



ProvenanceWidgets

A JavaScript Library of UI Controls
to Track and Dynamically Overlay Analytic Provenance



arpit narechania



kaustubh odak



mennatallah el-assady



alex endert

range slider

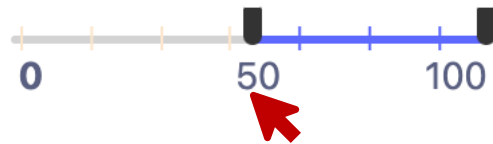


range slider



Initial State

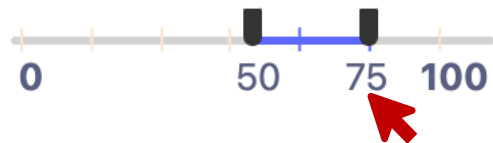
range slider



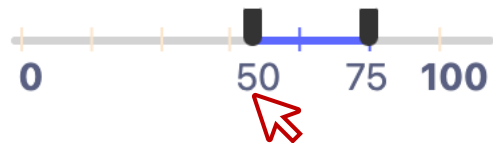
range slider



range slider



range slider



range slider



range slider



What can I tell about my interactions thus far?

range slider



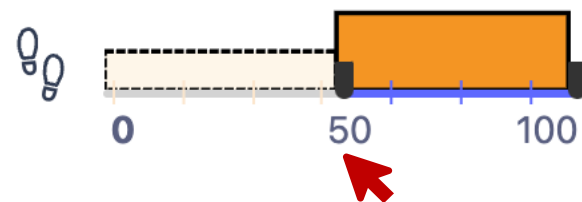
ProvenanceWidgets range slider



range slider



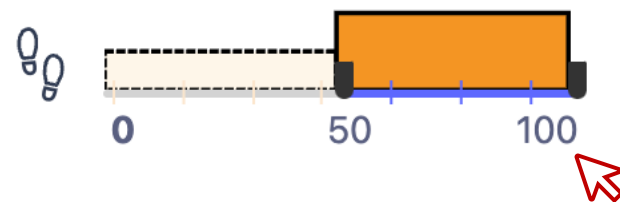
ProvenanceWidgets range slider



range slider



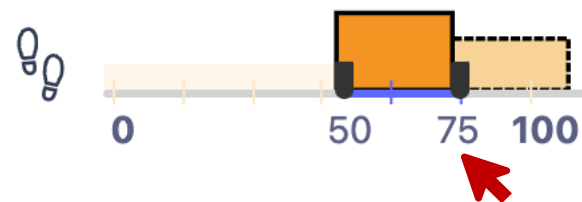
ProvenanceWidgets range slider



range slider



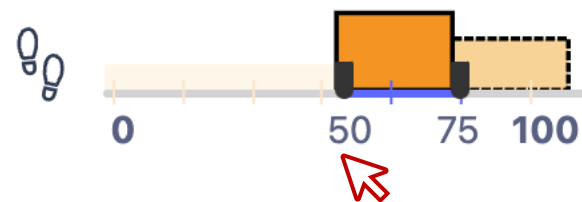
ProvenanceWidgets range slider



range slider



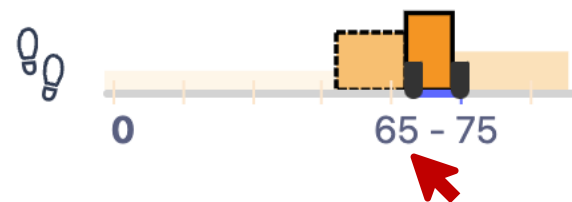
ProvenanceWidgets range slider



range slider



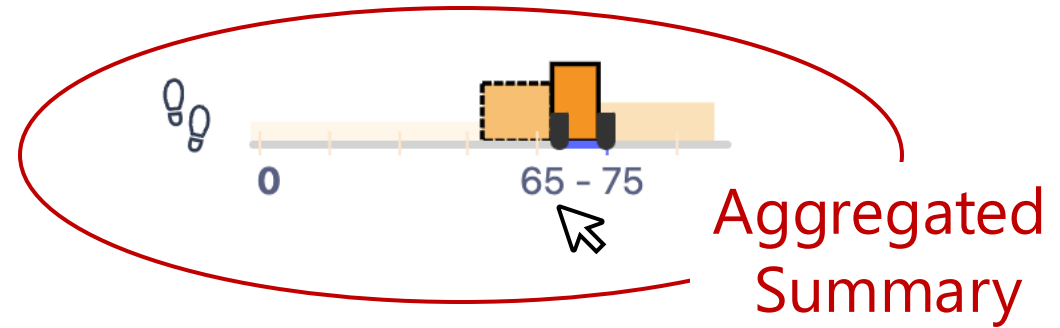
ProvenanceWidgets range slider

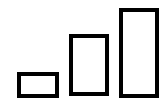


range slider



ProvenanceWidgets range slider



 height = frequency

 color = recency

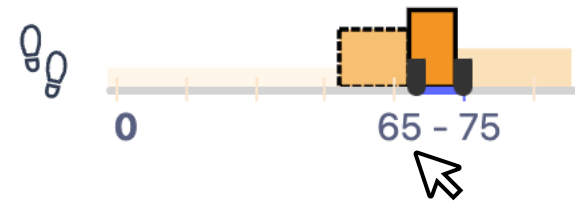
 solid border = current selection

 dashed border = previous selection

range slider



ProvenanceWidgets range slider



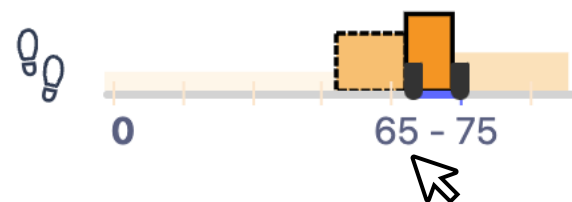
Can I review the historical sequence of my selections?

range slider

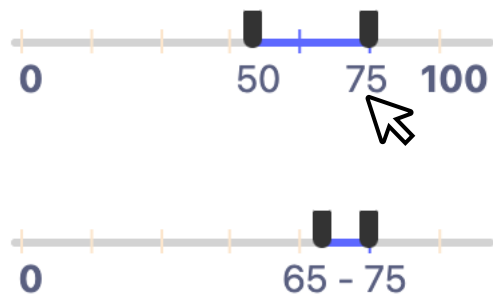


current
state

ProvenanceWidgets range slider

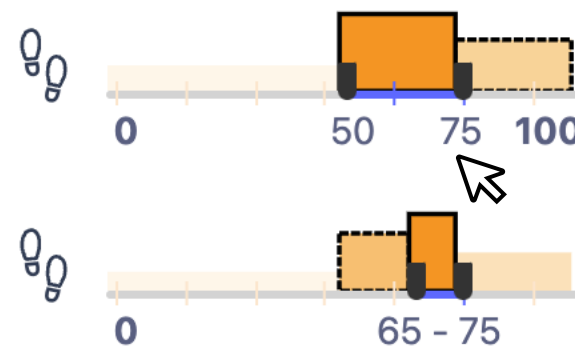


range slider

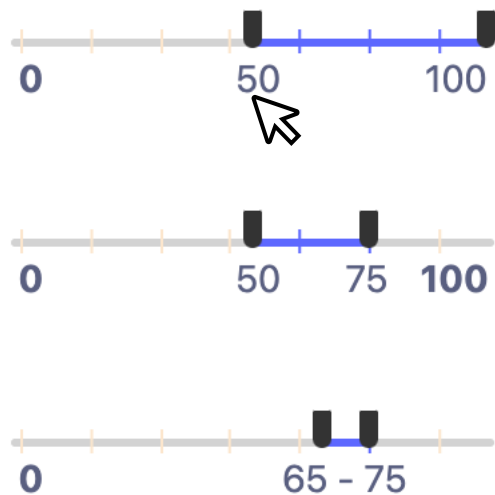


current
state

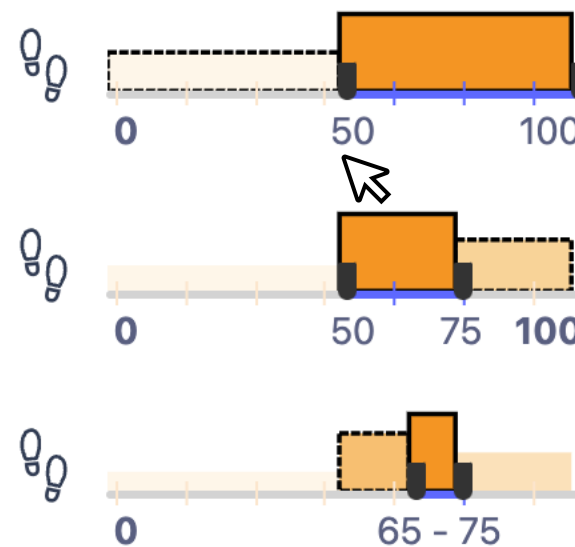
ProvenanceWidgets range slider



range slider

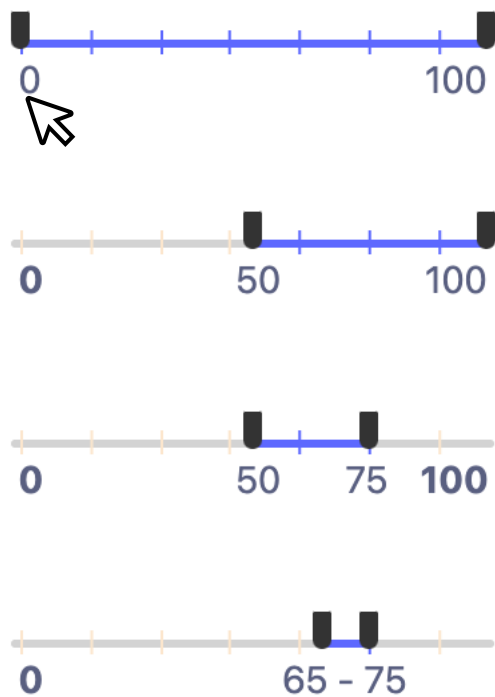


ProvenanceWidgets range slider



current
state

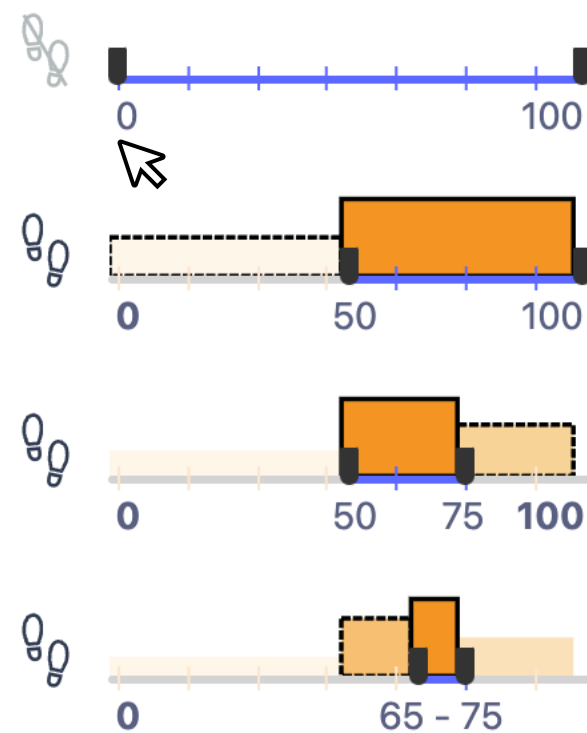
range slider



initial
state

current
state

ProvenanceWidgets range slider

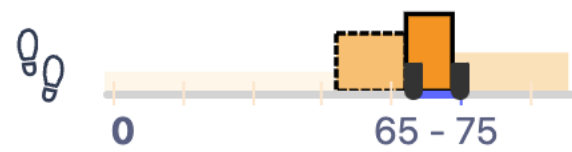


range slider



ProvenanceWidgets range slider

current
state

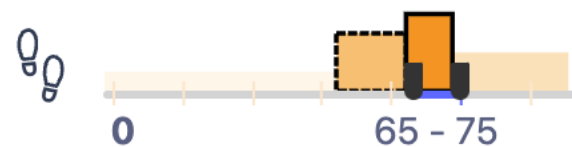


range slider



current
state

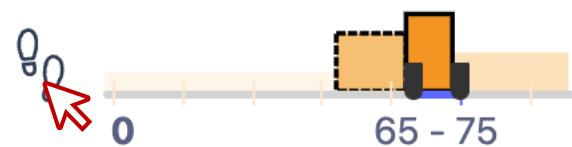
ProvenanceWidgets range slider



range slider



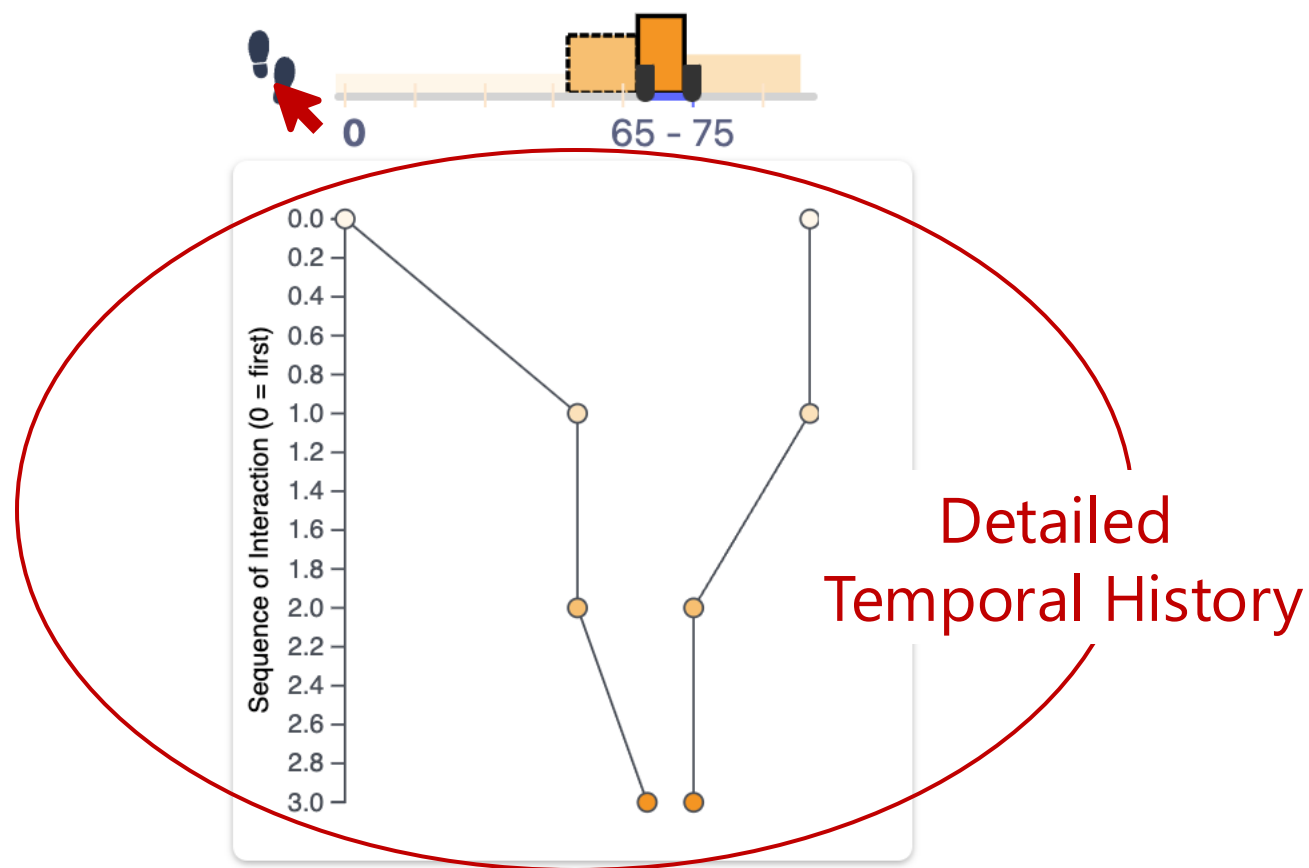
ProvenanceWidgets range slider



range slider



ProvenanceWidgets range slider



range slider



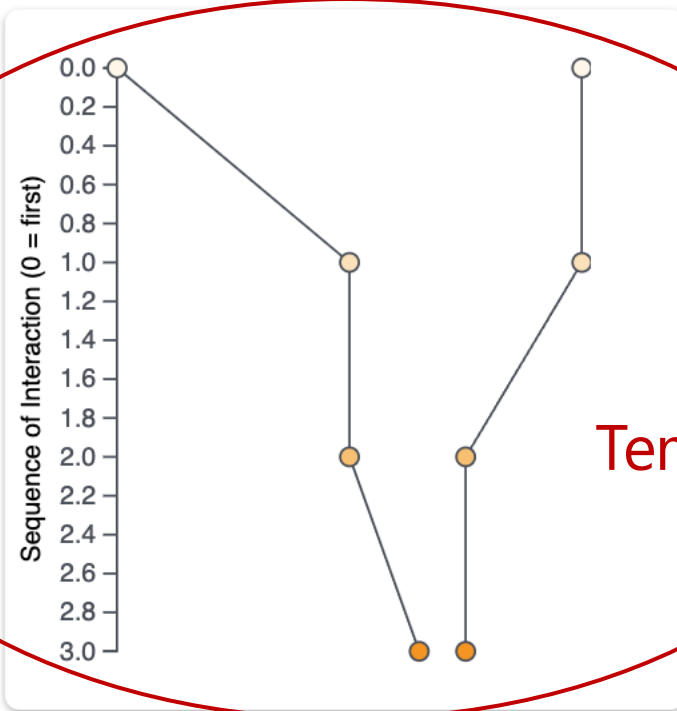
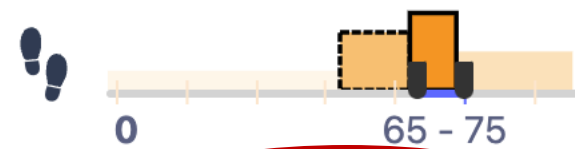
slider handle state = ○

recency = ● ● ●

initial state
[0, 100]

current state
[65, 75]

ProvenanceWidgets range slider

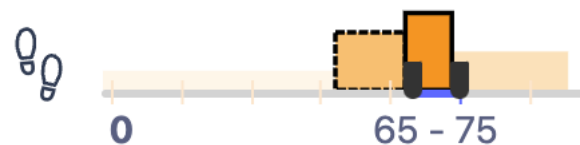


Detailed
Temporal History

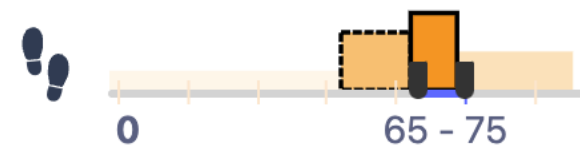
ProvenanceWidgets range slider



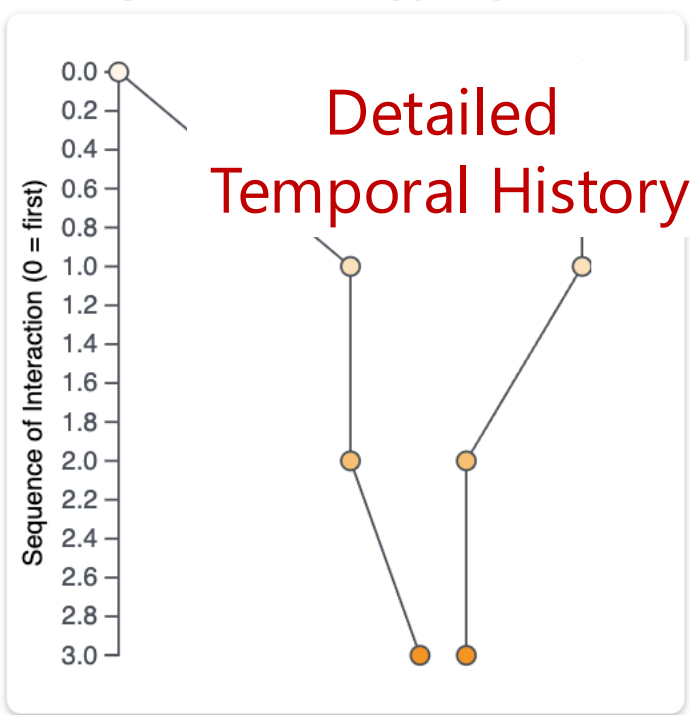
Initial State



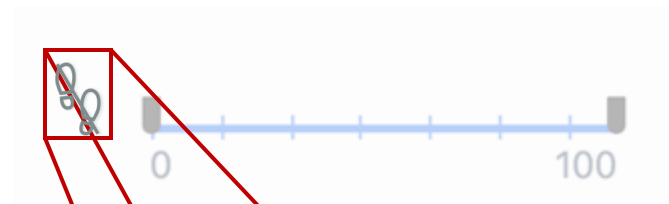
Aggregated Summary



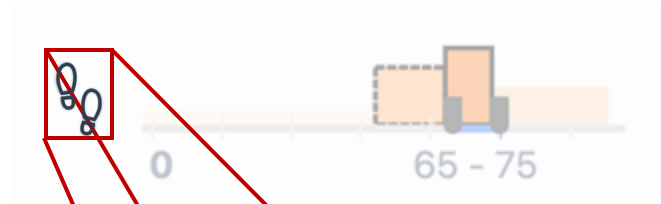
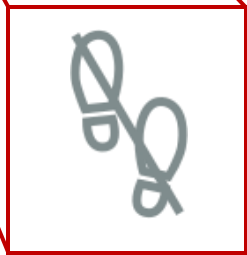
Detailed Temporal History



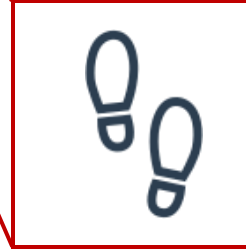
ProvenanceWidgets range slider



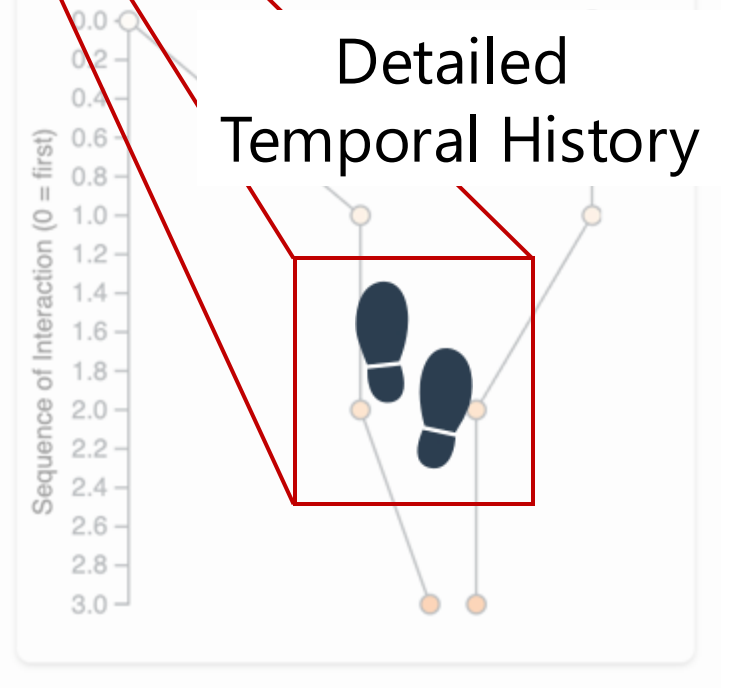
Initial State



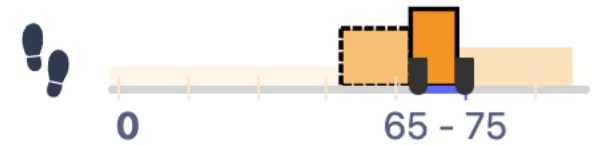
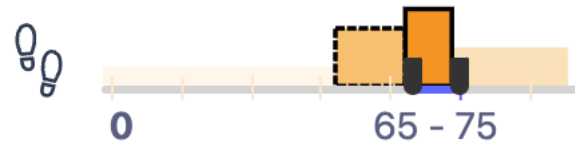
Aggregated Summary



Detailed Temporal History

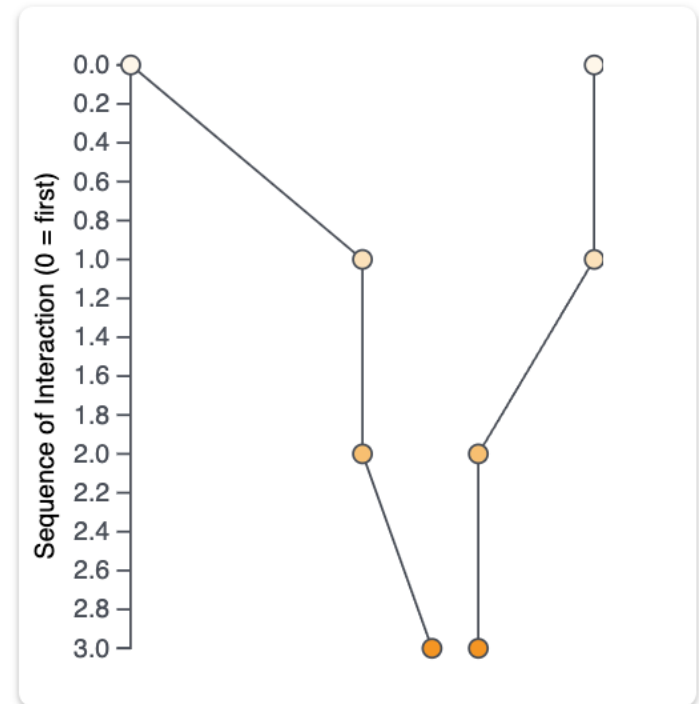


ProvenanceWidgets range slider

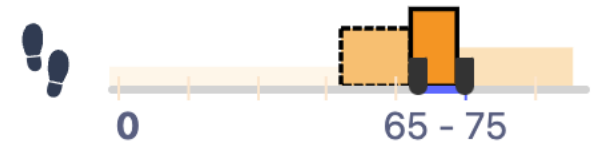
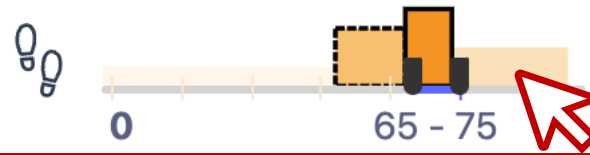


Hover a bar / point =

Tooltip for additional
details

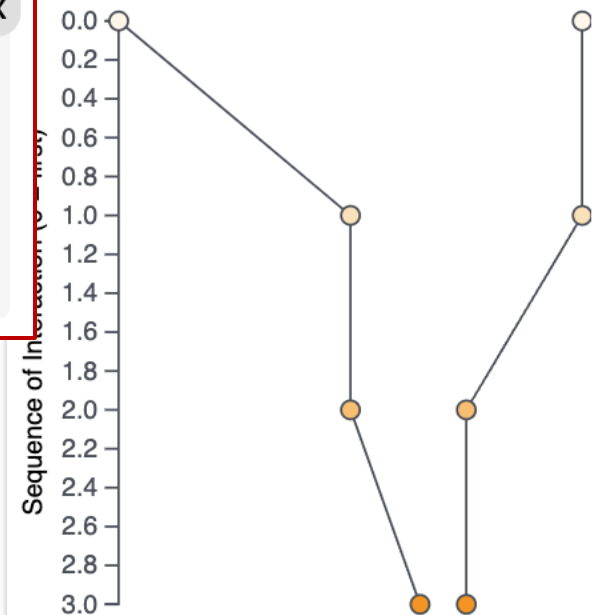


ProvenanceWidgets range slider

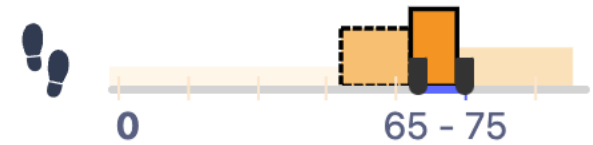
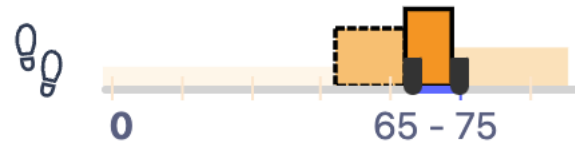


Hover a bar / point =
Tooltip for additional
details

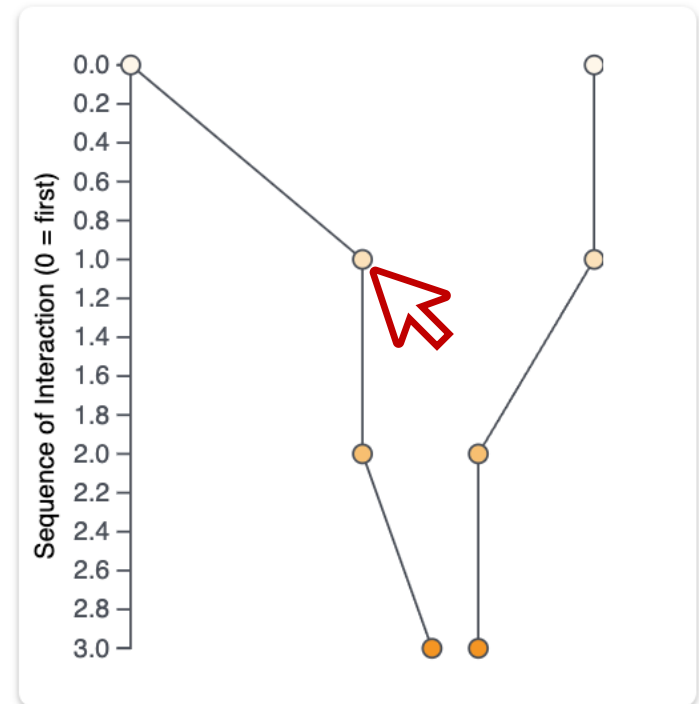
Label: id-rangeSlider
Selected range: [75, 100]
times selected: 2
Last selected at: 9/24/2024, 6:07:09 PM
1st/3rd selection



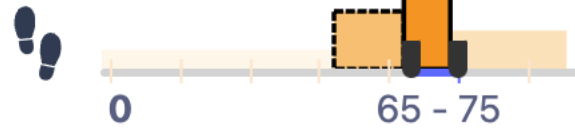
ProvenanceWidgets range slider



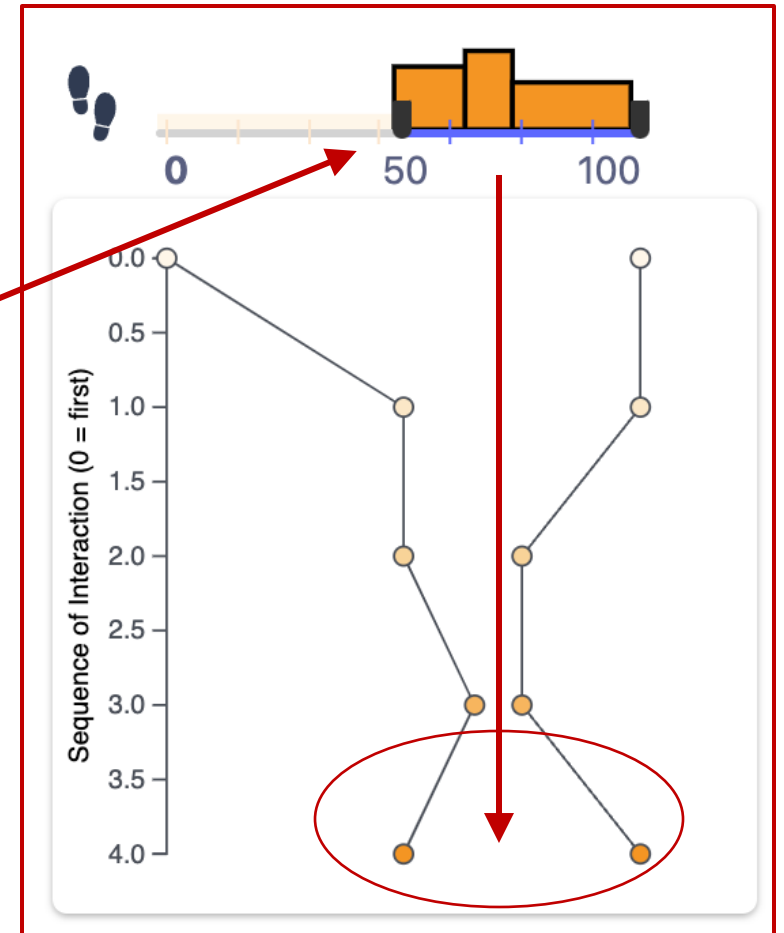
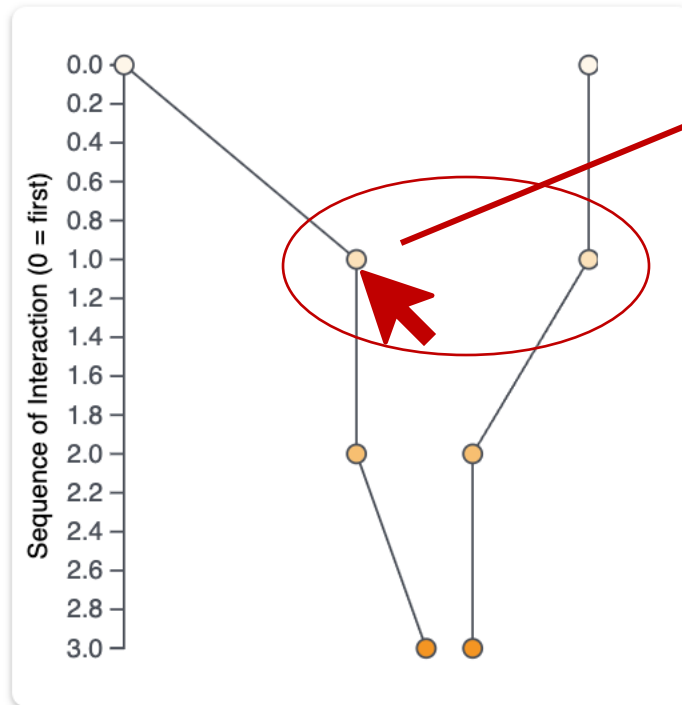
Click a bar / point =
Selects that range and
adds a new
provenance state



ProvenanceWidgets range slider



Click a bar / point =
Selects that range and
adds a new
provenance state



Single Slider



Range Slider



Radiobutton Group

- 
- ☒ New York
 - ☐ Rome
 - ☐ London
 - ☐ Istanbul
 - ☐ Paris

Checkbox Group

- 
- ☒ New York
 - ☒ Rome
 - ☐ London
 - ☐ Istanbul
 - ☐ Paris

Single Select Dropdown



New York × ▼

Multi Select Dropdown



New York, Rome × ▼

Input Text



motivation

no frontend library of UI controls

that tracks and presents *provenance out-of-the-box*.

design process

review UI controls

conduct design exercises

finalize designs

design process

review UI controls

conduct design exercises

finalize designs

design process

review UI controls

- structure

- layout

- initial/default values

- subsequent values

- interaction events

- conduct design exercises

- finalize designs

design process

review UI controls structure

layout

initial/default values

subsequent values

interaction events

conduct design exercises

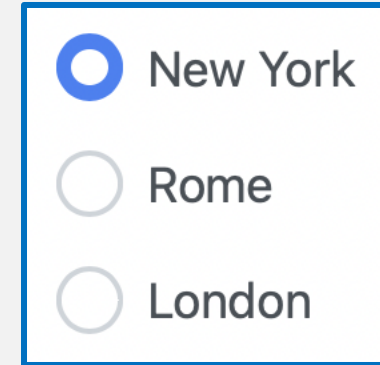
finalize designs

design process

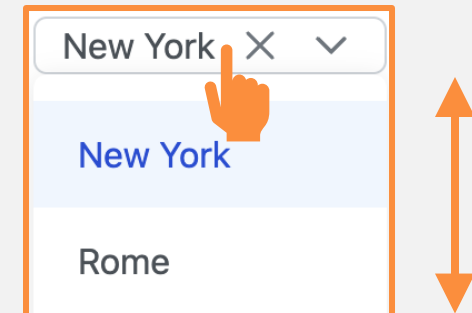
review UI controls
structure

radio buttons are *always completely visible*,
unlike **dropdowns**, which require an additional click.

layout
initial/default values
subsequent values
interaction events
conduct design exercises
finalize designs



A UI control showing three radio buttons stacked vertically. The first button, labeled 'New York', is selected and has a blue outline. The second button, labeled 'Rome', and the third button, labeled 'London', are unselected and have gray outlines.



A UI control showing a dropdown menu. The top bar displays 'New York' with a close button (X) and a dropdown arrow (v). Below the bar, a list of options is shown: 'New York' (highlighted in light blue) and 'Rome'. An orange hand icon points to the 'New York' option in the list. A vertical double-headed arrow is positioned to the right of the dropdown menu.

design process

review UI controls structure

layout

initial/default values

subsequent values

interaction events

conduct design exercises

finalize designs

design process

review UI controls

structure

layout

initial/default values

subsequent values

interaction events

conduct design exercises

finalize designs

design process

review UI controls

structure

layout

sliders can be oriented vertically or horizontally,
unlike **dropdowns**, which are always oriented horizontally.

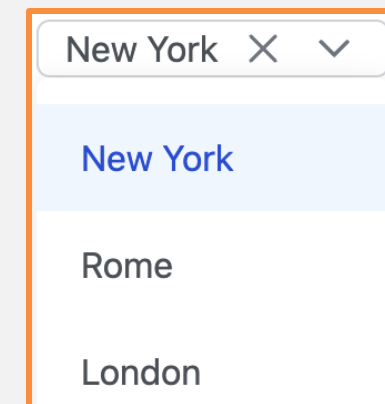
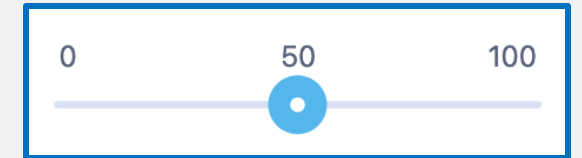
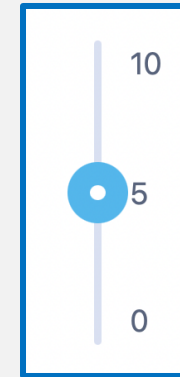
initial/default values

subsequent values

interaction events

conduct design exercises

finalize designs



design process

review UI controls

structure

layout

initial/default values

subsequent values

interaction events

conduct design exercises

finalize designs

design process

review UI controls

structure

layout

initial/default values

subsequent values

interaction events

conduct design exercises

finalize designs

design process

review UI controls

structure

layout

initial/default values

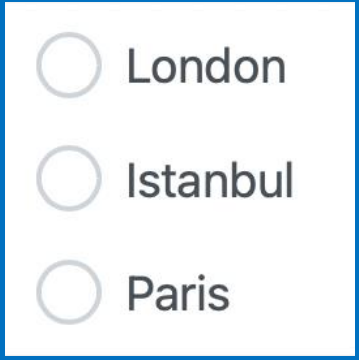
radio buttons can have an *uninitialized* state,
unlike **sliders**, which always have a *default* value.

subsequent values

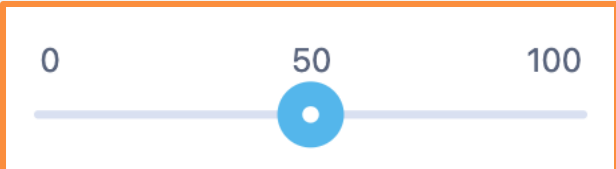
interaction events

conduct design exercises

finalize designs



A vertical list of three radio buttons, each with a light gray circular outline to its left. The text labels are 'London', 'Istanbul', and 'Paris' from top to bottom. The entire list is enclosed in a blue rectangular border.



A horizontal slider control. It features a light blue horizontal track with a blue circular knob in the center. Above the track, the numbers '0', '50', and '100' are displayed at the left, center, and right positions respectively. The entire slider is enclosed in an orange rectangular border.

design process

review UI controls

structure

layout

initial/default values

subsequent values

interaction events

conduct design exercises

finalize designs

design process

review UI controls

structure

layout

initial/default values

subsequent values

interaction events

conduct design exercises

finalize designs

design process

review UI controls

structure

layout

initial/default values

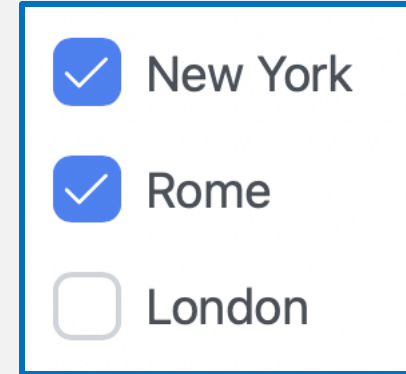
subsequent values

checkboxes can have multiple selected values,
unlike **radio buttons**, which can have at most one.

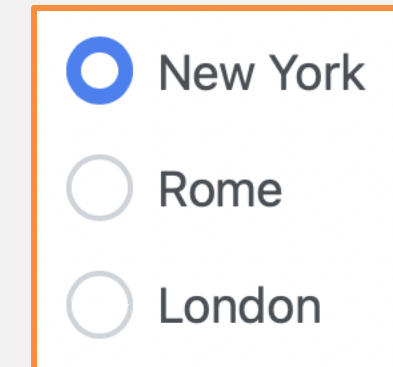
interaction events

conduct design exercises

finalize designs



☒	New York
☒	Rome
☐	London



☒	New York
☐	Rome
☐	London

design process

review UI controls

structure

layout

initial/default values

subsequent values

interaction events

conduct design exercises

finalize designs

design process

review UI controls

- structure

- layout

- initial/default values

- subsequent values

interaction events

- conduct design exercises

- finalize designs

design process

review UI controls

structure

layout

initial/default values

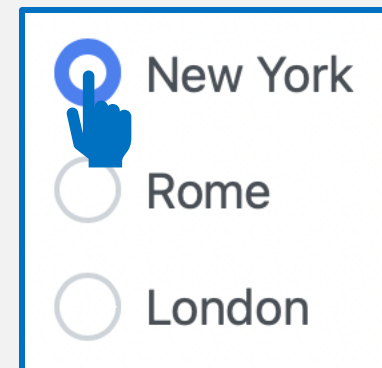
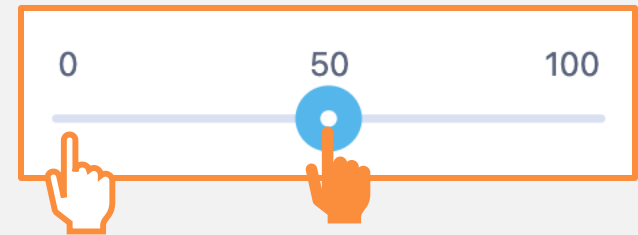
subsequent values

interaction events

drag handles / click on rails for **sliders**,
click an option for **radio buttons**.

conduct design exercises

finalize designs



design process

review UI controls

- structure

- layout

- initial/default values

- subsequent values

interaction events

- conduct design exercises

- finalize designs

design process

review UI controls

conduct design exercises

finalize designs

design process

review UI controls

conduct design exercises

finalize designs

design process

review UI controls

conduct design exercises

what metrics to log as provenance?

where to present provenance?

how to present provenance?

when to log provenance?

finalize designs

design process

review UI controls

conduct design exercises

what metrics to log as provenance?

where to present provenance?

how to present provenance?

when to log provenance?

finalize designs

design process

review UI controls

conduct design exercises

what metrics to log as provenance?

frequency and **recency** of selections
(i.e., state-changes)

where to present provenance?

how to present provenance?

when to log provenance?

finalize designs

t=0

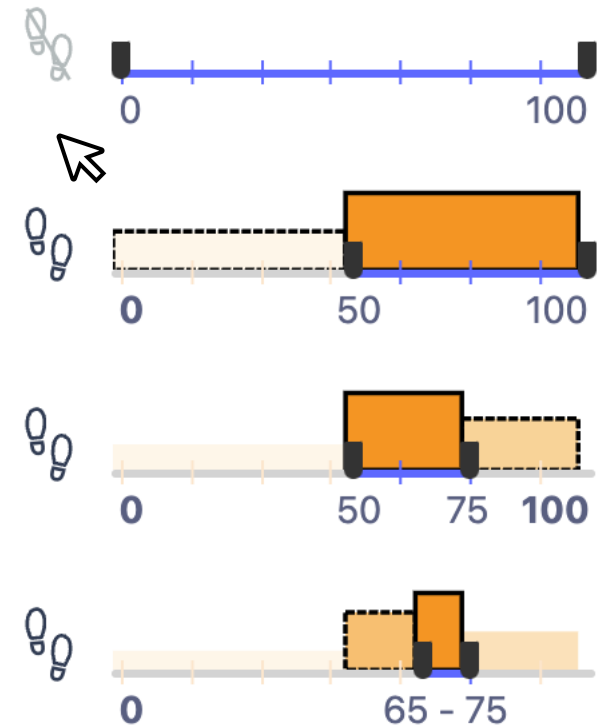
n=0



t=now()

n=3

Range Slider



design process

review UI controls

conduct design exercises

what metrics to log as provenance?

where to present provenance?

how to present provenance?

when to log provenance?

finalize designs

design process

review UI controls

conduct design exercises

what metrics to log as provenance?

where to present provenance?

how to present provenance?

when to log provenance?

finalize designs

design process

review UI controls

conduct design exercises

what metrics to log as provenance?

where to present provenance?

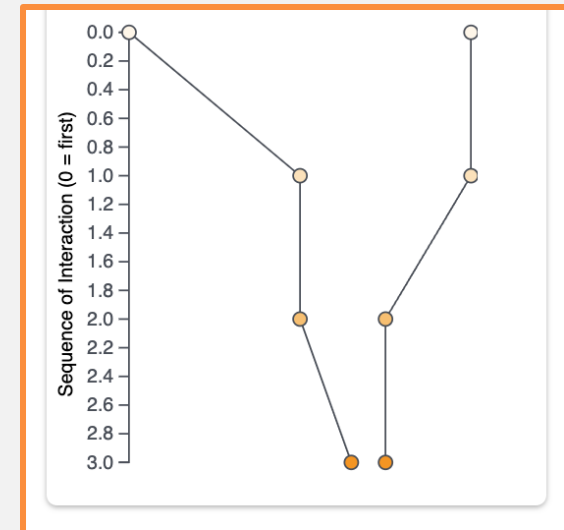
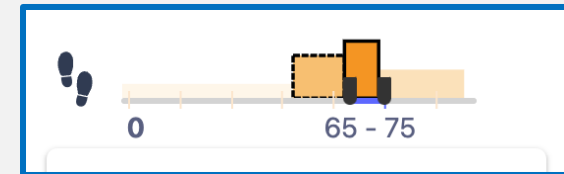
always visible near/on/around widget?
on-demand, in an external view?

how to present provenance?

when to log provenance?

finalize designs

Range Slider



design process

review UI controls

conduct design exercises

what metrics to log as provenance?

where to present provenance?

how to present provenance?

when to log provenance?

finalize designs

design process

review UI controls

conduct design exercises

what metrics to log as provenance?

where to present provenance?

how to present provenance?

when to log provenance?

finalize designs

design process

review UI controls

conduct design exercises

what metrics to log as provenance?

where to present provenance?

how to present provenance?

marks: area, bar, line?

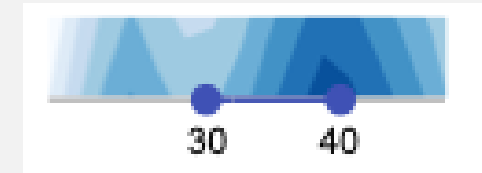
encodings: color, size, position?

when to log provenance?

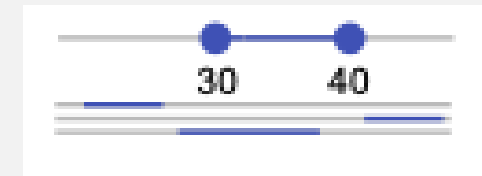
finalize designs

Range Slider

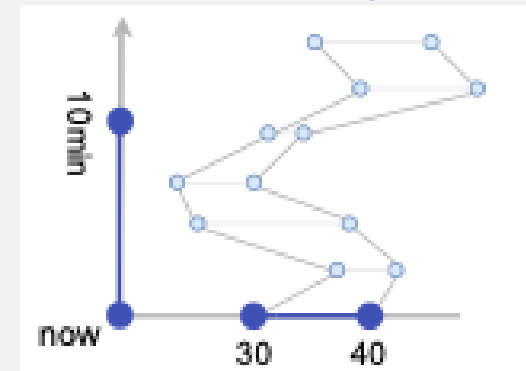
area/horizon, **color/y**



bar, **size**



line, **x/y**



design process

review UI controls

conduct design exercises

what metrics to log as provenance?

where to present provenance?

how to present provenance?

when to log provenance?

finalize designs

design process

review UI controls

conduct design exercises

what metrics to log as provenance?

where to present provenance?

how to present provenance?

when to log provenance?

finalize designs

design process

review UI controls

conduct design exercises

what metrics to log as provenance?

where to present provenance?

how to present provenance?

when to log provenance?

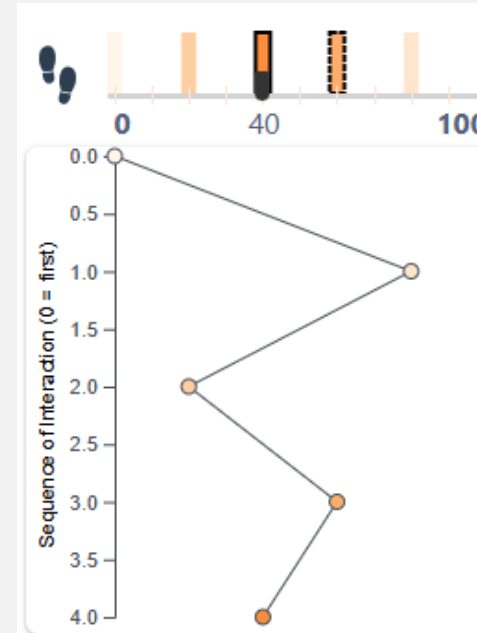
every **interaction (n)**?

every **second (t)**?

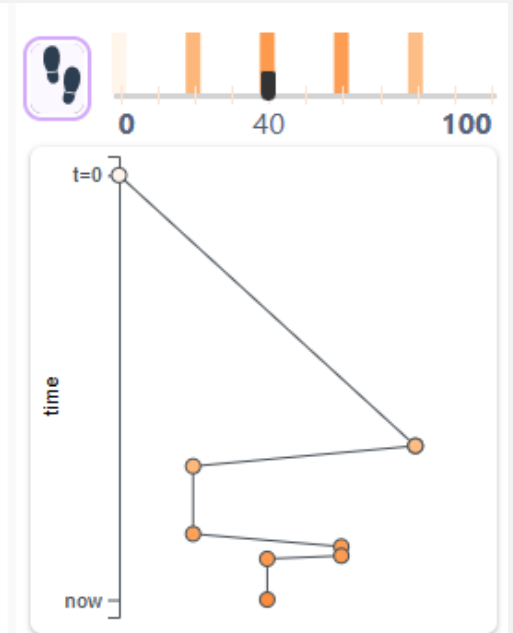
finalize designs

Range Slider analytic time “modes”

“interaction”



“time”



design process

review UI controls

conduct design exercises

what metrics to log as provenance?

where to present provenance?

how to present provenance?

when to log provenance?

finalize designs

design process

review UI controls

conduct design exercises

finalize designs

design process

review UI controls

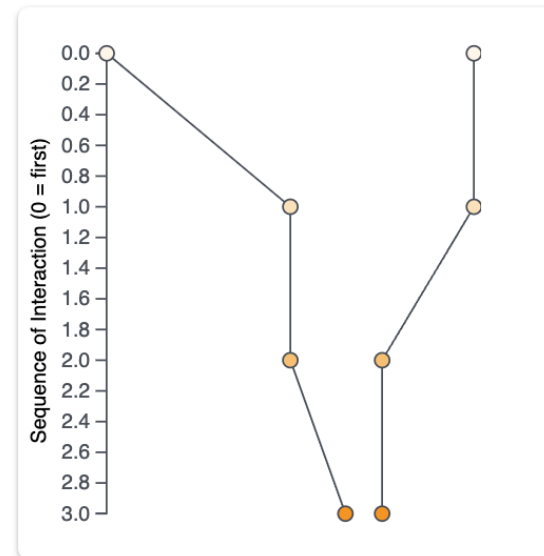
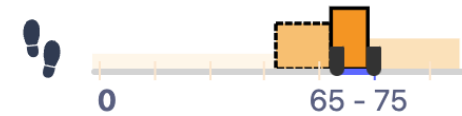
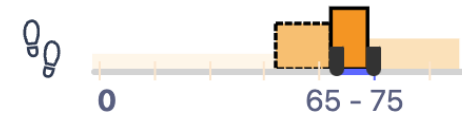
conduct design exercises

finalize designs

by balancing for

- utility
- simplicity
- aesthetics/clutter
- consistency
- ease of development

Range Slider

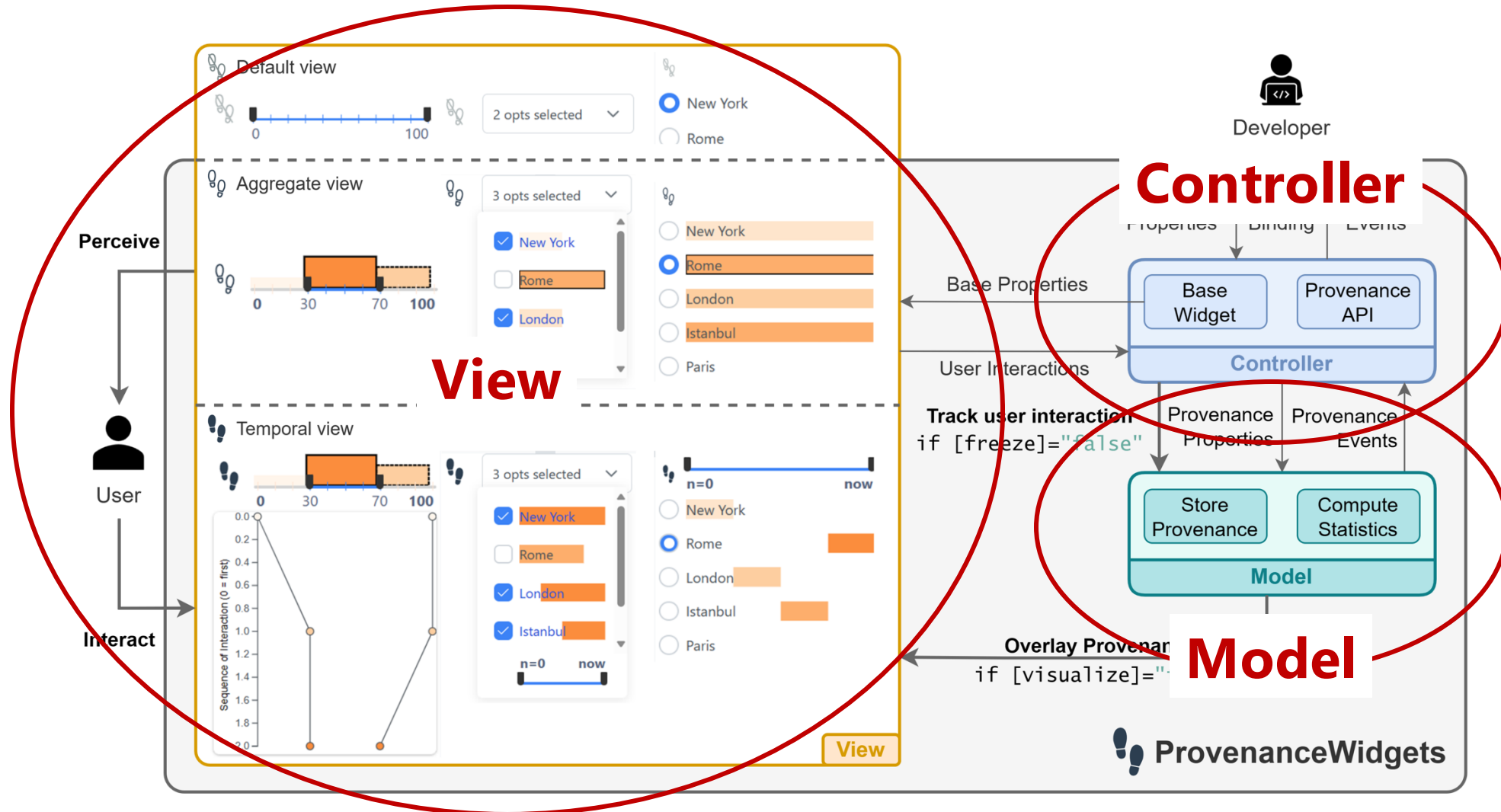


implementation

built on top of



¹² ngx-slider



implementation – range slider



```
1 value: number = 0
2 highValue?: number = 100 // Omit for Single Slider
3 handleChange(event: ChangeContext) {
4     value = event.value
5     highValue = event.highValue }
6 options: Options = { floor: 0, ceil: 100, step: 1 }
```

```
1 <provenance-slider [options]="options"
2     [value]="value" [highValue]="highValue"
3     (selectedChange)="handleChange($event)" />
```

range variables

event listener

other options

DOM element

use-cases

widgets to visualization (outbound)

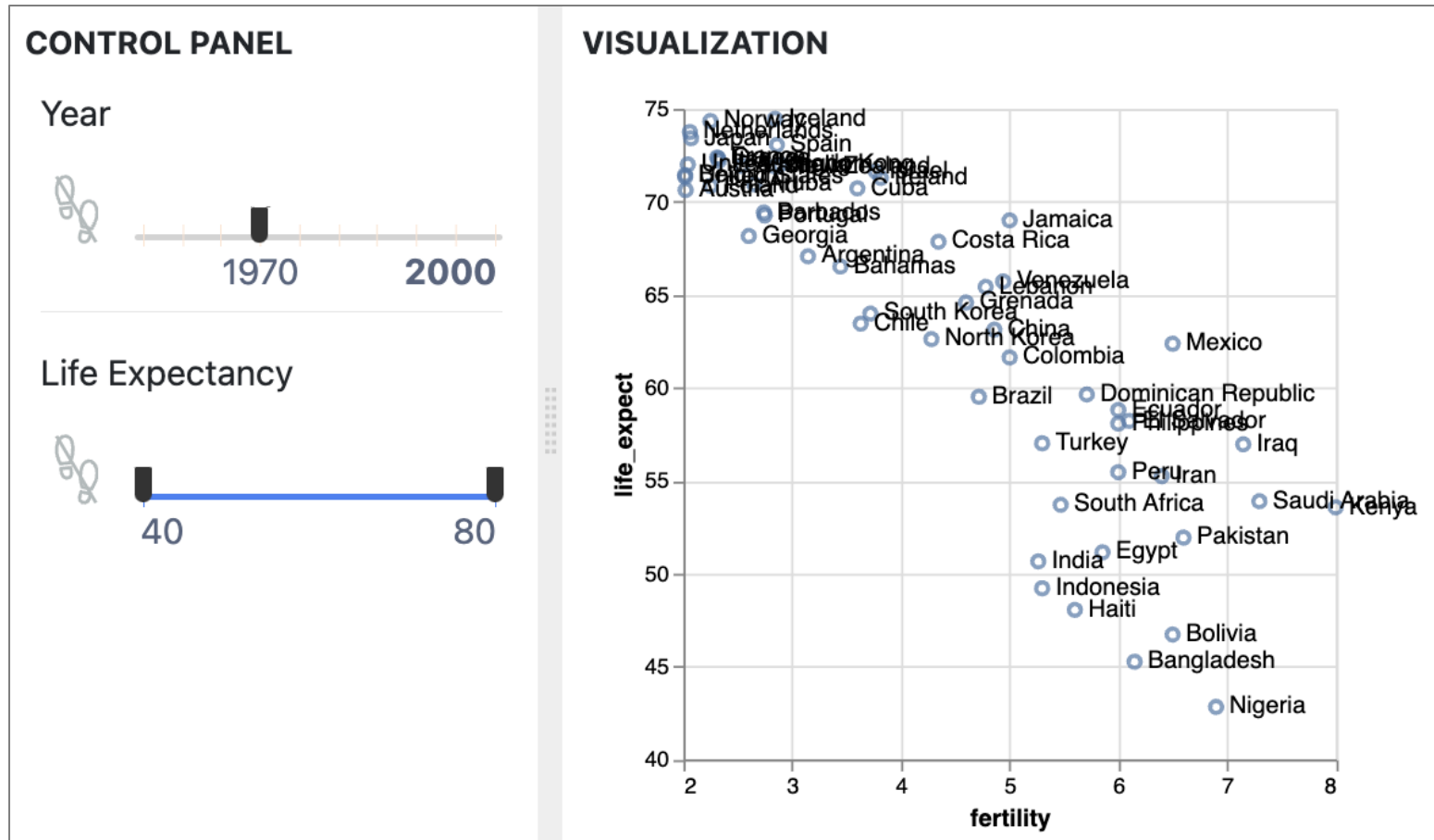
visualization to widgets (inbound)

use-cases

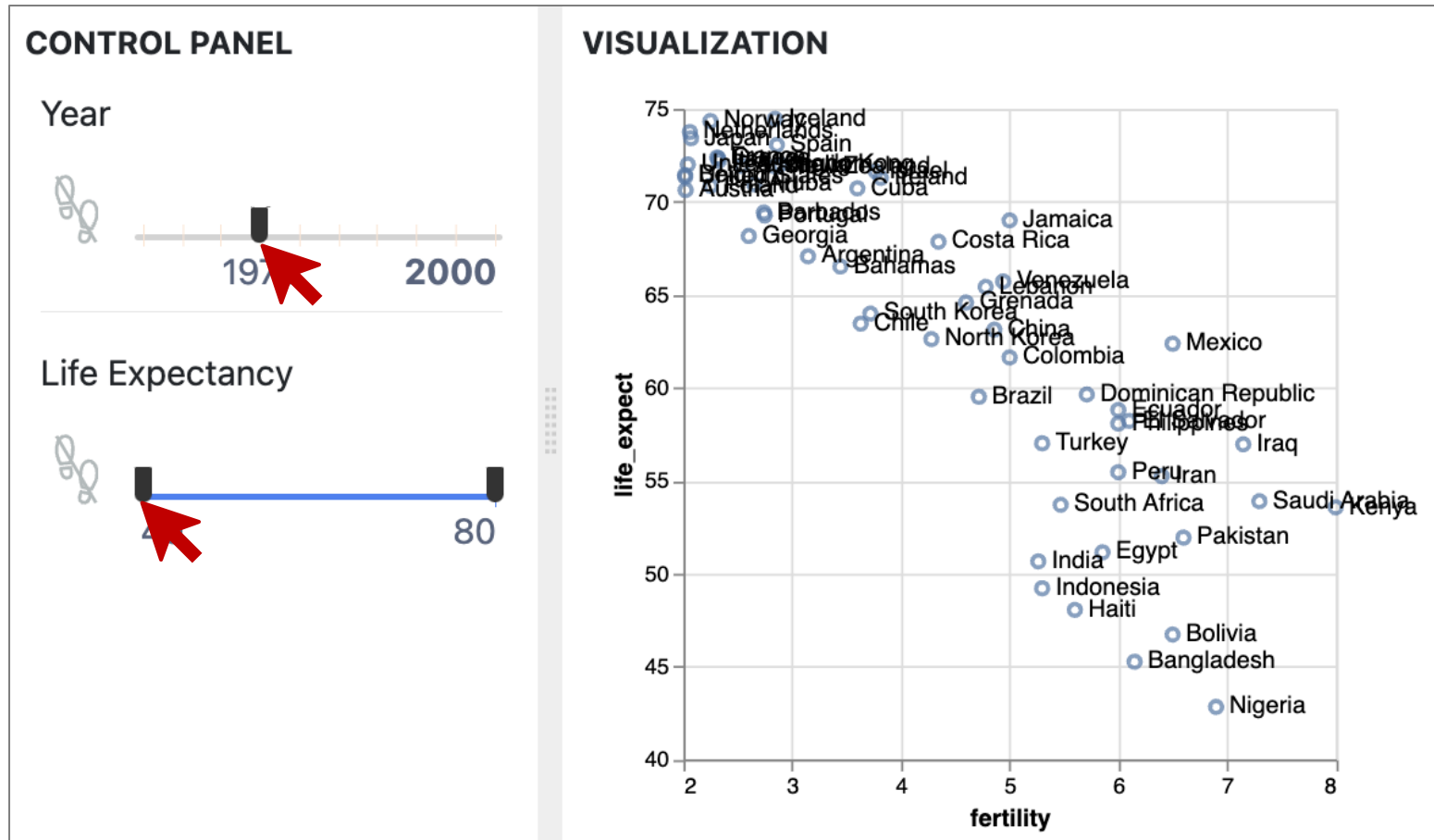
widgets to visualization (outbound)

visualization to widgets (inbound)

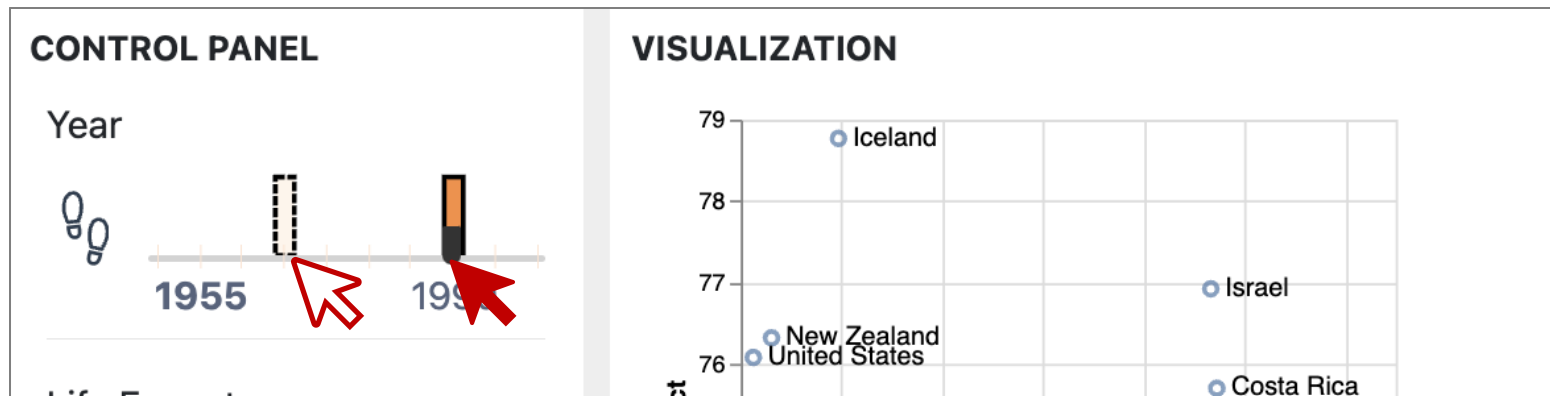
widget to visualization (outbound)



widget to visualization (outbound)



widget to visualization



```
1 // Apply the filter and update the visualization
2 widgetUpdated(model) {}
```

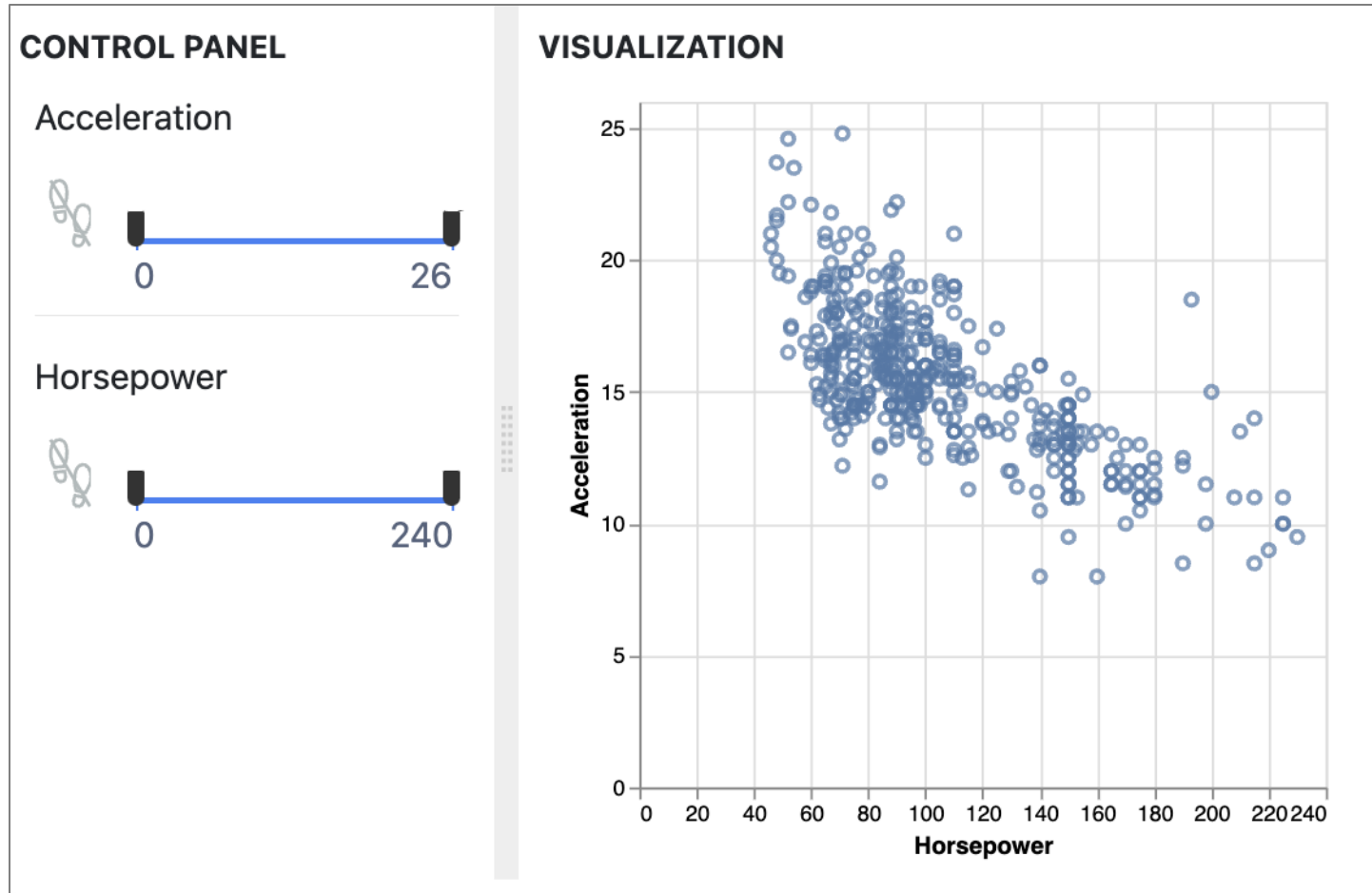
```
1 <provenance-slider
2 (selectedChange)="widgetUpdated($event.value)"/>
```

use-cases

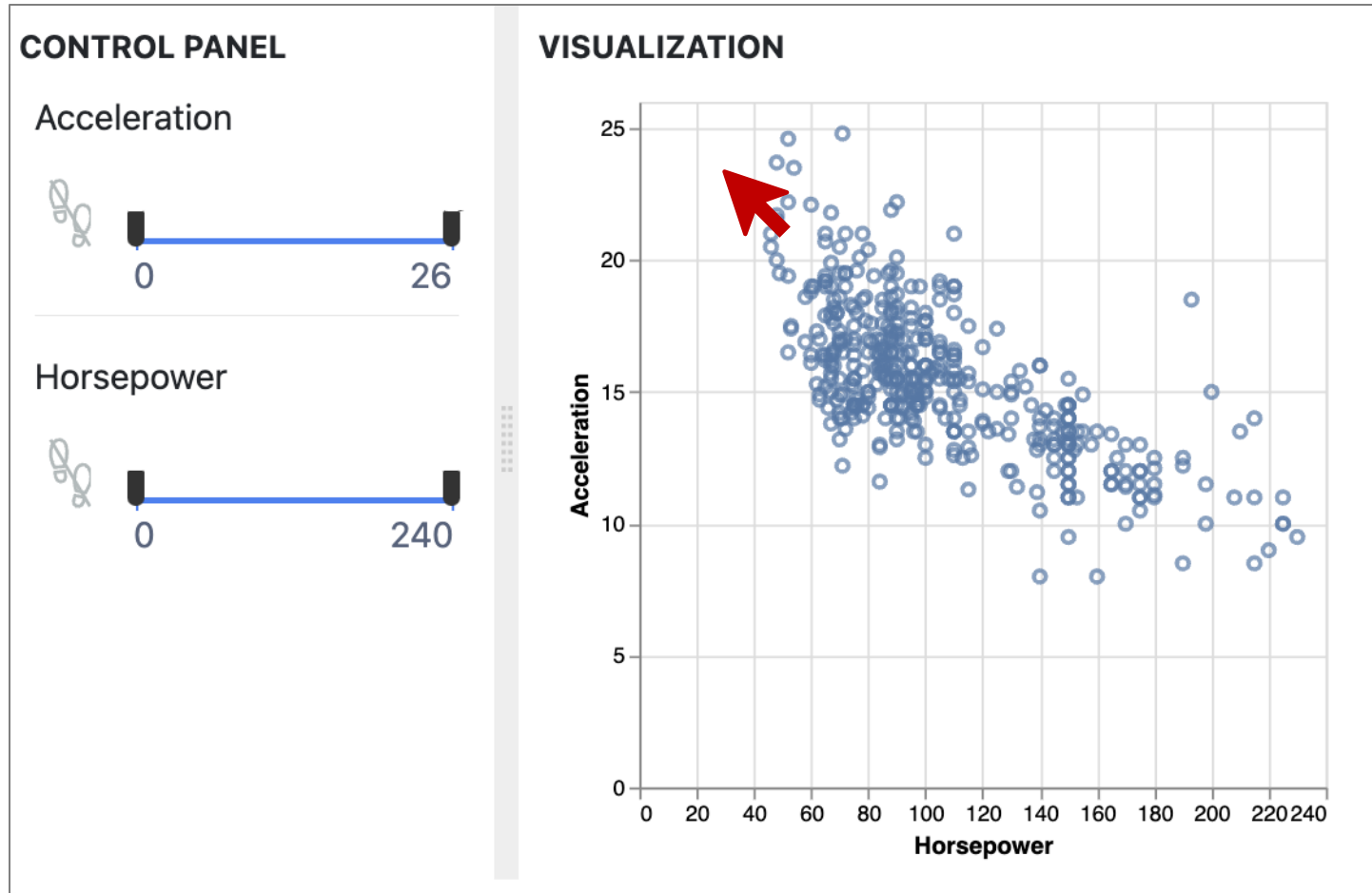
widgets to visualization (outbound)

visualization to widgets (inbound)

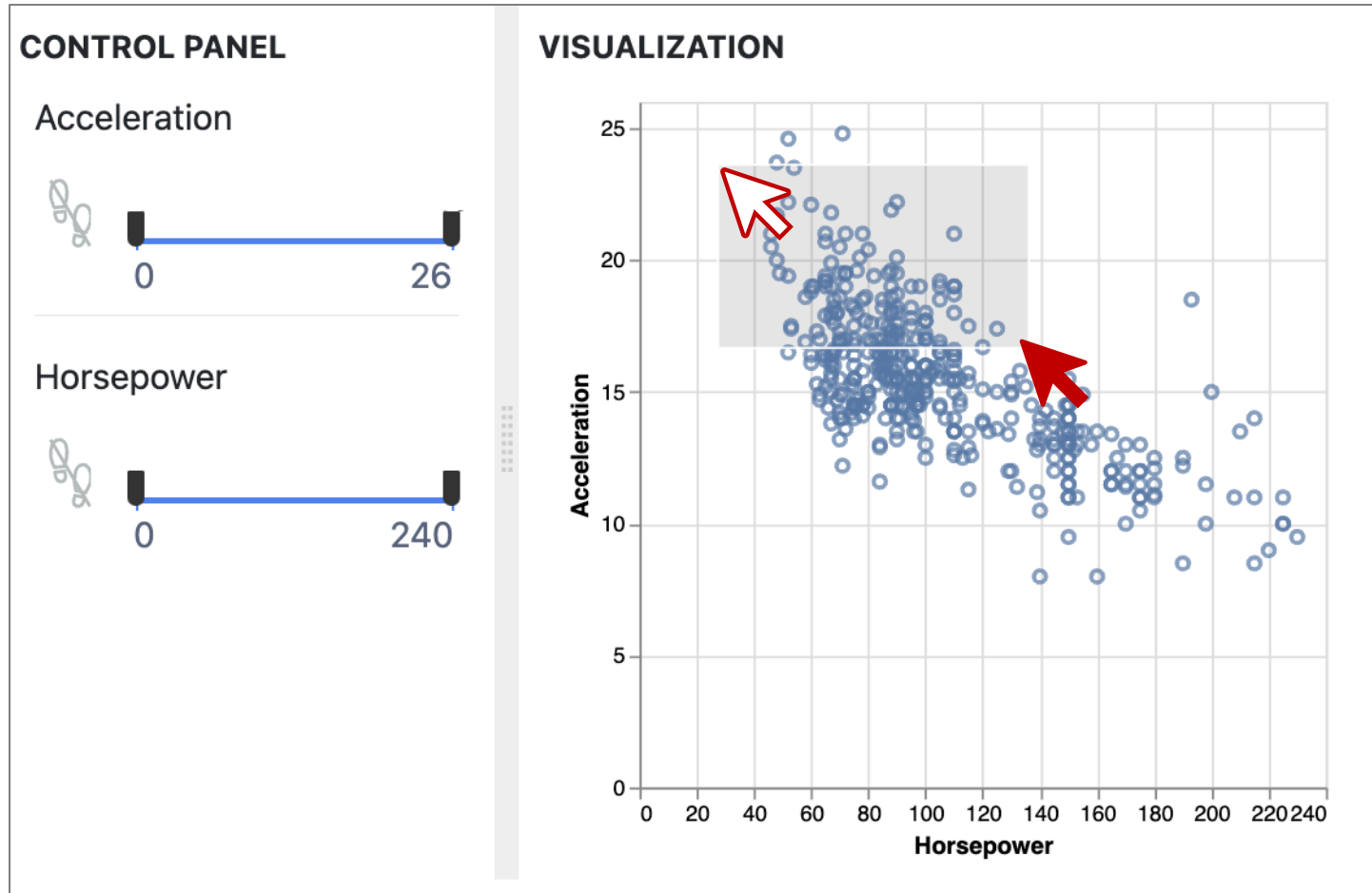
visualization to widget



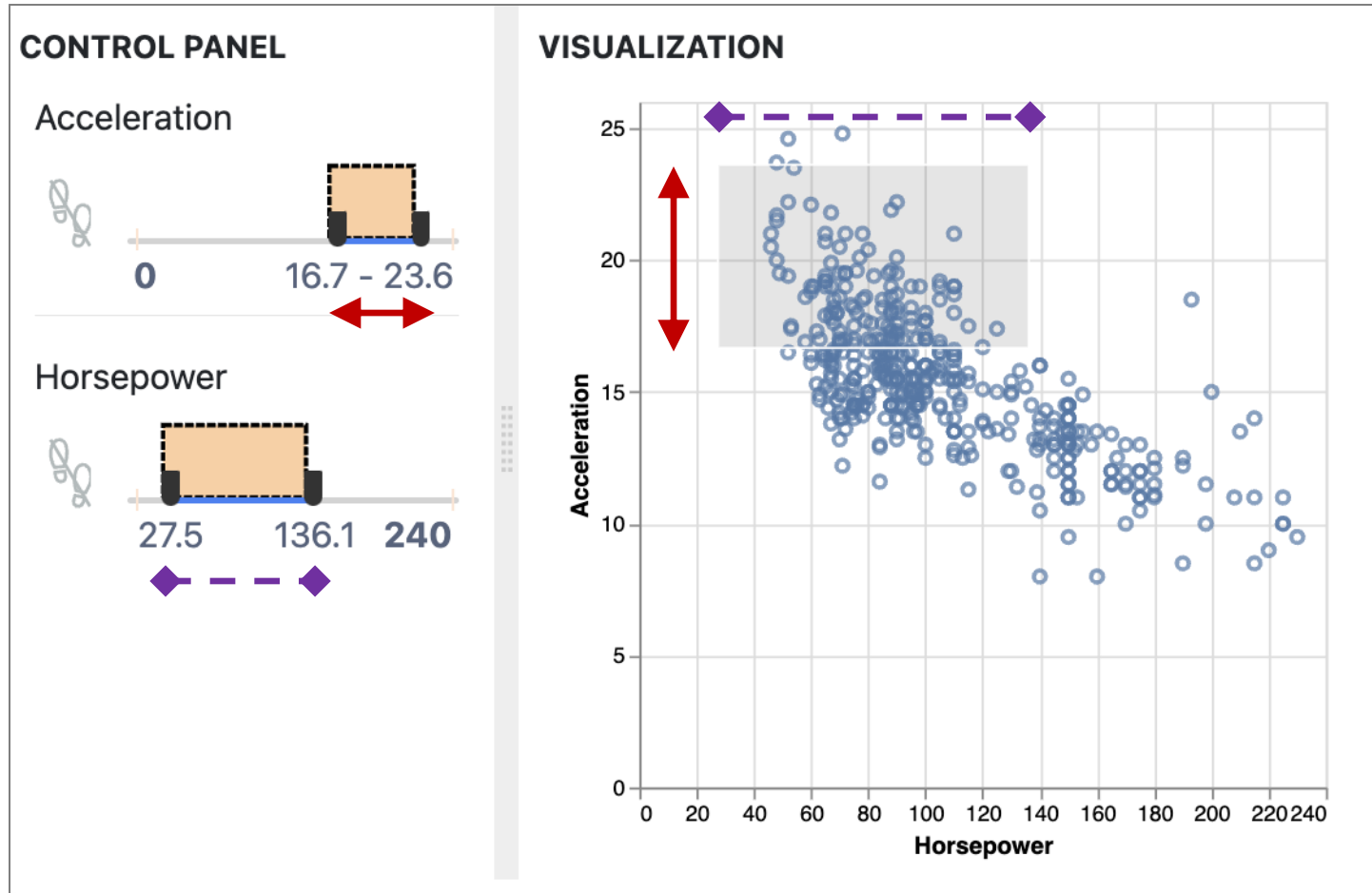
visualization to widget



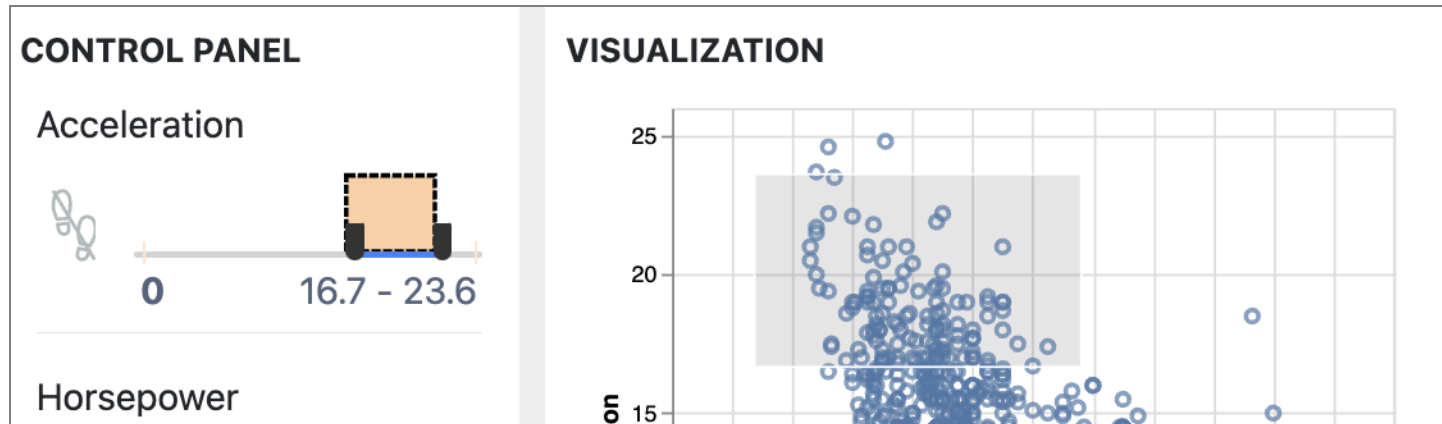
visualization to widget



visualization to widget



visualization to widget



```
1 visBrushed(brush_extent) {  
2     acceleration["data"] = [{ ..., "value": brush_extent  
3         }] // [16.7, 23.6]  
4     acceleration["revalidate"] = true  
5 }
```

```
1 <provenance-slider [(provenance)]="acceleration" />
```

replicating prior work

scented widgets

Willett, W., Heer, J., & Agrawala, M. *"Scented Widgets: Improving Navigation Cues with Embedded Visualizations."* **TVCG 2007**.

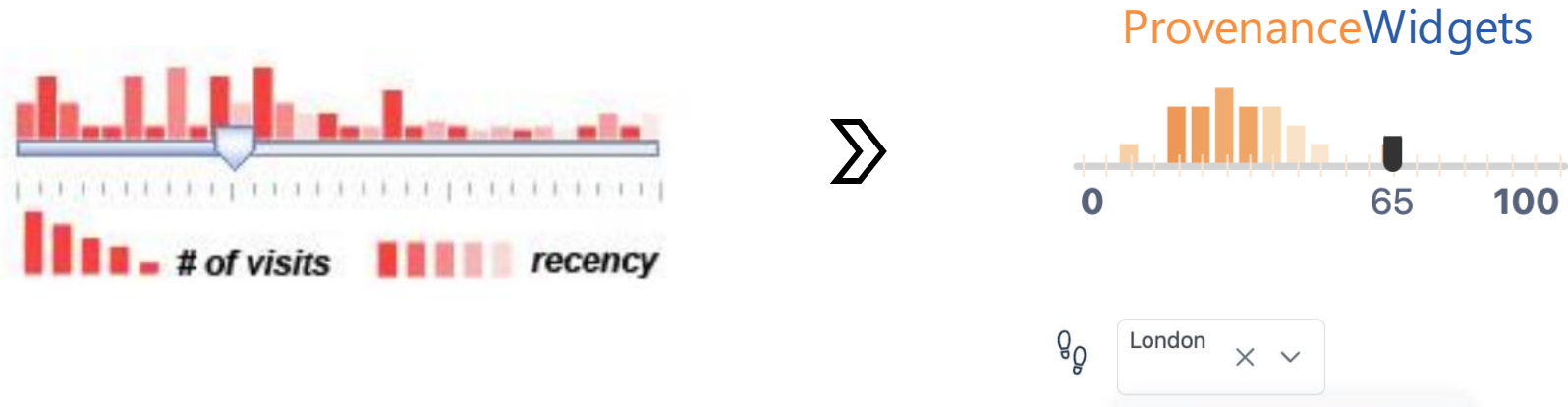
phosphor objects

Baudisch, P., Tan, D., Collomb, M., Robbins, D., Hinckley, K., Agrawala, M., Zhao, S., & Ramos, G. *"Phosphor: Explaining Transitions in the User Interface using Afterglow Effects."* **UIST 2006**.

dynamic query widgets

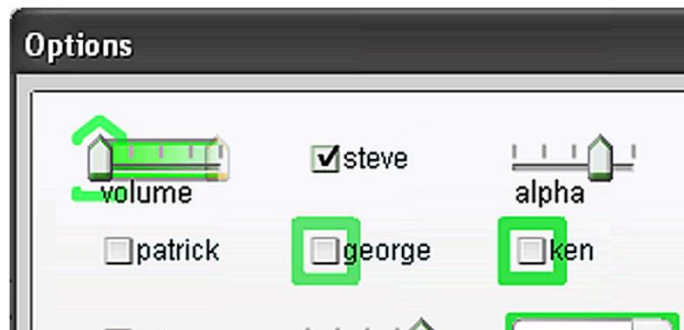
Williamson, C., & Shneiderman, B. *"The Dynamic HomeFinder: Evaluating Dynamic Queries in a Real-estate Information Exploration System."* **SIGIR 1992**.

scented widgets – replicating prior work

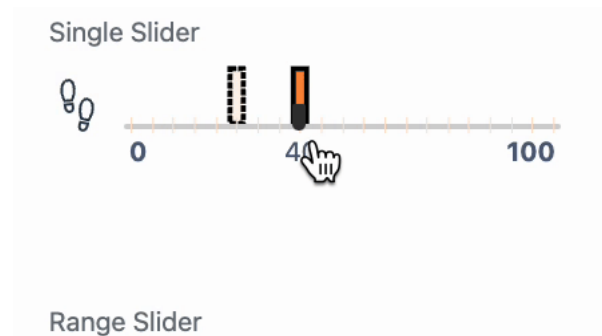


```
1 historical_usage_logs = {  
2   "revalidate": true,  
3   "data": [{"value": [40], "timestamp": _},  
4             {"value": [40], "timestamp": _},  
5             {"value": [50], "timestamp": _}, ...]  
6 }  
  
1 <provenance-slider [freeze]="true"  
2   [(provenance)]="historical_usage_logs" />
```

phosphor objects – replicating prior work



ProvenanceWidgets



```
1 widgetUpdated() {  
2   if (provenance) provenance = {  
3     "data": provenance["data"].slice(-2),  
4     "revalidate": true }  
5 }  
  
1 <provenance-slider [(provenance)]="provenance"  
2   (provenanceChange)="widgetUpdated()" />
```

expert case studies with 4 developers

"Develop a  explorer system using ProvenanceWidgets."

1



2

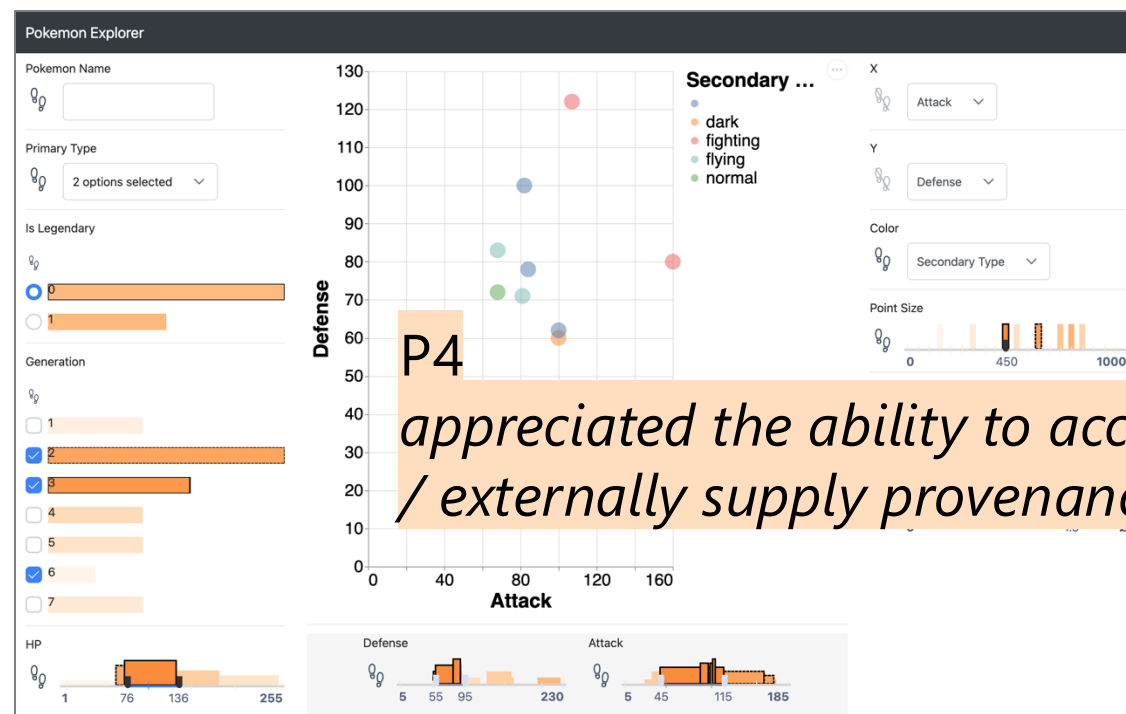
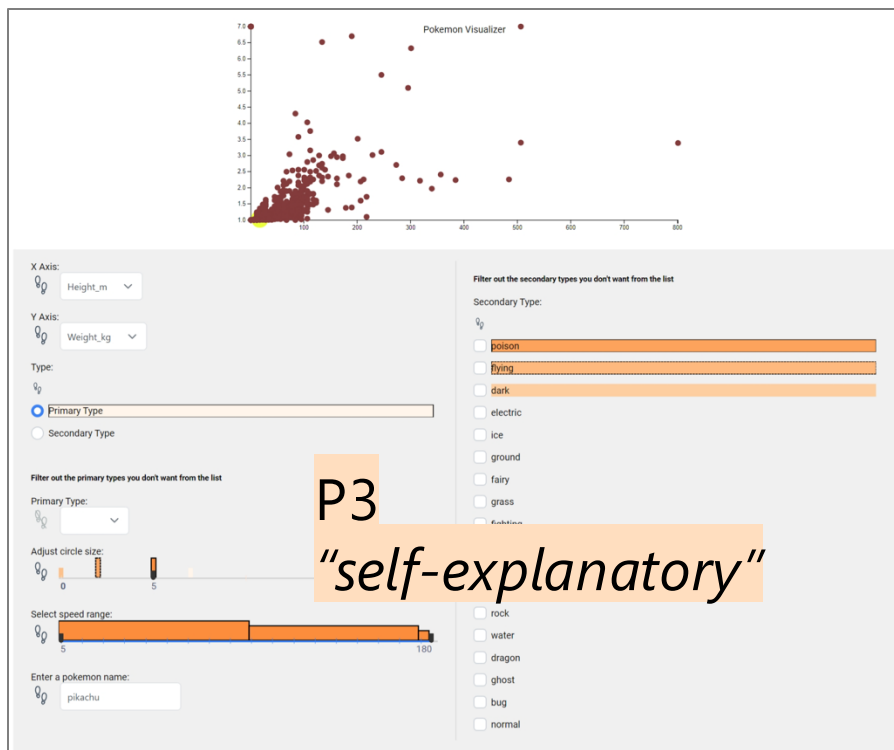
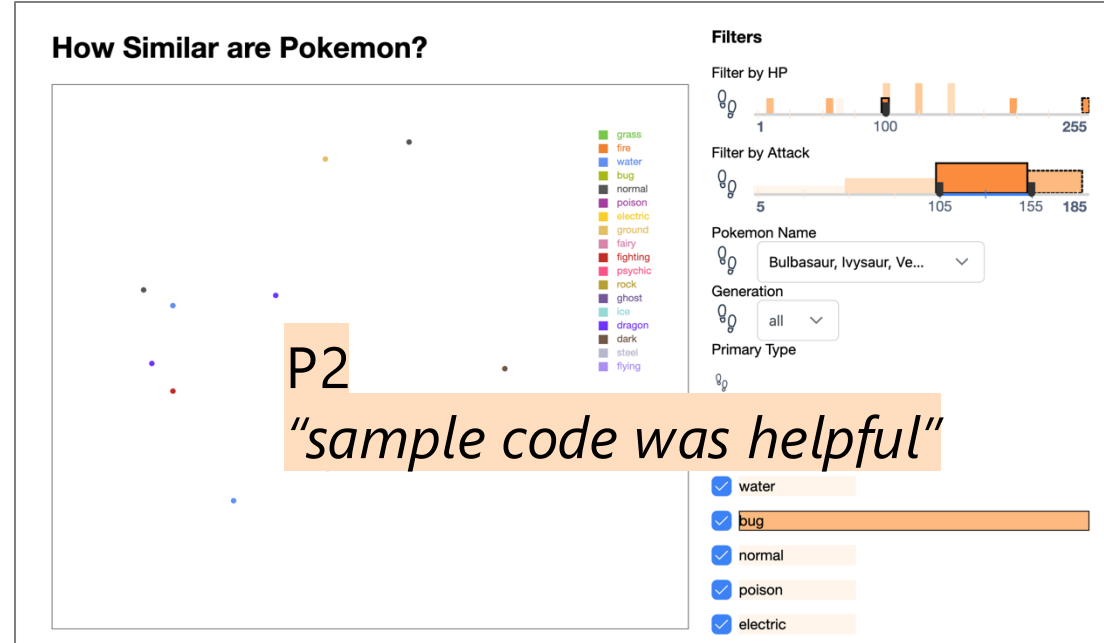
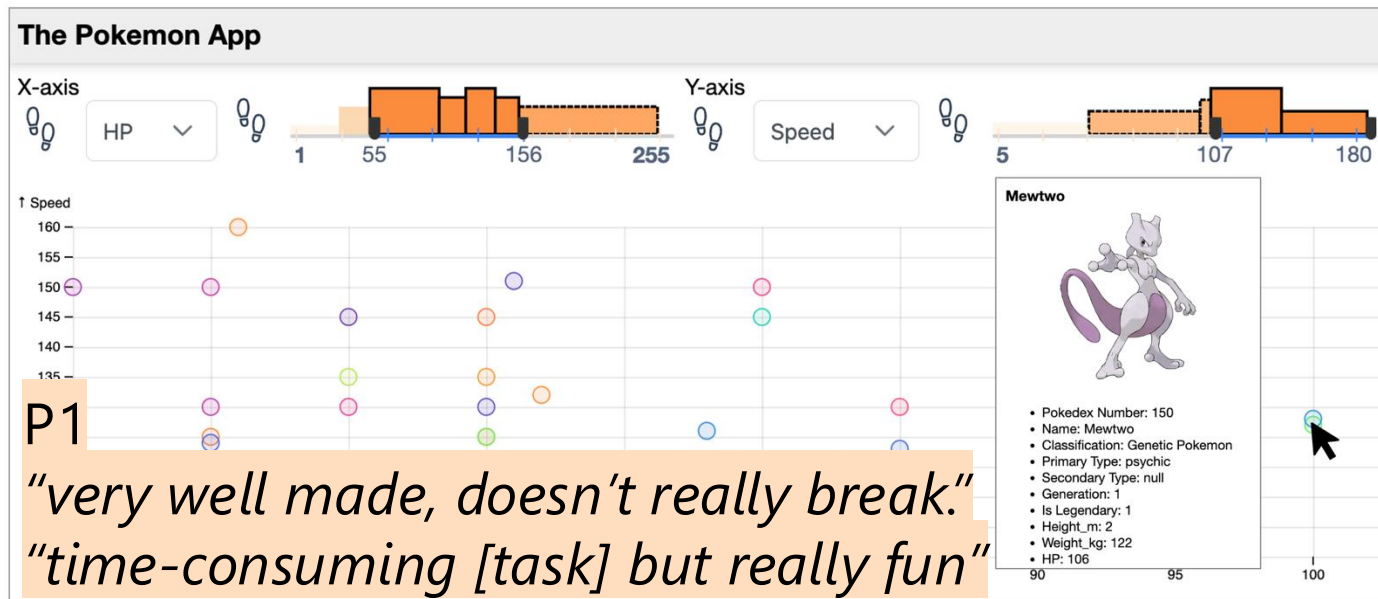


3

Onboarding
Interview (30 mins)

Work on Task
(~1 week)

Offboarding
Interview (30 mins)



ProvenanceWidgets is open-source and  !

Install the library

```
npm install provenance-widgets
```

available as both **Angular** & **Web-Components**

For the documentation, showcase, and starter code, go to:

ProvenanceWidgets.github.io

ProvenanceWidgets.github.io



arpit **narechania**
anarechania3@gatech.edu



kaustubh **odak**
kodak@gatech.edu



mennatallah **el-assady**
menna.elassady@ai.ethz.ch



alex **endert**
endert@gatech.edu