mage: Fluid Moves Between Code and Graphical Work in Computational Notebooks

Kery, M. B., Ren, D., Hohman, F., Moritz, D., Wongsuphasawat, K., & Patel, K. (2020)
In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*

Wenjing Lin
Role: Paper Author

01

Introduction

Berkeley
UNIVERSITY OF CALIFORNIA

Problem Statement

⛔ No “GUI” cell provided in computational notebooks

NO-CODE meets CODE

📈 Graphical interface tools (GUI)

- Charting tools
- Dashboards
- Spreadsheets

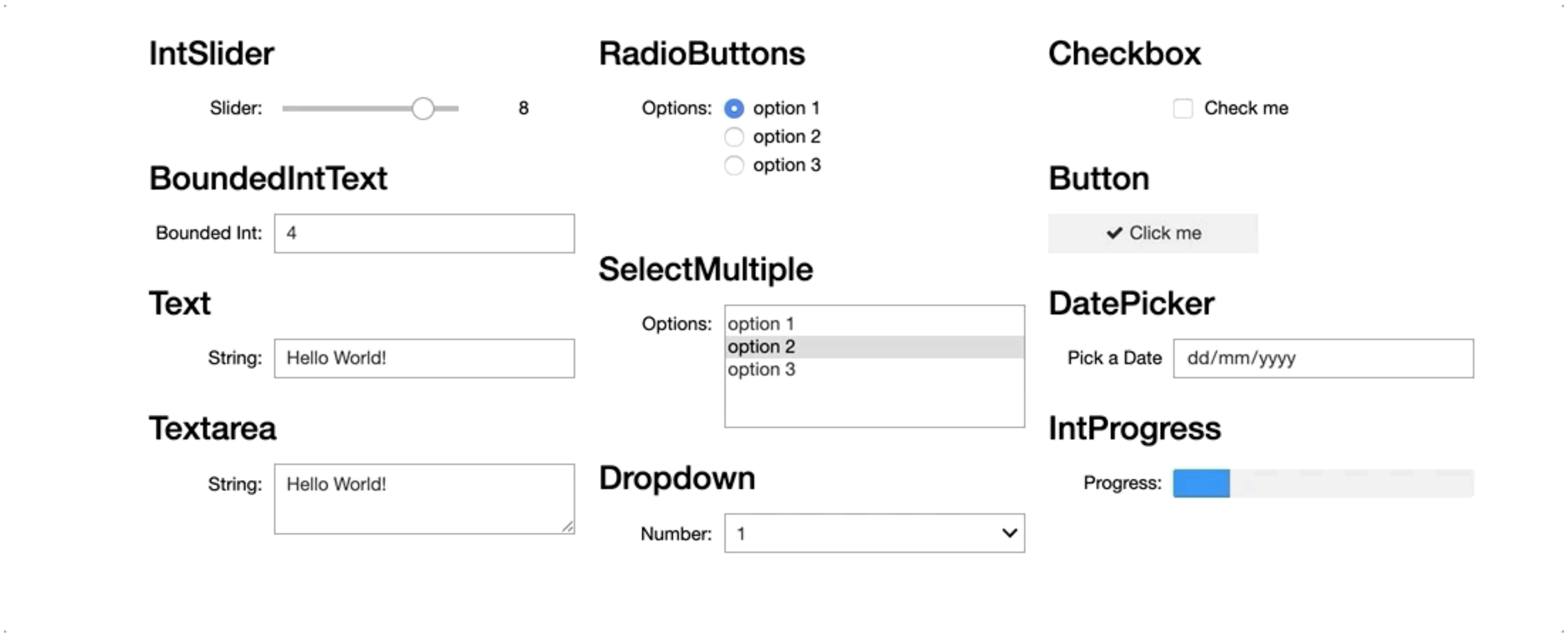
🧑‍💻 3 formats

- Text
- Code
- Output

01 Introduction

Problem Statement

⚠️ Some “GUI” widgets but not enough to support full data workflow



IntSlider

Slider: 8

BoundedIntText

Bounded Int: 4

Text

String: Hello World!

Textarea

String: Hello World!

RadioButtons

Options: ● option 1 ○ option 2 ○ option 3

SelectMultiple

Options: option 1 option 2 option 3

Dropdown

Number: 1

Checkbox

□ Check me

Button

✓ Click me

DatePicker

Pick a Date dd/mm/yyyy

IntProgress

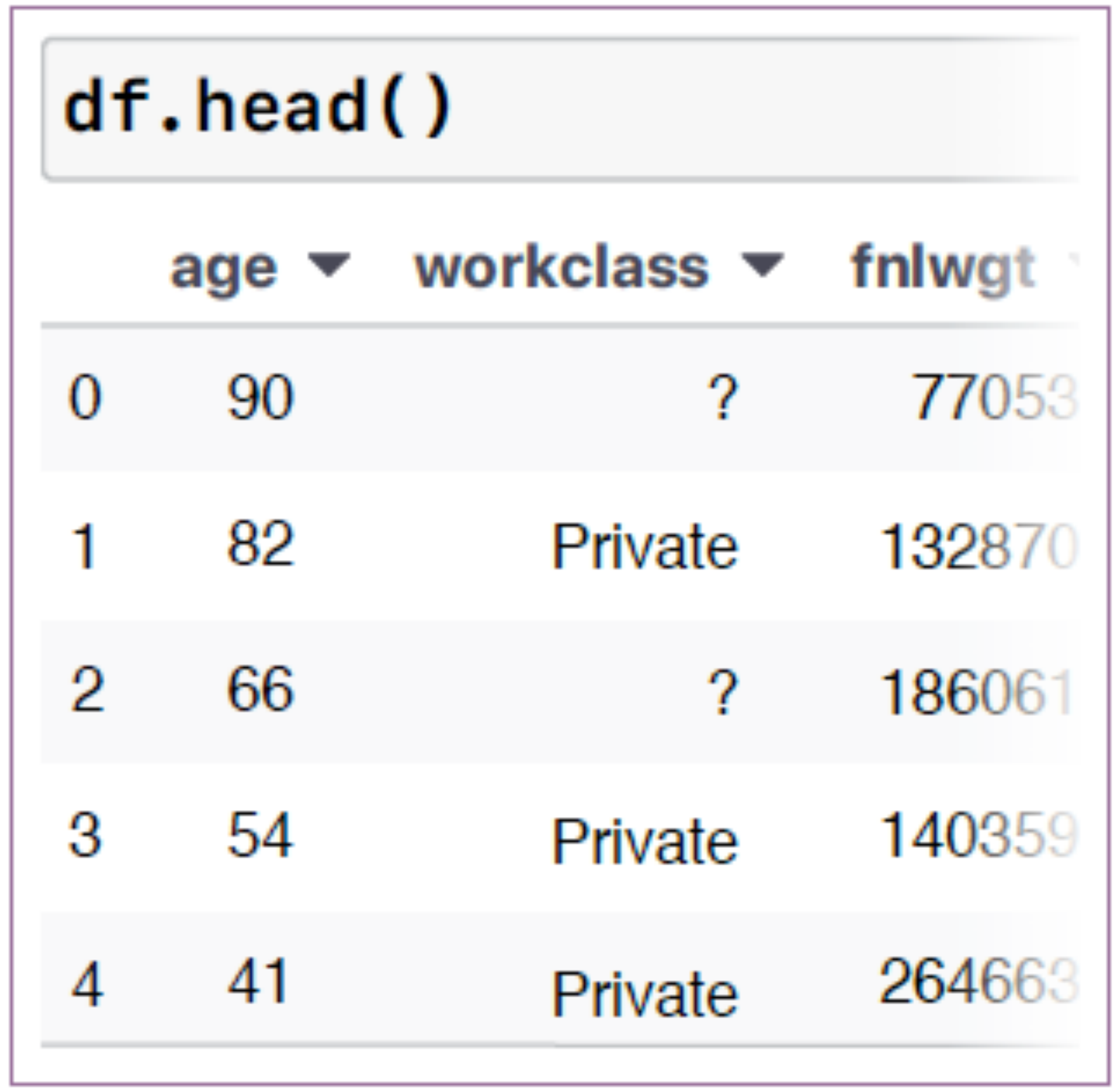
Progress:

01 Introduction

Proposed Solution

✅ Mage: View-only output to interactive interface

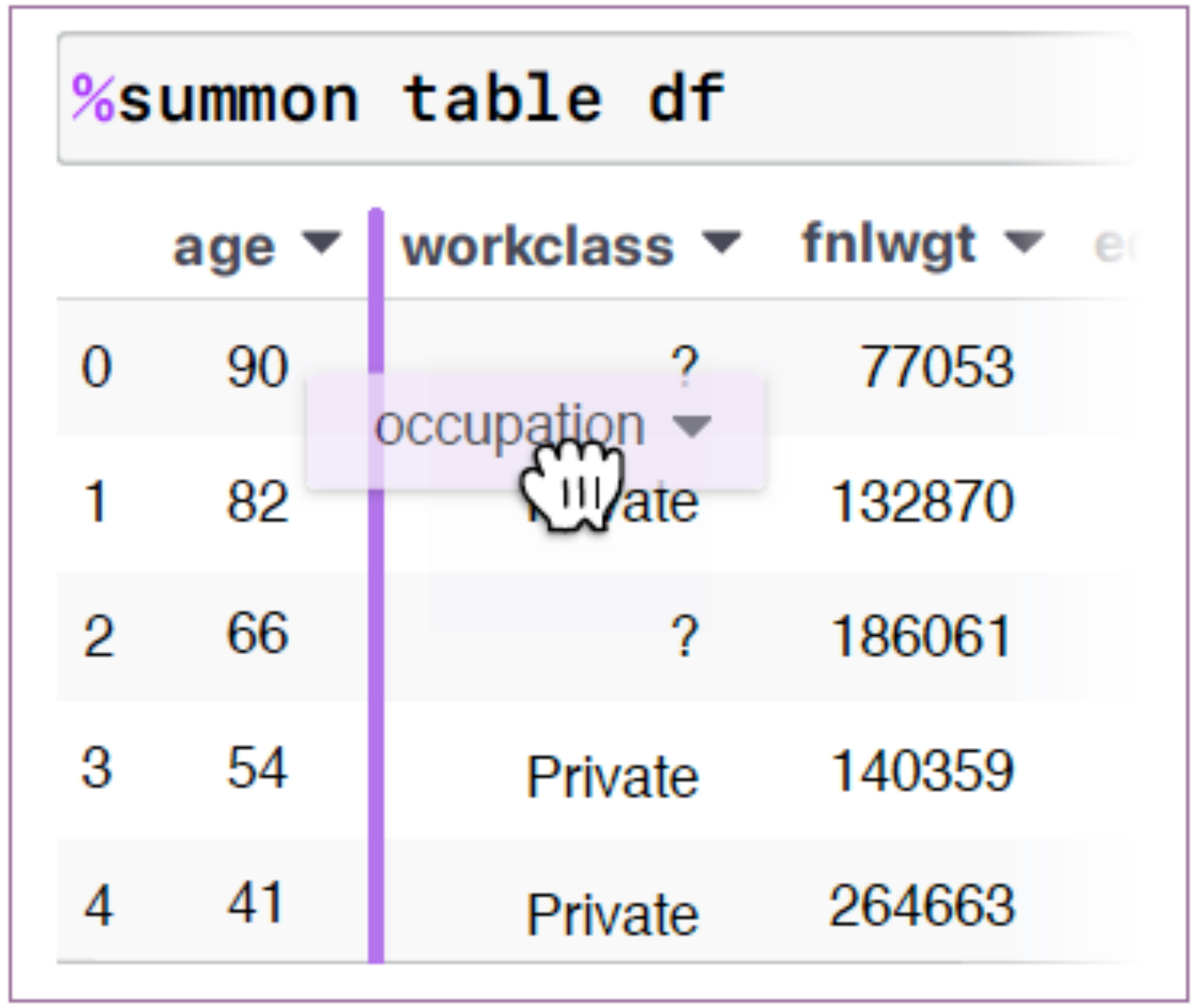
standard notebook



```
df.head()
```

	age	workclass	fnlwgt
0	90	?	77053
1	82	Private	132870
2	66	?	186061
3	54	Private	140359
4	41	Private	264663

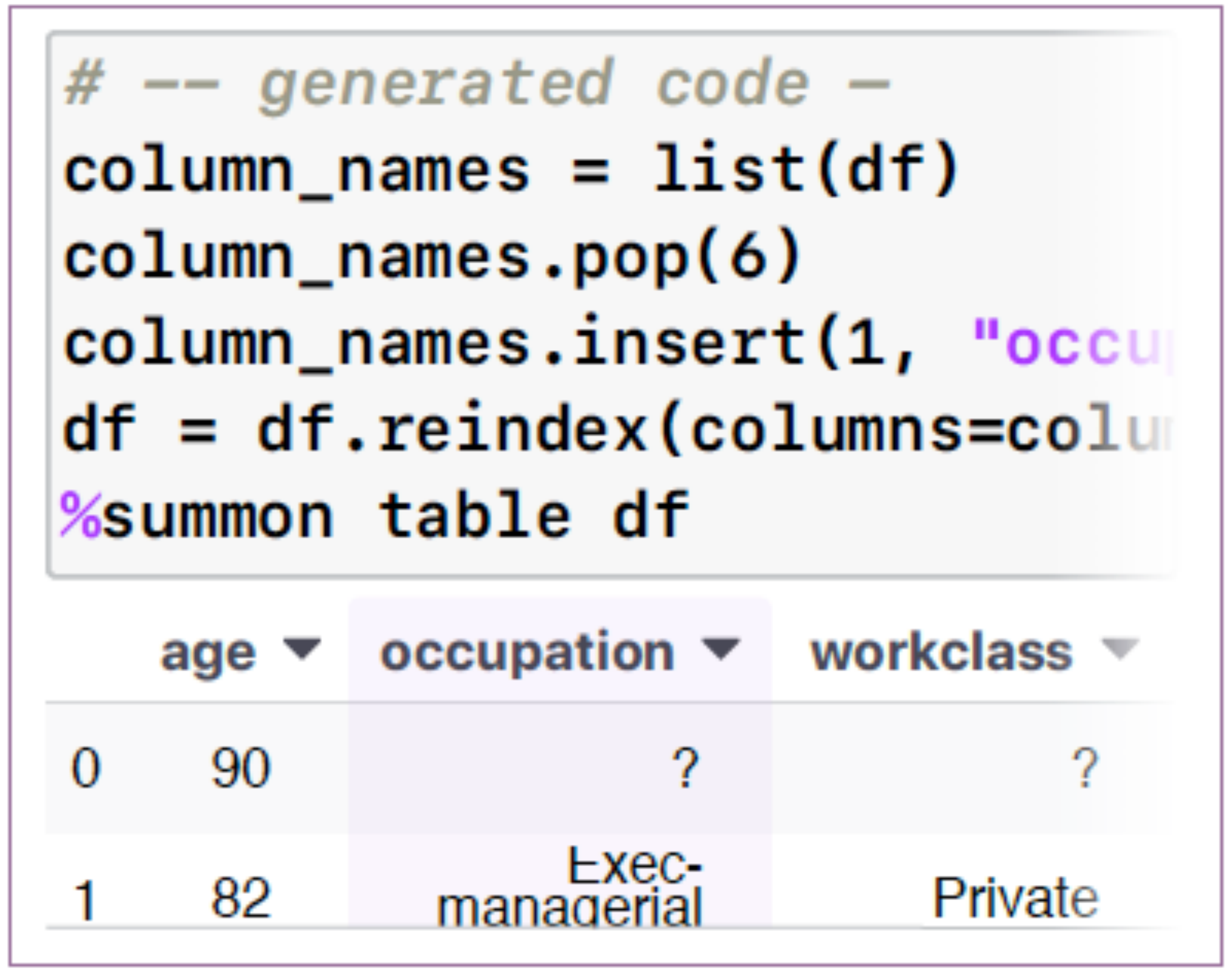
1 mage : user edits table



```
%summon table df
```

	age	workclass	fnlwgt	e
0	90	?	77053	
1	82	Private	132870	
2	66	?	186061	
3	54	Private	140359	
4	41	Private	264663	

2 mage : edits reflect in code



```
# -- generated code -
column_names = list(df)
column_names.pop(6)
column_names.insert(1, "occu
df = df.reindex(columns=colu
%summon table df
```

	age	occupation	workclass
0	90	?	?
1	82	Exec-managerial	Private

Solution Novelty

✨ An API allows tool builders make GUI widget flexible

Previous approach

👀 Visual domain-focused

- Drawing
- UI Design
- Data visualization
- 3D Modeling

🙋 Synchronization vs. Expressivity

- Limited expressiveness

1️⃣ **Parameterization:** lack of replicability

2️⃣ **Domain-specific language:** lack of generality

3️⃣ **Programming by demonstration:** lack of generality

Flexible GUI/code system

Mage

❌ GUI/code tool

✅ Jupyter Notebook extension/API: empower tool builders

02

System Overview

02 System Overview

Getting Started

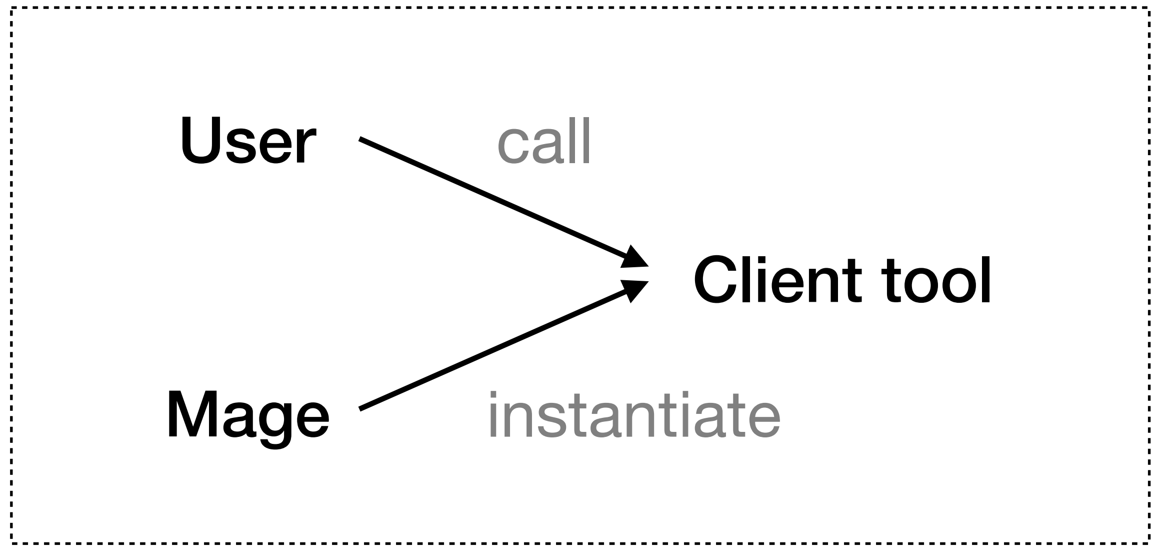
🛠 Suppose we are building an interactive spreadsheet-like tool called “Table”

0 **`Table`**

Interactive spreadsheet-like tool

1 **`Configurate mage API`**

- Client tool name
- Parameter requirements: data receive from user
- UI view: JavaScript class

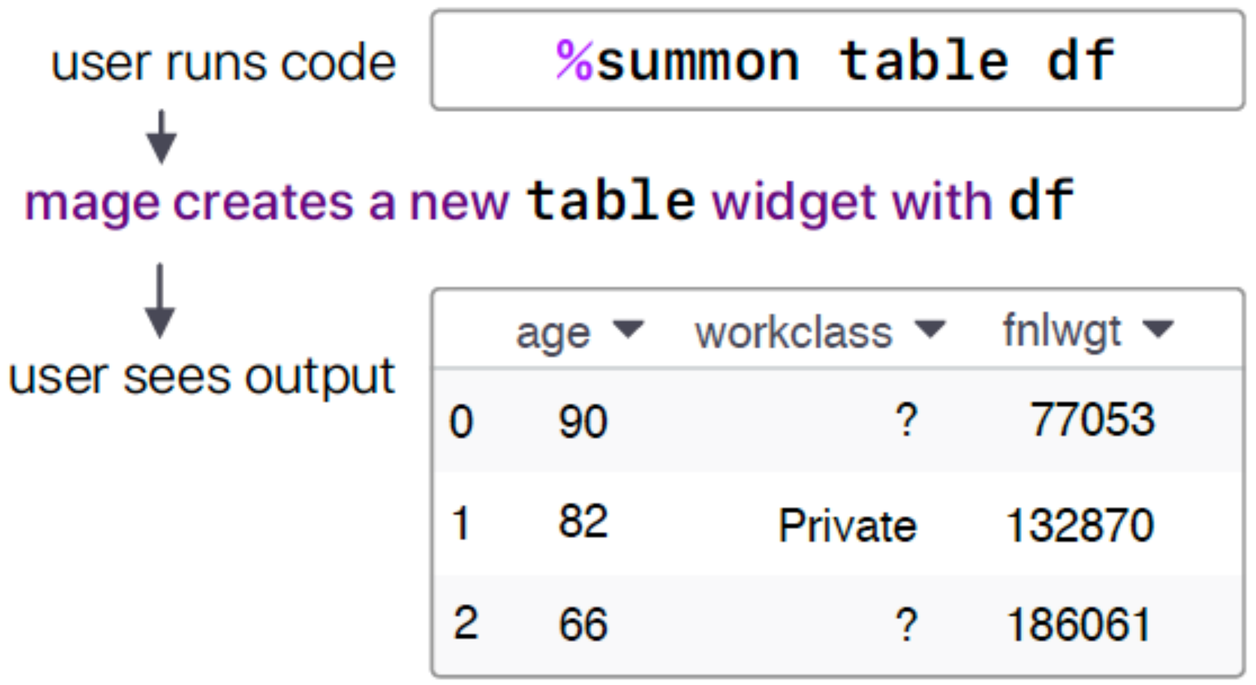


At this stage, UI won’t be able to affect `df`
Jupyter notebooks’ sandboxing: copy of the user’s runtime

2 **`Call Table tool`**

Magics command syntax

```
%summon <tool name> <parameters>
```



user runs code

```
%summon table df
```

mage creates a new `table` widget with `df`

user sees output

	age	workclass	fnlwgt
0	90	?	77053
1	82	Private	132870
2	66	?	186061

Berkeley
UNIVERSITY OF CALIFORNIA

02 System Overview

GUI to Code

✍🏻 Mage modifies state through `handoff()` & template

Client

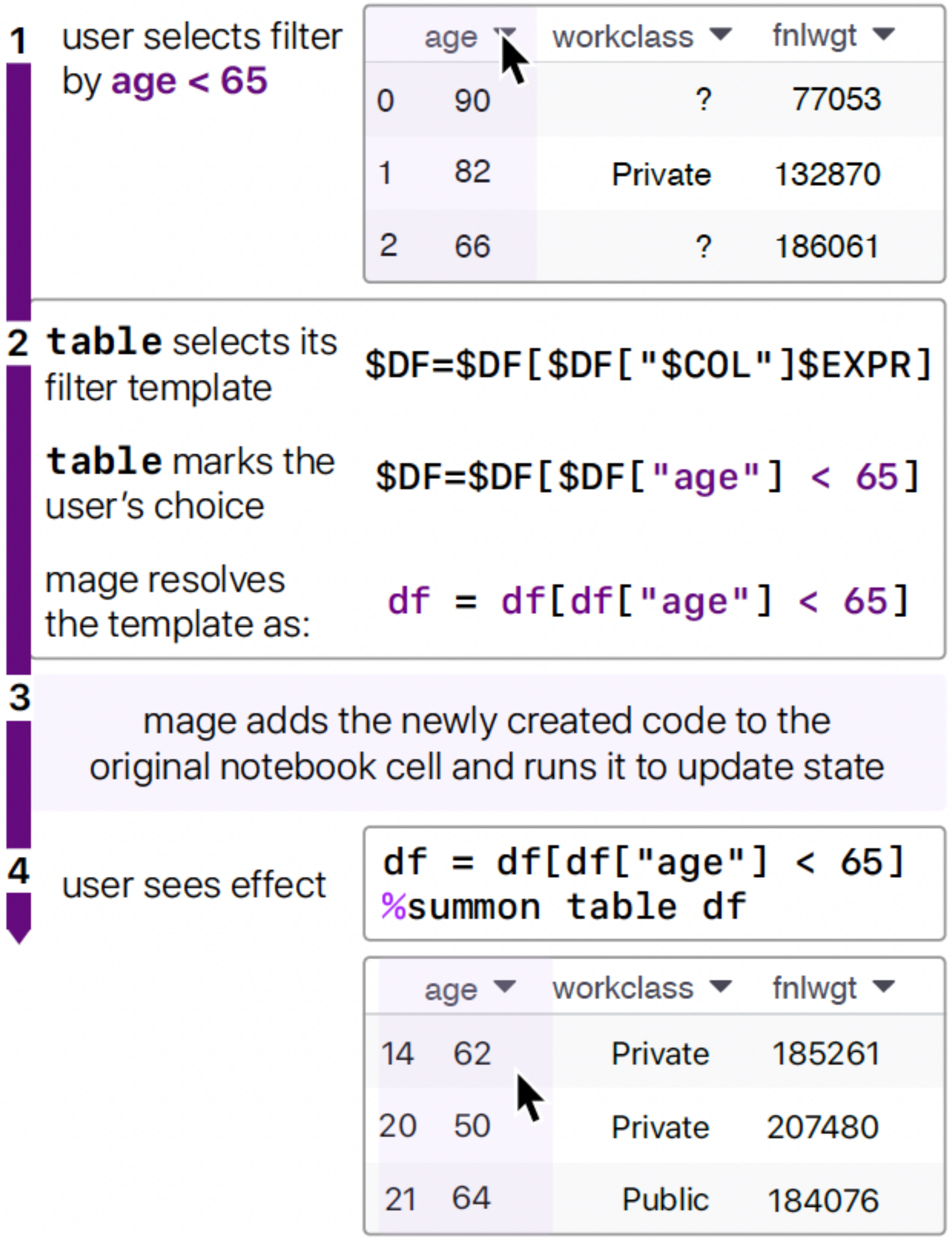
Table filter

- Filter in UI also apply to user's data variable
- Call mage API `handoff(<template>, <data>)`
- Provide `code templates`

Mage

Be instructed to affect state

- Receive `handoff()` API call
- Resolve all blanks in the `code template`
- Insert new code into code cell
- Request notebook to re-execute the cell



1 user selects filter by **age < 65**

	age ▾	workclass ▾	fnlwgt ▾
0	90	?	77053
1	82	Private	132870
2	66	?	186061

2 `table` selects its filter template

```
$DF=$DF[$DF["$COL"]$EXPR]
```

`table` marks the user's choice

```
$DF=$DF[$DF["age"] < 65]
```

mage resolves the template as:

```
df = df[df["age"] < 65]
```

3 mage adds the newly created code to the original notebook cell and runs it to update state

4 user sees effect

```
df = df[df["age"] < 65]
%summon table df
```

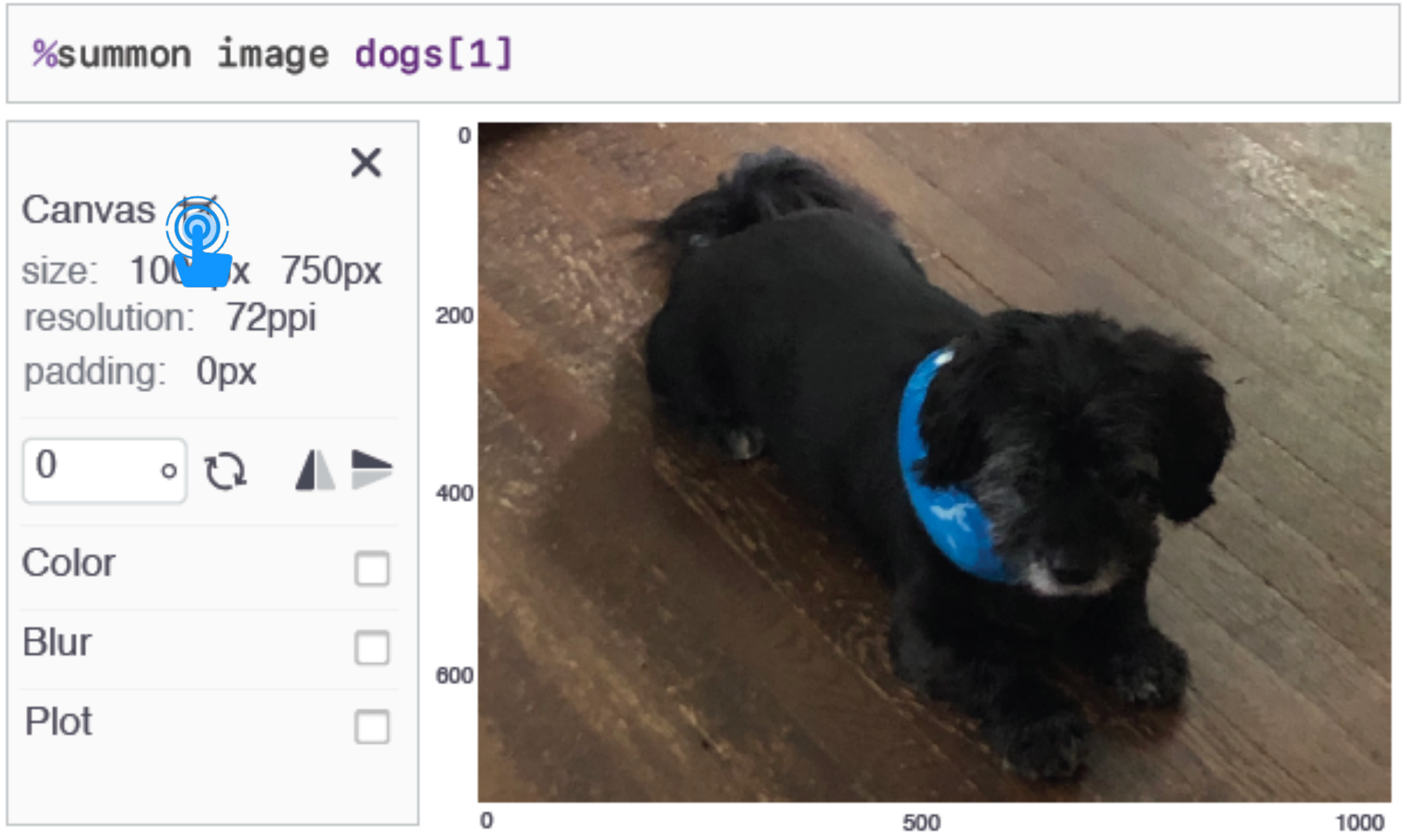
	age ▾	workclass ▾	fnlwgt ▾
14	62	Private	185261
20	50	Private	207480
21	64	Public	184076

Berkeley
UNIVERSITY OF CALIFORNIA

02 System Overview
Code to GUI

Mage uses client tools’ code template to read actions

Image: Another client GUI tool



```
%summon image dogs[1]
```

Canvas
size: 10[illegible]x 750px
resolution: 72ppi
padding: 0px

Color □
Blur □
Plot □

User

1 Change the dimensions of the image

2 Make edits to the code

```
crop_img = dogs[1][0:750, 100:850]
foo(crop_img)
%summon image crop_img
```

Mage

Match back to action template

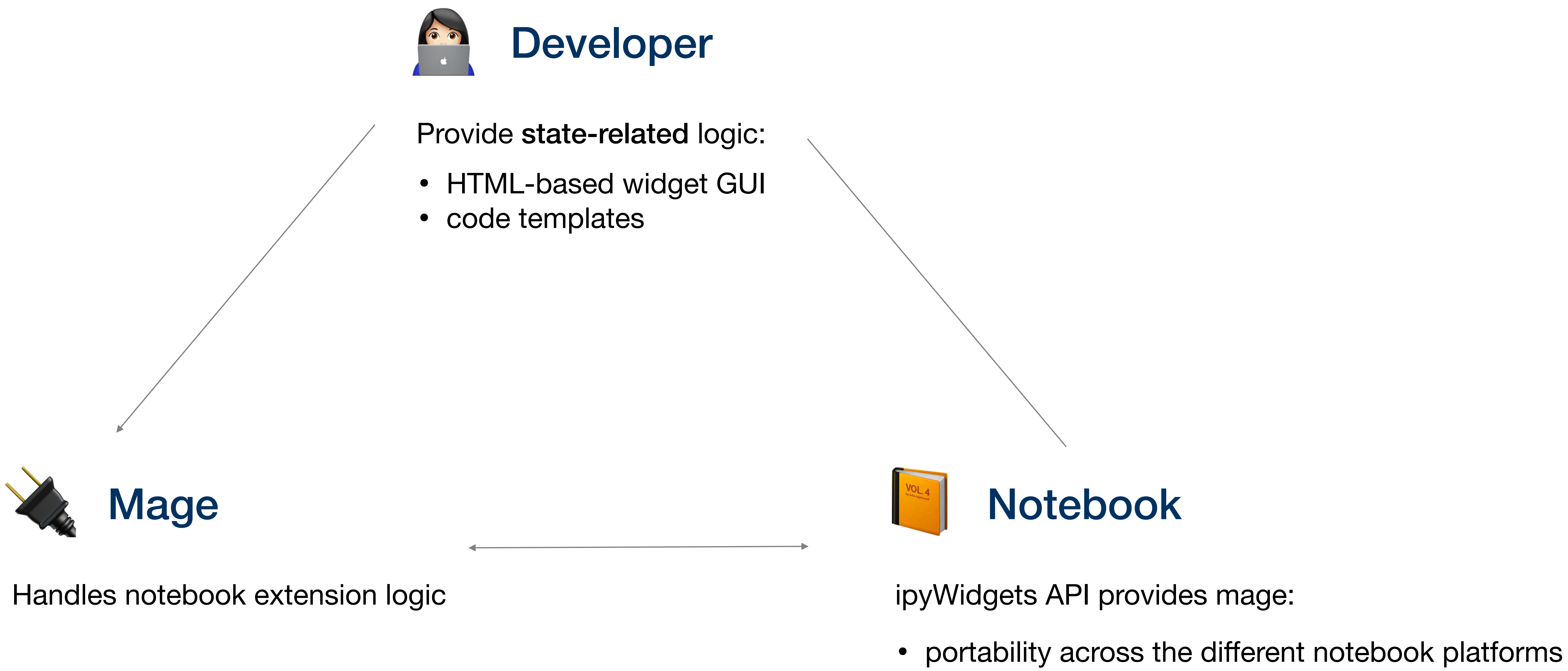
```
crop_img = dogs[1][0:750, 0:750]
%summon image crop_img
```

`foo()` is not within the action template

mage assumes all code preceding an unrecognized line of code is untouchable user code

Berkeley
UNIVERSITY OF CALIFORNIA

📡 Mage makes GUI widgets more generalizable



👩🏻‍💻 Developer

Provide **state-related** logic:

- HTML-based widget GUI
- code templates

🔌 Mage

Handles notebook extension logic

📙 Notebook

ipyWidgets API provides mage:

- portability across the different notebook platforms

03

Use Cases

Berkeley
UNIVERSITY OF CALIFORNIA

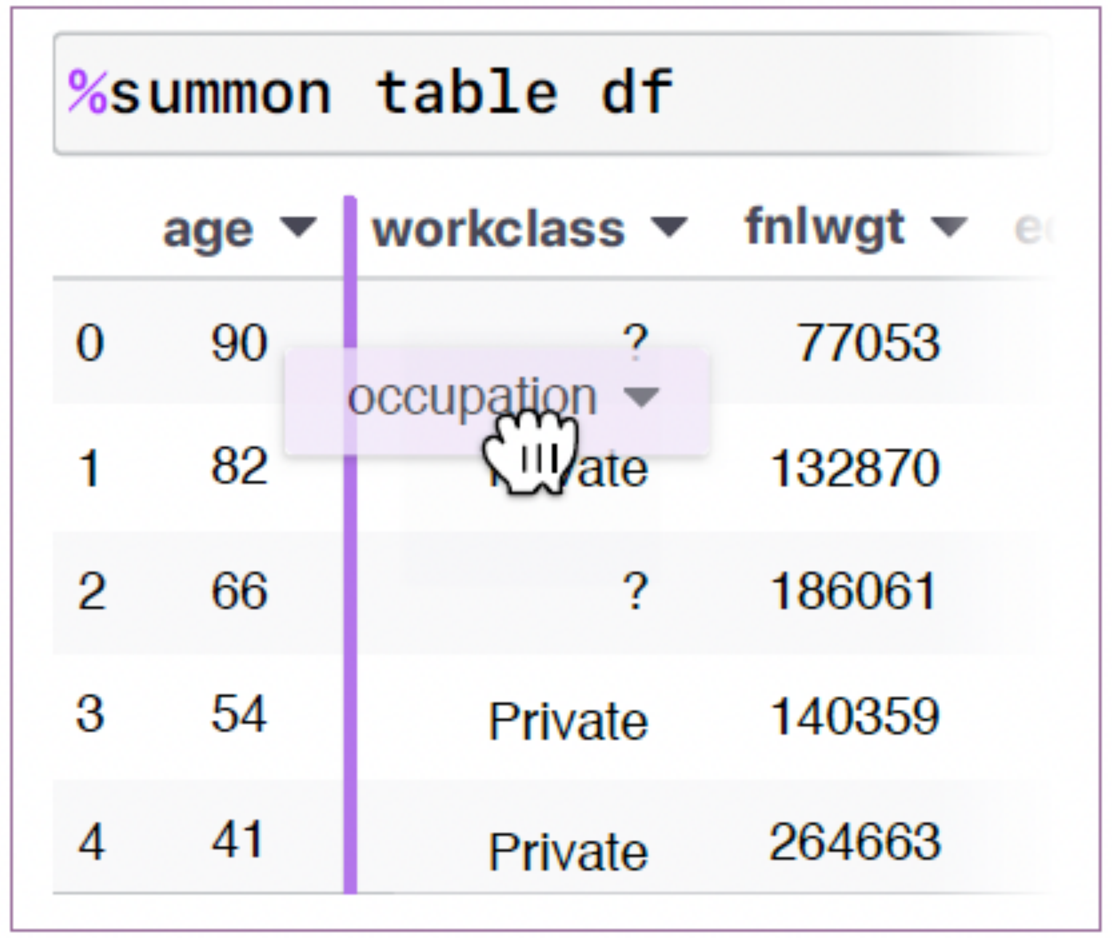
03 Use Cases

Table & Image

💃 Mage enables table & image data interactions

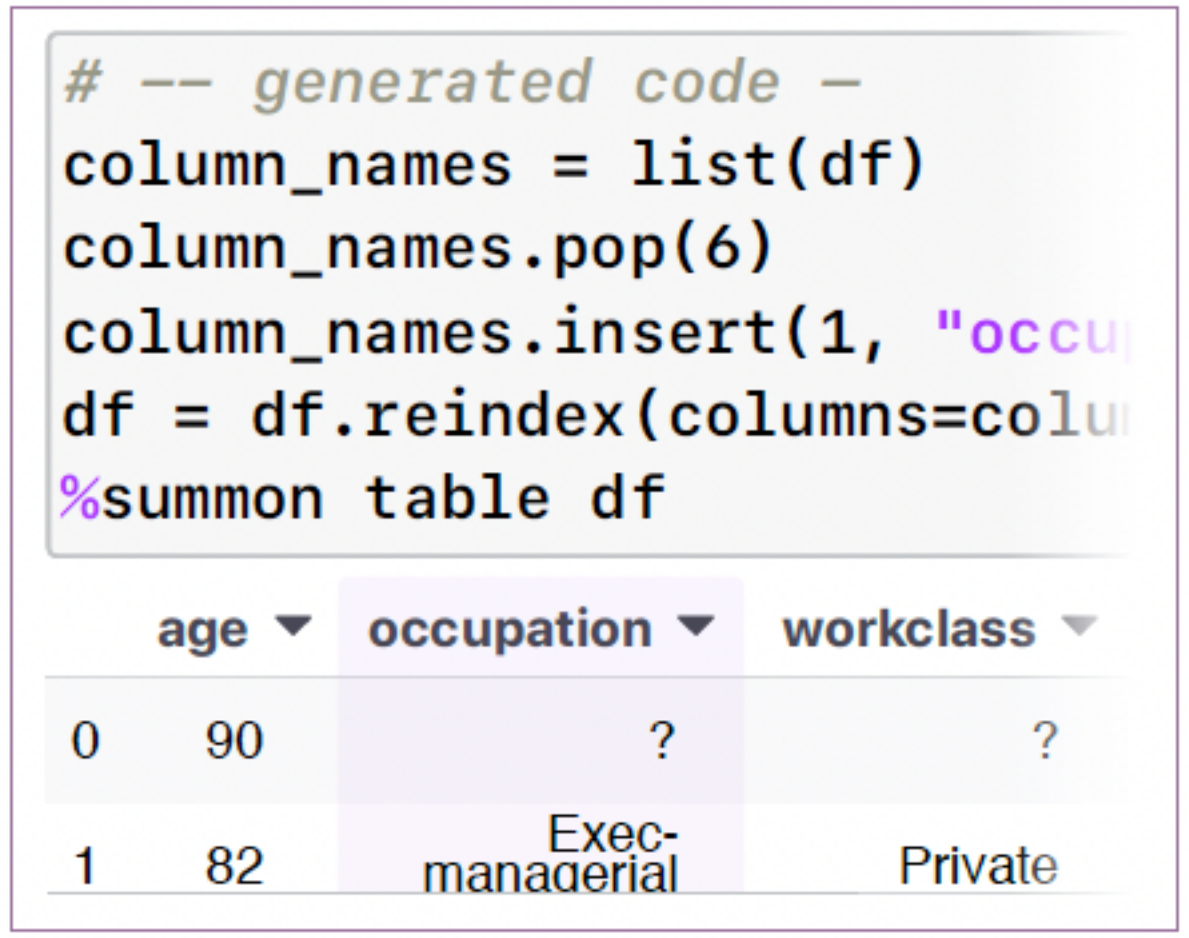
1️⃣ `table`

transforming and cleaning data



```
%summon table df
```

	age	workclass	fnlwgt	e
0	90	?	77053	
1	82	Private	132870	
2	66	?	186061	
3	54	Private	140359	
4	41	Private	264663	

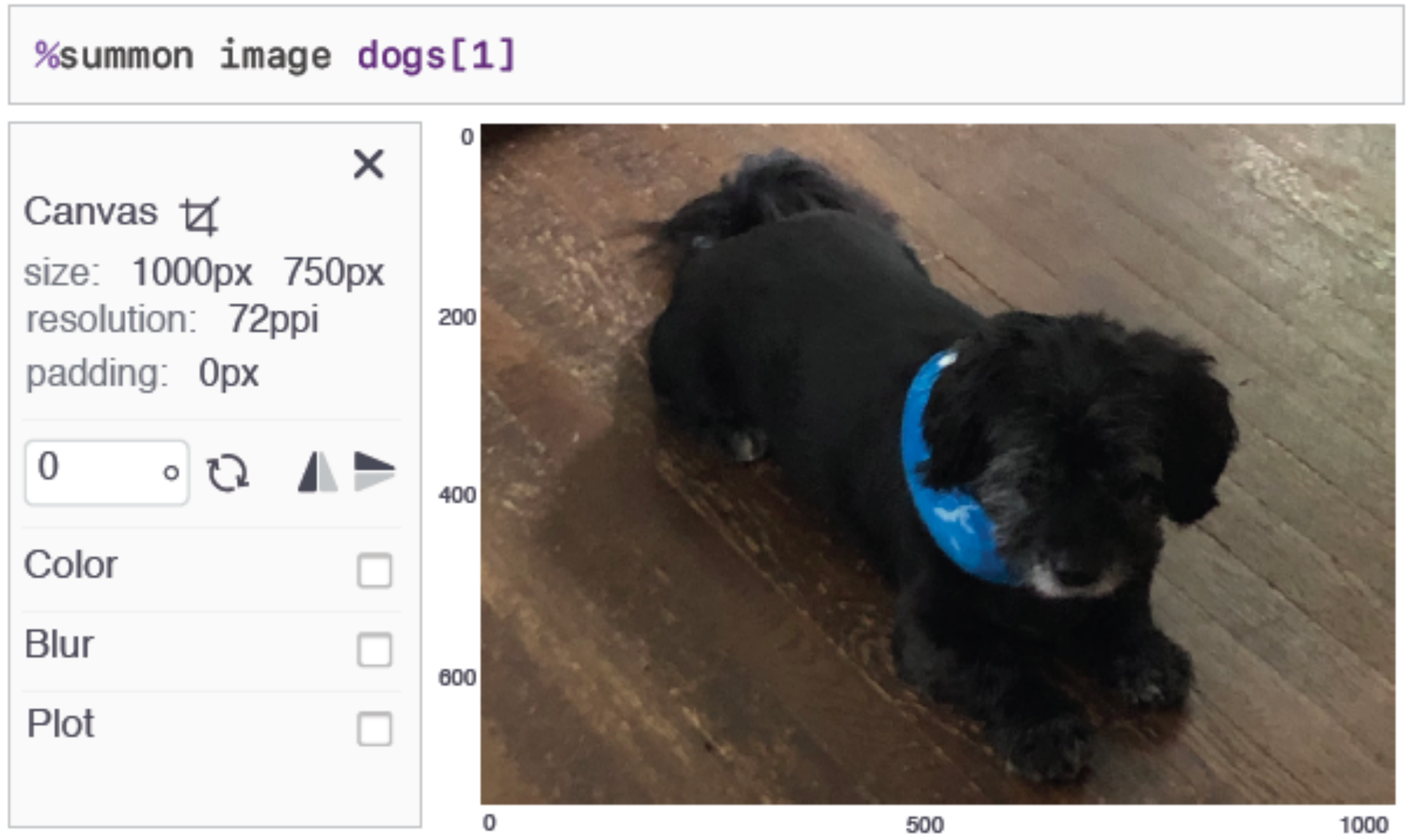


```
# -- generated code —
column_names = list(df)
column_names.pop(6)
column_names.insert(1, "occu
df = df.reindex(columns=colu
%summon table df
```

	age	occupation	workclass
0	90	?	?
1	82	Exec-managerial	Private

2️⃣ `image`

editing image data



```
%summon image dogs[1]
```

Canvas

size: 1000px 750px

resolution: 72ppi

padding: 0px

0 °

Color □

Blur □

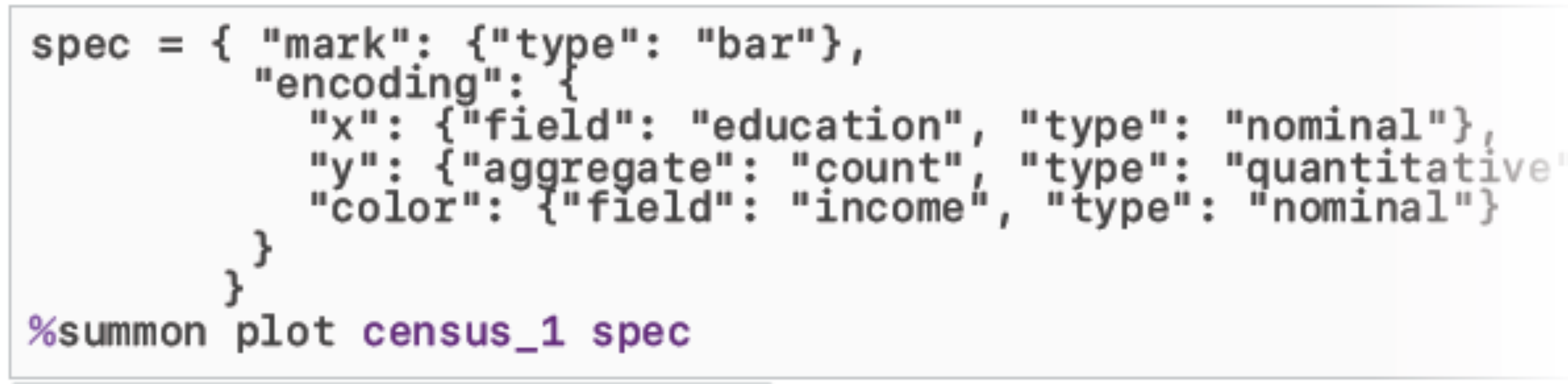
Plot □

03 Use Cases
Plot & Save

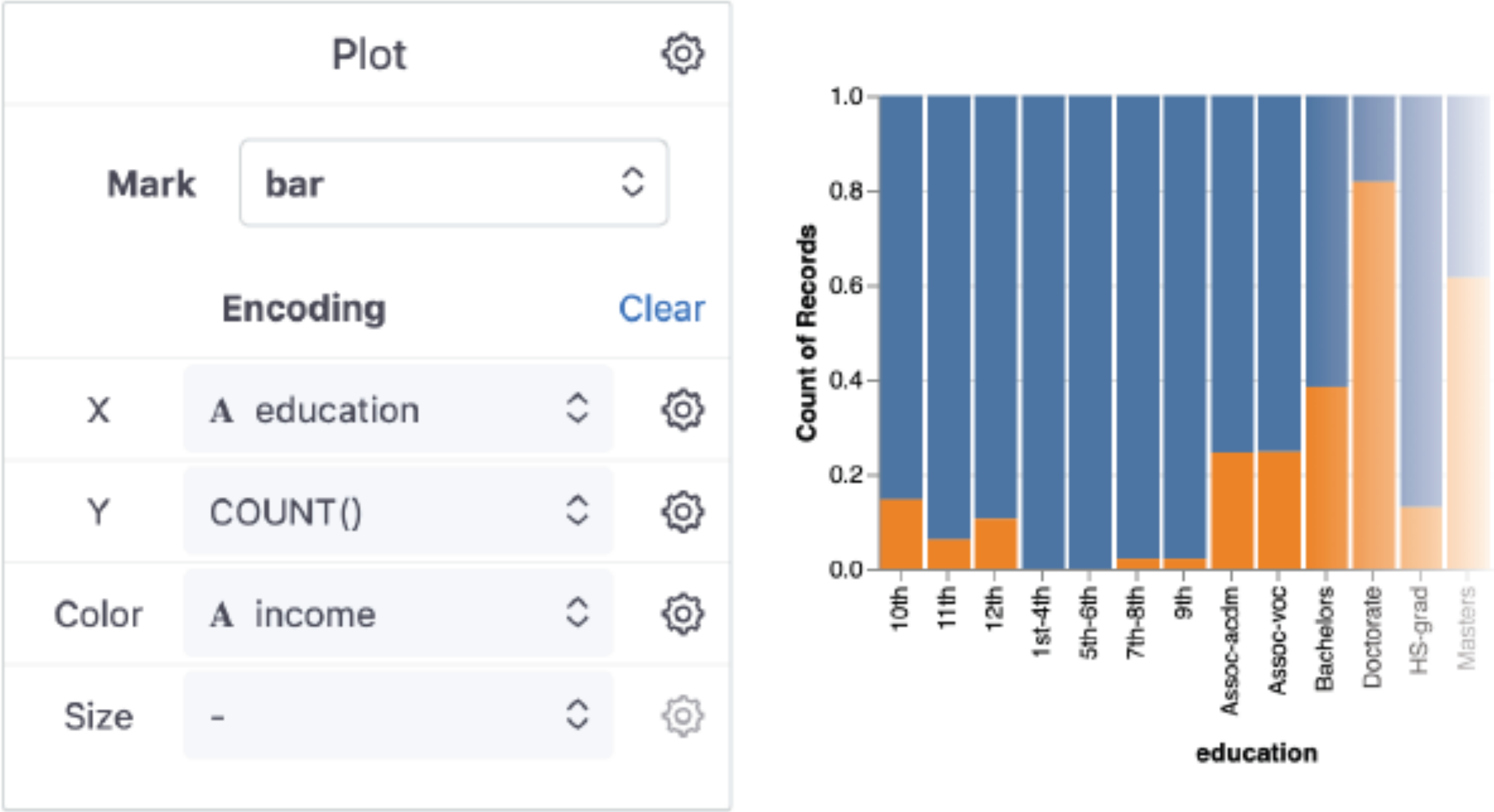
♻️ Interactive plotting and saving are supported

3️⃣ `plot`

visualizing data as charts
generate Vega-Lite JSON specifications



```
spec = { "mark": {"type": "bar"},
         "encoding": {
             "x": {"field": "education", "type": "nominal"},
             "y": {"aggregate": "count", "type": "quantitative"
             "color": {"field": "income", "type": "nominal"}
         }
       }
%summon plot census_1 spec
```



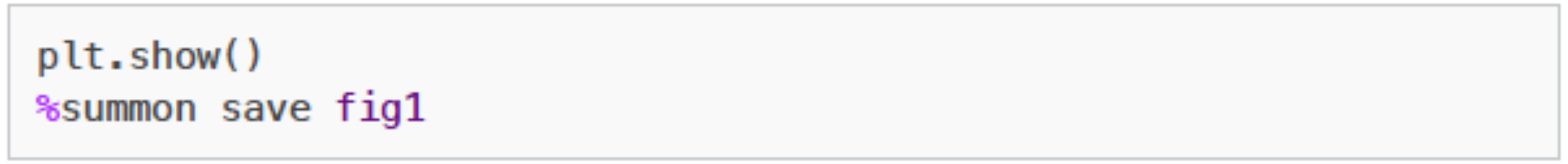
Mark: bar

Encoding

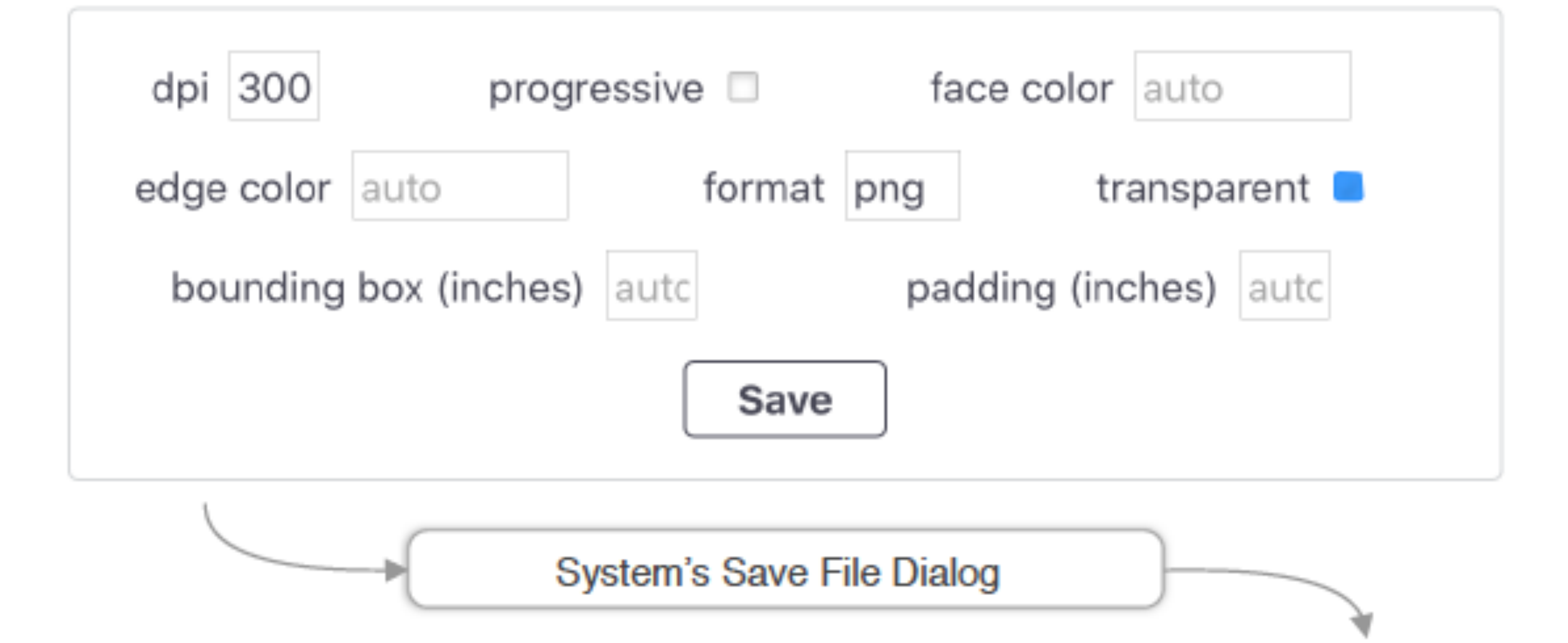
X: education
Y: COUNT()
Color: income
Size: -

4️⃣ `save`

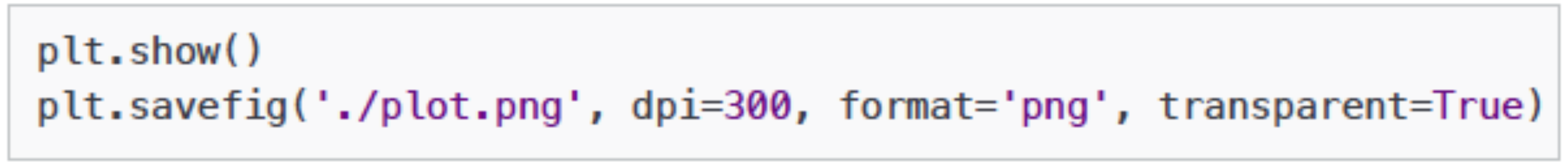
exporting (anything) to a file



```
plt.show()
%summon save fig1
```



dpi 300 progressive □ face color auto
edge color auto format png transparent ☑
bounding box (inches) auto padding (inches) auto



```
plt.show()
plt.savefig('./plot.png', dpi=300, format='png', transparent=True)
```

Berkeley
UNIVERSITY OF CALIFORNIA

03 Use Cases

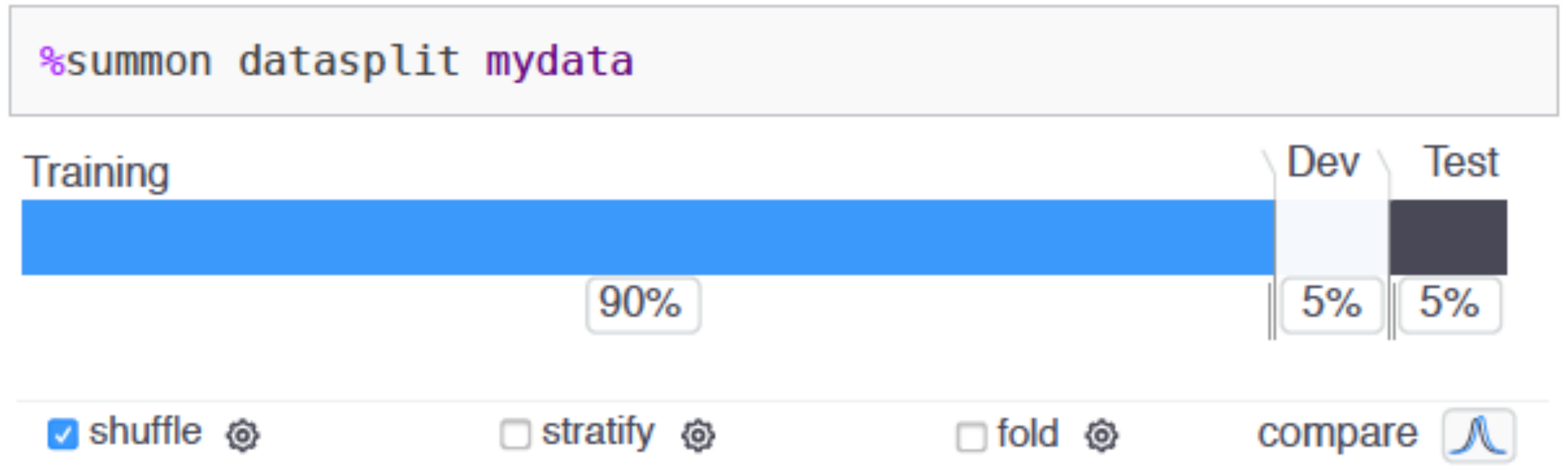
Datasplit & Confusion

🤖 Mage also works for machine learning tasks

5 `datasplit`

segmenting data for machine learning

visual version `train_test_split()` in `scikit-learn`

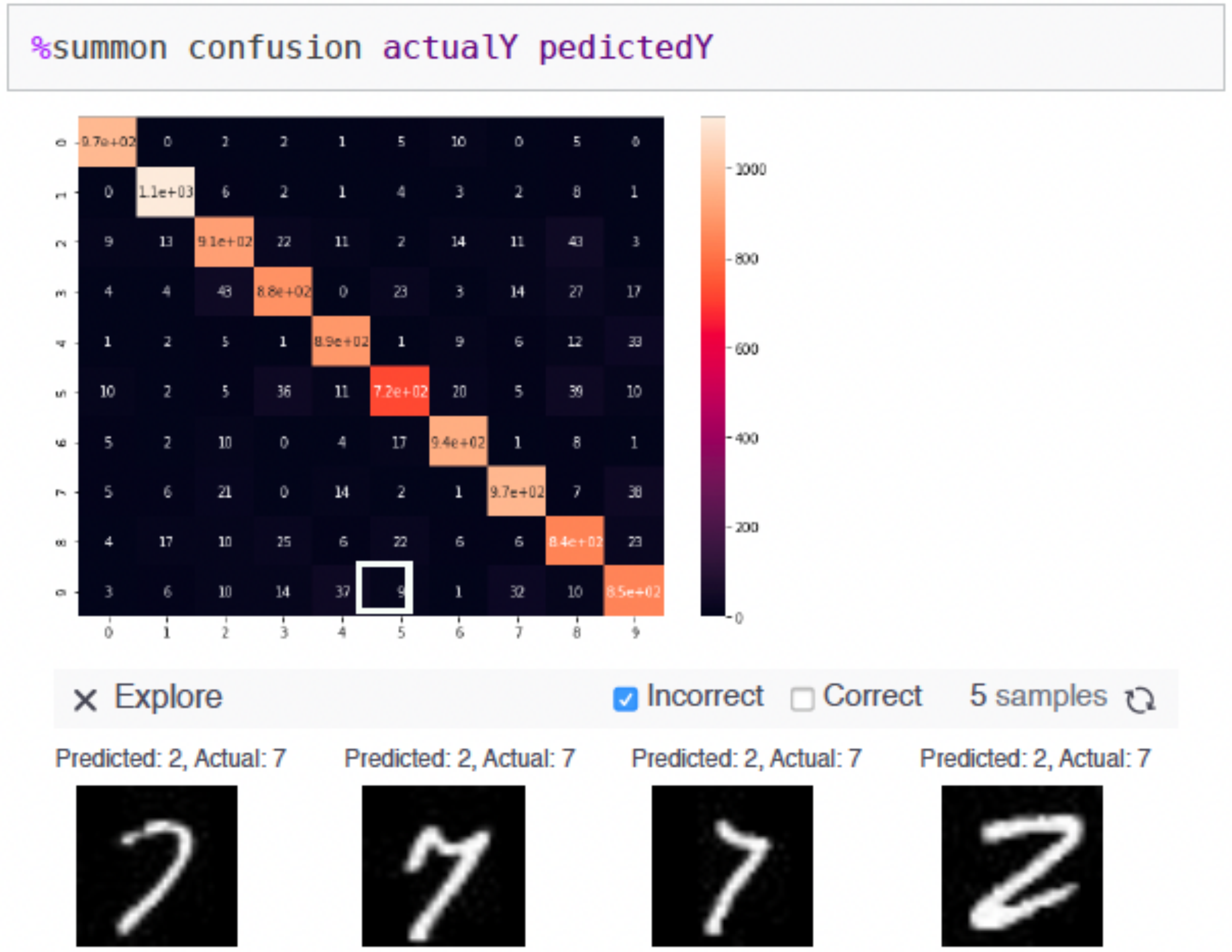


```
%summon datasplit mydata
```

☑ shuffle ⚙ □ stratify ⚙ □ fold ⚙ compare

6 `confusion`

exploring classifier model performance



```
%summon confusion actualY pedictedY
```

	0	1	2	3	4	5	6	7	8	9
0	9.7e+02	0	2	2	1	5	10	0	5	0
1	0	1.1e+03	6	2	1	4	3	2	8	1
2	9	13	9.1e+02	22	11	2	14	11	43	3
3	4	4	43	8.8e+02	0	23	3	14	27	17
4	1	2	5	1	8.9e+02	1	9	6	12	33
5	10	2	5	36	11	7.2e+02	20	5	39	10
6	5	2	10	0	4	17	9.4e+02	1	8	1
7	5	6	21	0	14	2	1	9.7e+02	7	38
8	4	17	10	25	6	22	6	6	8.4e+02	23
9	3	6	10	14	37	9	1	32	10	8.5e+02

✕ Explore ☑ Incorrect □ Correct 5 samples

Predicted: 2, Actual: 7

Predicted: 2, Actual: 7

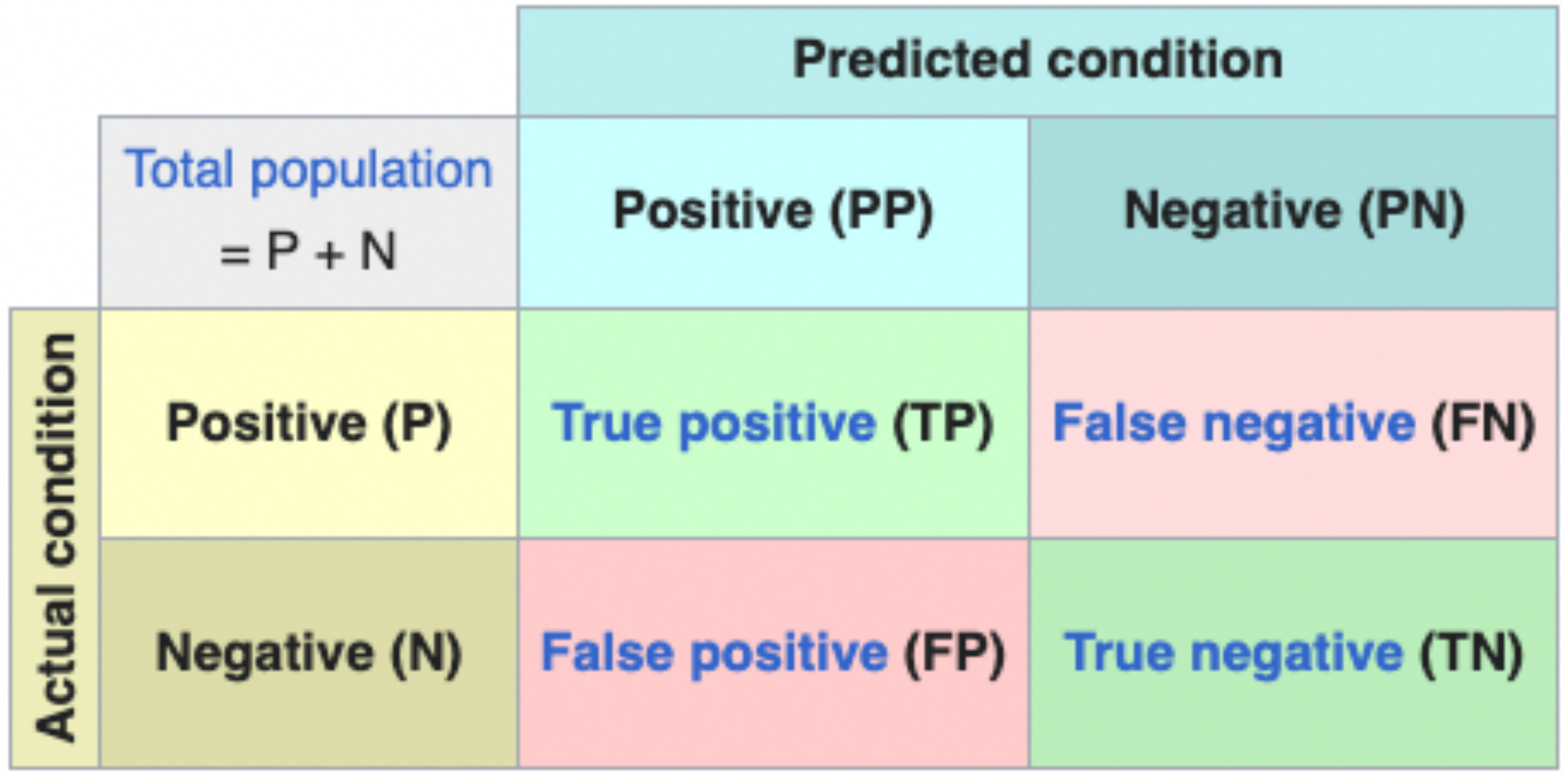
Predicted: 2, Actual: 7

Predicted: 2, Actual: 7

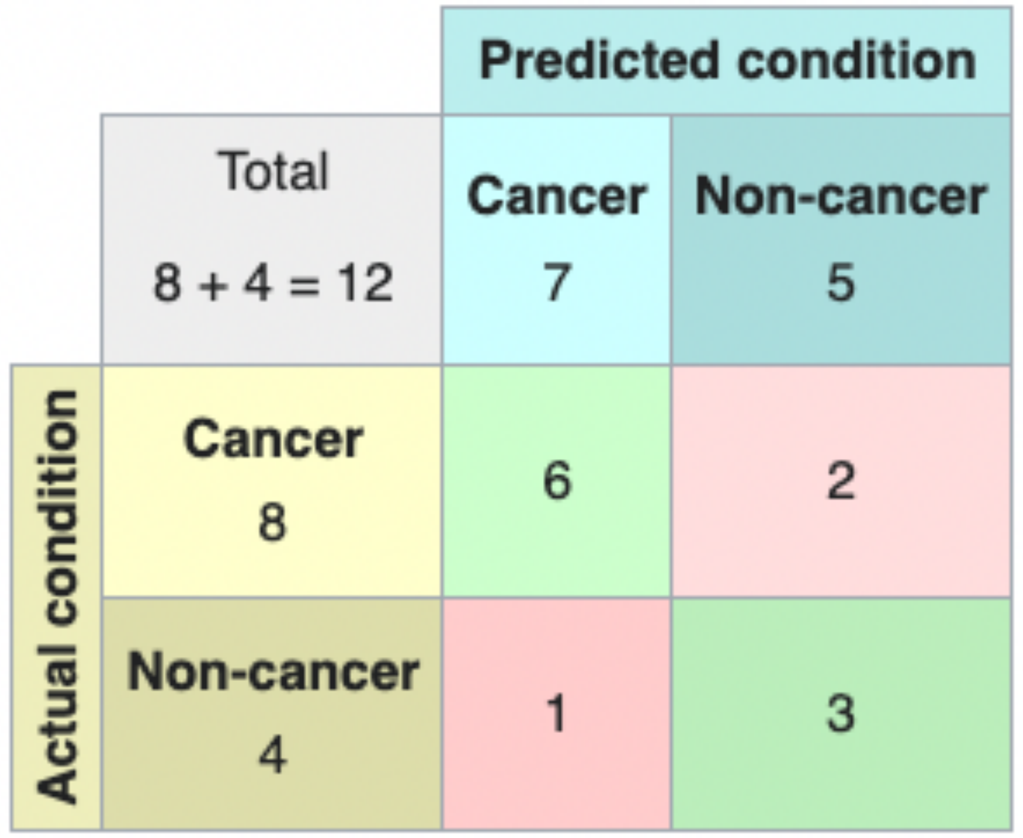
Berkeley
UNIVERSITY OF CALIFORNIA

👾 Side: Confusion matrix

In the field of machine learning and specifically the problem of statistical classification, a **confusion matrix** is a specific table layout that allows visualization of the performance of an algorithm, typically a supervised learning one.



		Predicted condition	
	Total population = P + N	Positive (PP)	Negative (PN)
Actual condition	Positive (P)	True positive (TP)	False negative (FN)
	Negative (N)	False positive (FP)	True negative (TN)



		Predicted condition	
	Total 8 + 4 = 12	Cancer 7	Non-cancer 5
Actual condition	Cancer 8	6	2
	Non-cancer 4	1	3

03 Use Cases

Multi-tool Senario

Move data between different modalities simpler

Background

An analyst is exploring the **1994 US Census dataset**

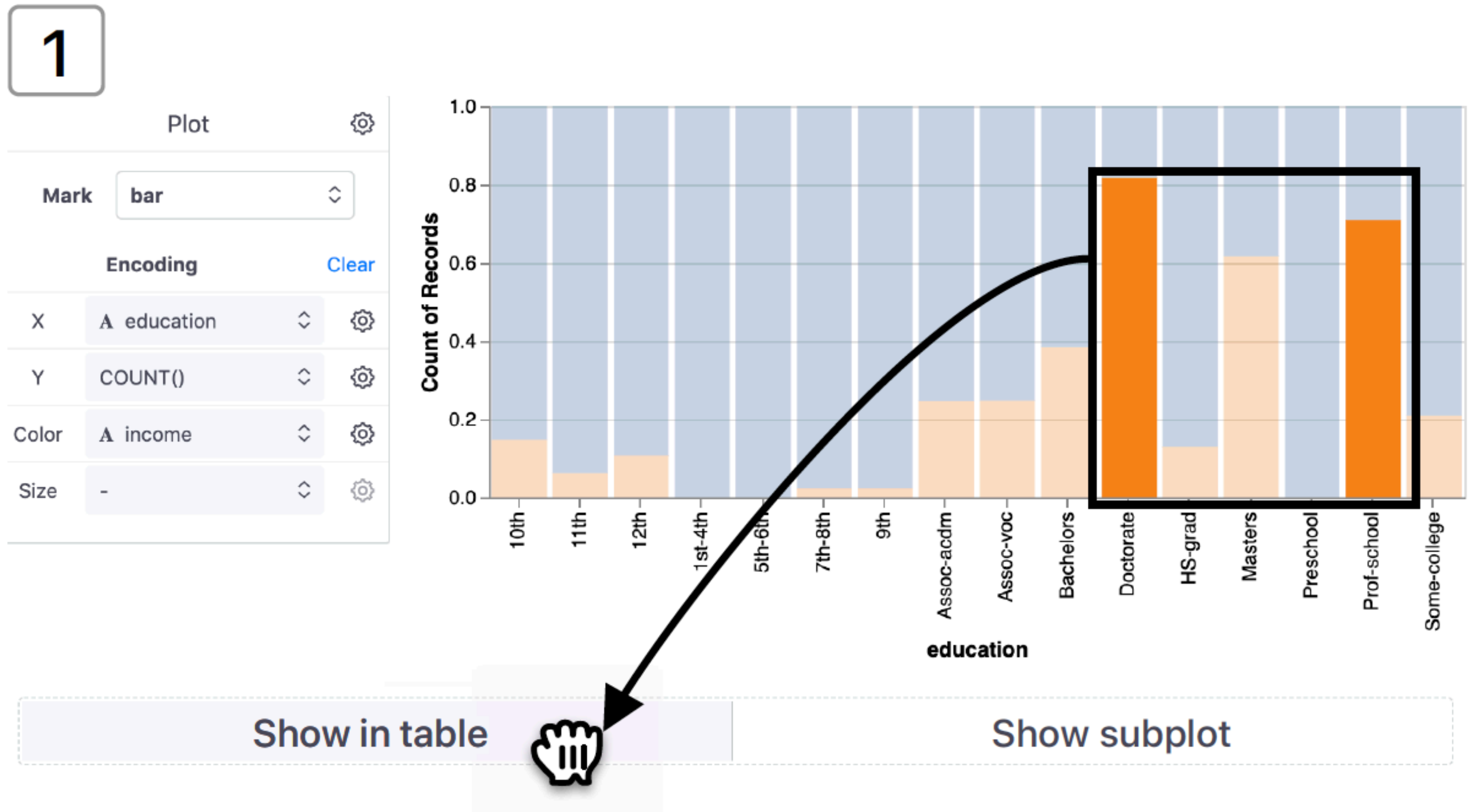
Goal

Investigate historical gender bias in high income and high education workers

Step

- `%summon plot census`
- adds “income” to he color channel and enables “normalize” stacking for the “Y” axis to compare the percentage of low vs. high income in each education level
- select the “>50k” income bars with “Doctorate” and “Prof school” education levels
- drag selection into “Show in table”, resulting in a new instance of the `table`

1



Plot

Mark: bar

Encoding — Clear

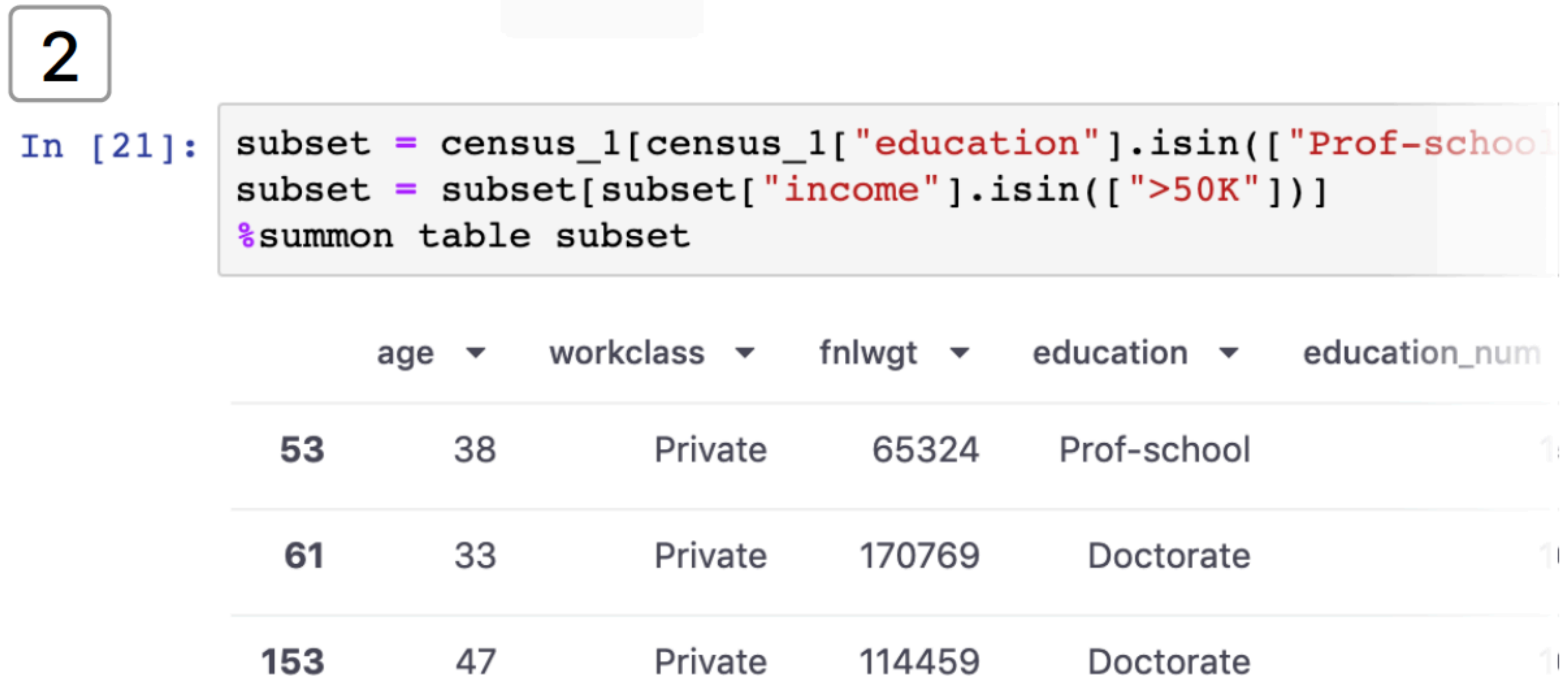
X: education

Y: COUNT()

Color: income

Size: -

2



```
In [21]: subset = census_1[census_1["education"].isin(["Prof-schoo
         subset = subset[subset["income"].isin([">50K"])]
         %summon table subset
```

	age	workclass	fnlwgt	education	education_num
53	38	Private	65324	Prof-school	
61	33	Private	170769	Doctorate	
153	47	Private	114459	Doctorate	

Berkeley
UNIVERSITY OF CALIFORNIA

04

Evaluation

04 Evaluation

Participants & Procedure

🙊 9 data practitioners evaluate the usefulness

Participants

👩‍💼 👨‍💼 👩‍💼 👨‍💼 👩‍💼 👨‍💼 👩‍💼 👨‍💼 👩‍💼

9 Industry practitioners

- ALL use notebooks and do professional data work
- MOST participants also worked with machine learning

Procedure

7 Starter tool ideas presentation

table	plot	image	confusion
api	datasplit	save	

🟧 Functioning tool in notebook ■ High-fidelity paper prototype

Usefulness + General impressions

❎ usability evaluation

😸 Most functionalities receive positive feedback

Automatically Generating Code 😐

Programming experience ⬆️

Preference toward generated code ⬇️

Negative correlation

Visual Data Selection 😍

All participants wanted easier ways to select data
and use direct manipulation to pull a selection into code

Interact to Explore Model Performance 😍

Better understand model performance without switch contexts

Make Good Practices Easier 😍

Ability to compare distributions in `datasplit`

05

Future Work

Berkeley
UNIVERSITY OF CALIFORNIA

05 Future Work
Improvements

🔛 3 major improvements are proposed by the authors

Generated Code Quality

1️⃣ Inserts a new column at one position

2️⃣ Repositions it

👎 Current generated code

```
df.insert(14, "educ", dat)
column_names = list(df)
column_names.pop(14)
column_names.insert(2, "educ")
df = df.reindex(columns=column_names)
%summon table df
```

👍 Improved generated code

```
df.insert(2, "educ", dat)
%summon table df
```

Movement Between GUI Tools

- Transfer content between two GUI tools
- Not just between GUI and code
- E.g. coordinated visualizations, like an impromptu data dashboard

Make Tool Builder’s life Easier

Tool-specific understanding is needed from tool builders