ["id"],
      output_cols: ["id"],
      extract_rules: '(\d+)'
   }
}
```

Experiments

- Two aspects of evaluation:
  - How each part of COMANTICS contributes to the overall **performance**
  - How well can COMANTICS **generalize** to different programming languages
- Dataset
  - 606 lines of Python code
  - 315 lines of R code
- Results

Table 2. The training time, Top-1, and Top-3 performances of COMANTICS and its CNN-based component in different experiment settings.

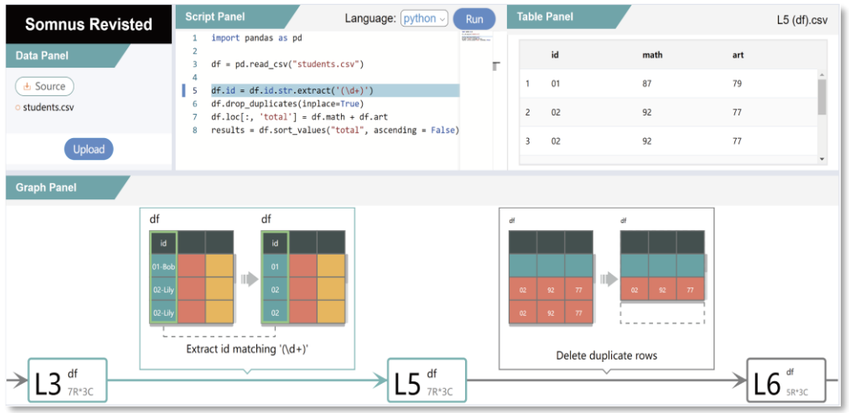
	Setting		CNN		COMANTICS		Time
	Train	Test	Top-1	Top-3	Top-1	Top-3	(minutes)
1st	Python	R	19.1	30.3	53.8	76.1	19.5
2nd	R	Python	23.2	32.3	44.5	82.6	15.1
3rd	Python	Python	62.1	83.3	80.3	94.7	18.3
4th	All	Python	72.0	90.9	90.2	98.5	21.9
5th	All	All	78.0	92.2	92.2	99.0	19.6

Example Applications

jupyter

```
# Transform Columns: Extract (\d+) from id into id
df.id = df.id.str.extract('(\d+)')
# Delete Rows: Delete duplicate rows
df.drop_duplicates(inplace=True)
# Create Columns: Mutate math and art into total
df.loc[:, 'total'] = df.math + df.art
# Transform Tables: Sort total in descending order
results = df.sort_values("total", ascending = False)
results
```

Code Annotation



Semantics Visualization

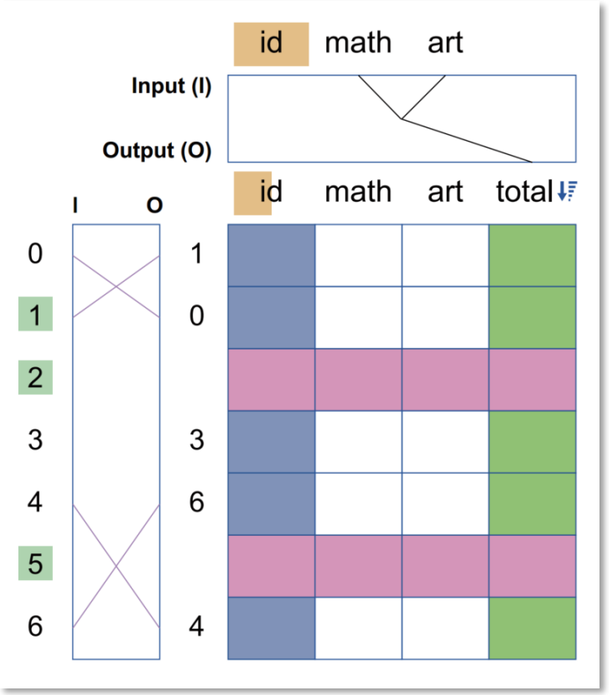


Table Comparison

jupyter Comantics Case 最新检查点: 4 小时前 （更改未保存）

Logout

File Edit View Insert Cell Kernel Navigate Help

可信 siamese

运行 代码

In [1]:
```
import pandas as pd
```

In [2]:
```
df = pd.read_csv("students.csv")
df
```

Out[2]:

	id	math	art
0	01-Bob	87	79
1	02-Lily	92	77
2	02-Lily	92	77
3	04-Alan	64	93
4	05-John	68	81
5	05-John	68	81
6	07-Iris	71	84

In [3]:
```
df.id = df.id.str.extract('(\d+)')
df.drop_duplicates(inplace=True)
df.loc[:, 'total'] = df.math + df.art
results = df.sort_values("total", ascending = False)
results
```

Out[3]:

	id	math	art	total
1	02	92	77	169
0	01	87	79	166
3	04	64	93	157
6	07	71	84	155
4	05	68	81	149

This case shows that the Comantics-augmented Jupyer notebook can perform code annotation automatically.

Conclusion

Contributions

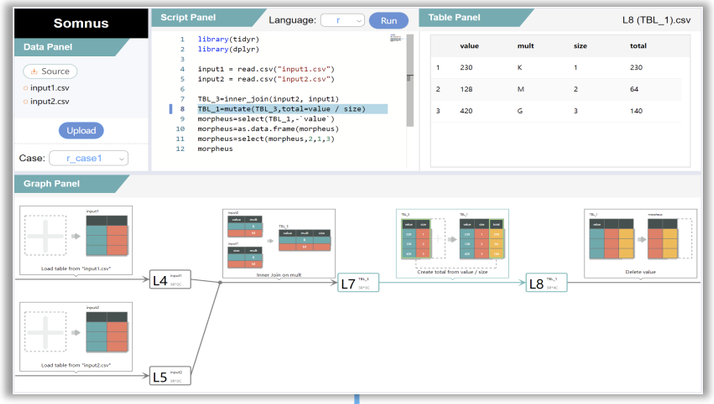
- summarize a space including 103 characteristics of table changes
- construct COMANTICS, a novel pipeline for inferring semantics of data wrangling scripts
- build a real-world dataset including 921 lines of wrangling code where each is annotated with a data transformation
- perform quantitative experiments to evaluate the performance of COMANTICS
- apply COMANTICS to three applications in different domains

Future Work

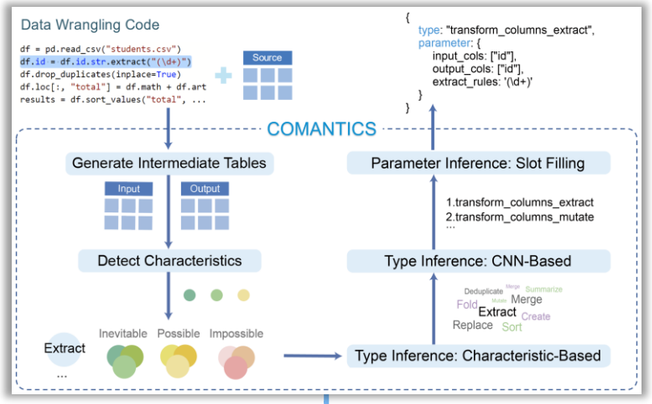
- improve the performance of COMANTICS by refining the Data Preprocessing step and Type Inference module
- enhance the capability to meet the need for production usage
- incorporate COMANTICS in a broad range of applications, including spreadsheet software, data wrangling tool, and source-to-source compiler

IDG
Interactive Data Group
https://zjuidg.org

SOMNUS



COMANTICS



IEEE TVCG 2022

IEEE VIS 2022

Thank You!

E-mail: *kaixiong@zju.edu.cn*