Problems in Software Development

Chenglong Wang

Jan.13.2015

Problem 1: Automatically API migration / API upgrade

- A new version of API may introduce some incompatible changes compared to its old version.
  - Google Calendar v2.0 → v3.0
  - Angular JS v1.2 → v1.3
  - Facebook Platform API v2.4 → v2.5

- Users need to modify their original code to continue their development.

- Developers will provide some guide and examples to demonstrate the modification.

Automatically generating patches?

Angular JS migration guide

Due to c8700f04, animation-related callbacks are now fired on `$animate.on` instead of directly being on the element.

```
// < 1.4
element.on('$animate:before', function(e, data) {
  if (data.event === 'enter') { ... }
});
element.off('$animate:before', fn);

// 1.4+
$animate.on('enter', element, function(data) {
  //...
});
$animate.off('enter', element, fn);
```

myElement.on('$animation:before', function(e, data) {
    if (data.event === 'enter') { dosomething();}
});
myElement.off('$animate:before', fn);

$animate.on('enter', myElement, function(data) {
    dosomething();
});
$animate.off('enter', myElement, fn);
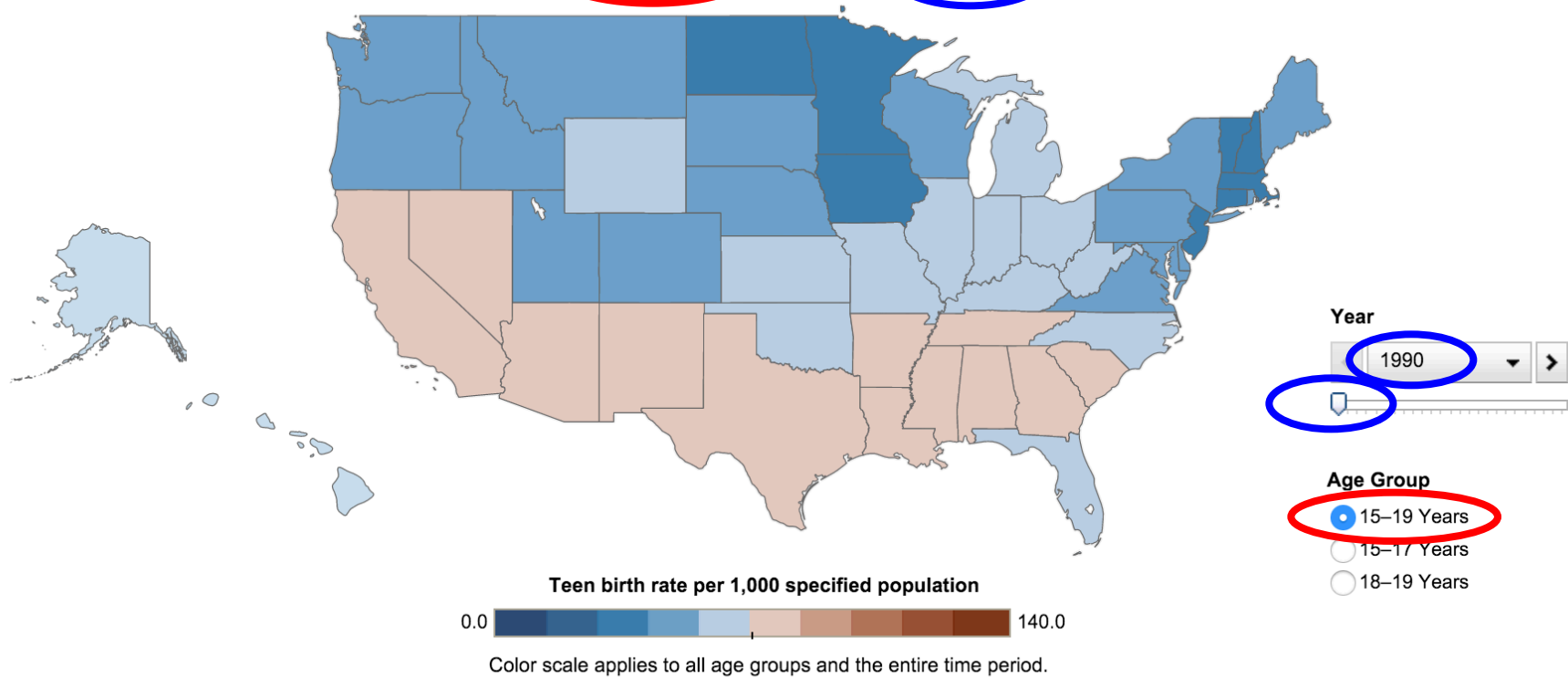
Why challenging?
What are the possible solutions?

- Web APIs are commonly used in JavaScript, and most current research focus on Java/C.
  - 'Infer' their 'types' by learning their names.

- Current research focus more on rule-based transformation (to achieve safety).
  - Automation as a method to help programmers, rather than a substitution of them. (Not necessarily all correct.)
  - Infer the transformation rules through the examples.

- Existing approach to support auto-migration supports only one-to-many substitution. [1]
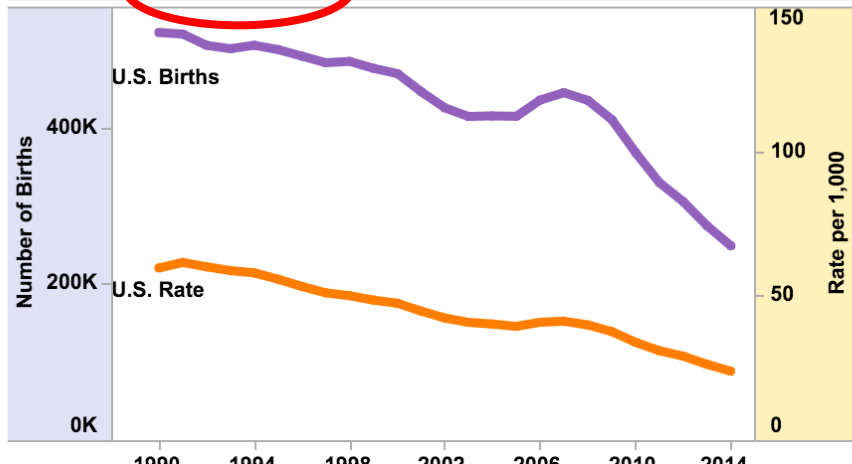
If enough time…

Problem 2: Data consistency issues in interactive Web applications

- Suppose we have a value ***date.***

- ***date*** may be displayed in different position in the webpage, possibly:
  - Title, caption of a graph, text in a paragraph ...

- ***date*** can be modified in multiple ways:
  - A slide bar, input box, clock ...

- Making sure all occurrences of ***date*** are consistent can be challenging.
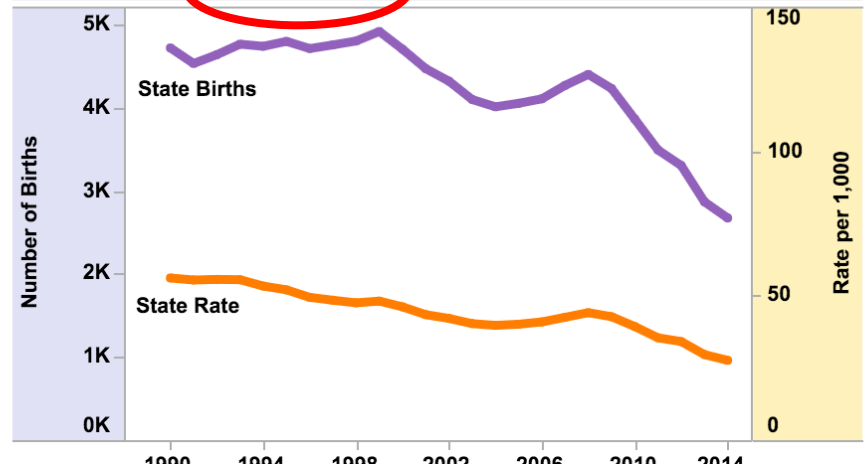
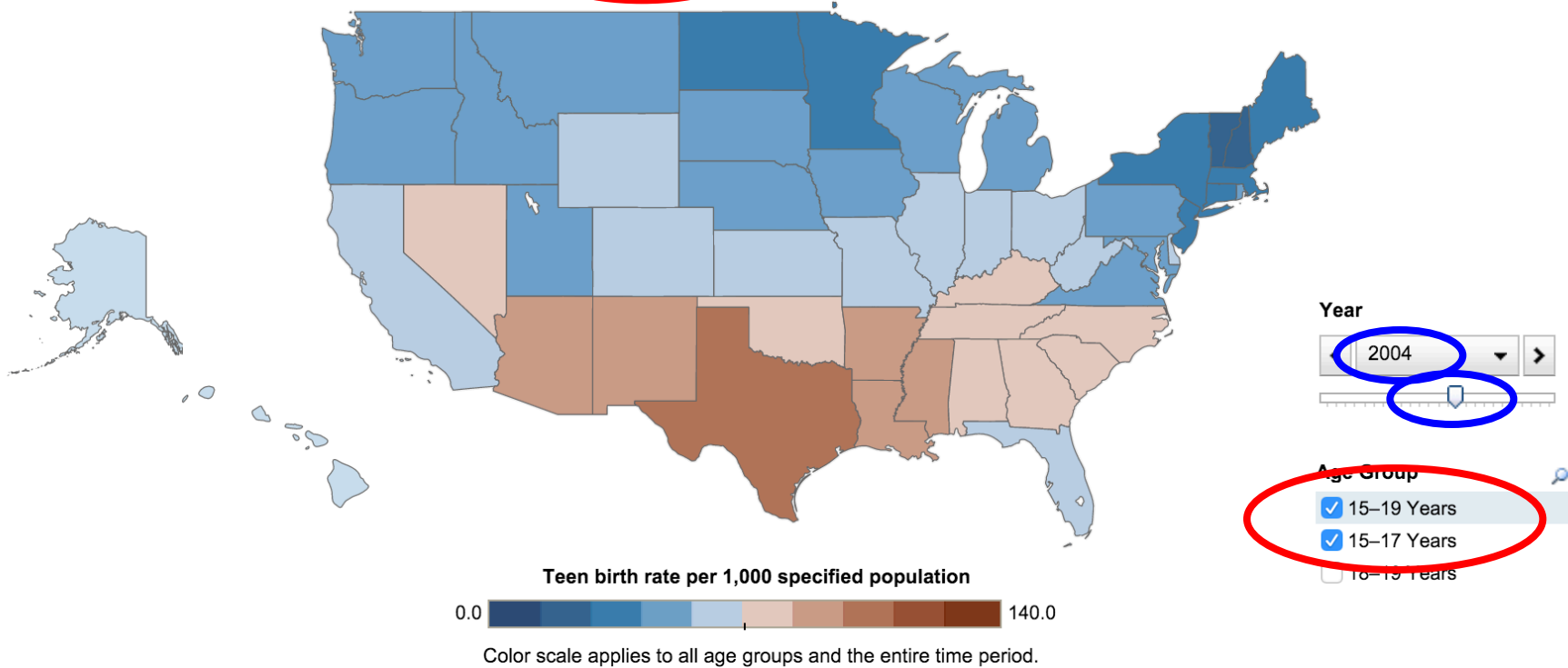# State Teen Birth Rates for Females Aged 15–19 Years: United States, 1990

Year

1990

Age Group
- ● 15–19 Years
- ○ 15–17 Years
- ○ 18–19 Years

Teen birth rate per 1,000 specified population

0.0 — 140.0

Color scale applies to all age groups and the entire time period.

Births and Birth Rates for Females Aged 15–19 Years: United States, 1990–2014

U.S. Births

U.S. Rate

Births and Birth Rates for Females Aged 15–19 Years: Kansas, 1990–2014

State Births

State Rate

State Teen Birth Rates for Females Aged
*: United States, *2004*

Teen birth rate per 1,000 specified population

0.0 ──────────────────────────── 140.0

Color scale applies to all age groups and the entire time period.

Year

2004 ▾ >

Age Group 🔍

☑ 15–19 Years
☑ 15–17 Years
☐ 18–19 Years

Births and Birth Rates for Females Aged
15–19 Years & 15–17 Years: United States, 1990–2014

U.S. Births
U.S. Rate
U.S. Births
U.S. Rate
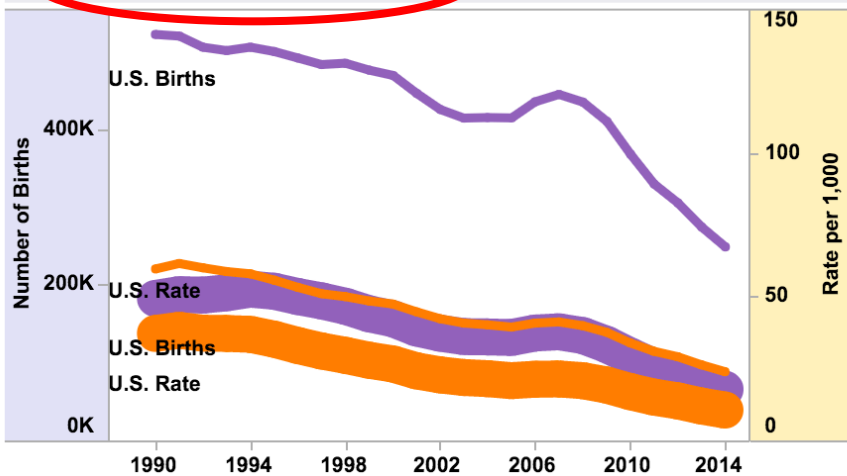
Births and Birth Rates for Females Aged
15–19 Years & 15–17 Years: Kansas, 1990–2014
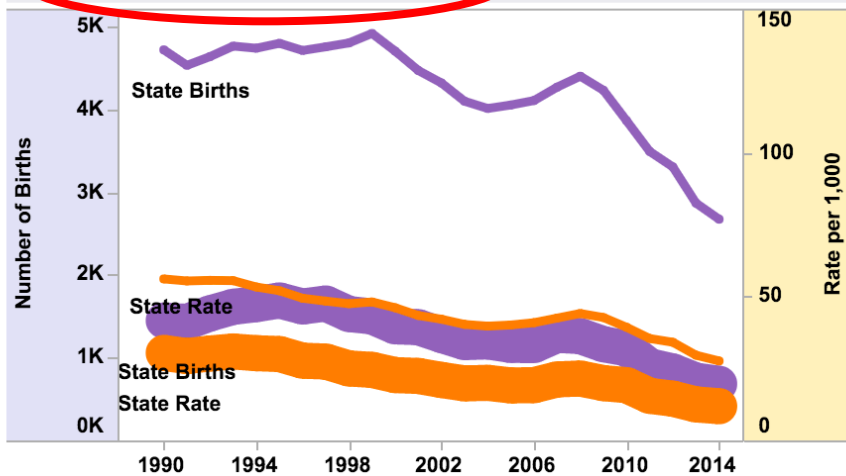
State Births
State Rate
State Births
State Rate

NFL players: height & weight over time

By Noah Veltman

NFL players: height & weight over time

By Noah Veltman

Color legend: 0.3% 0.6% 0.8% 1.1% 1.4% 1.7% 1.9% 2.2%
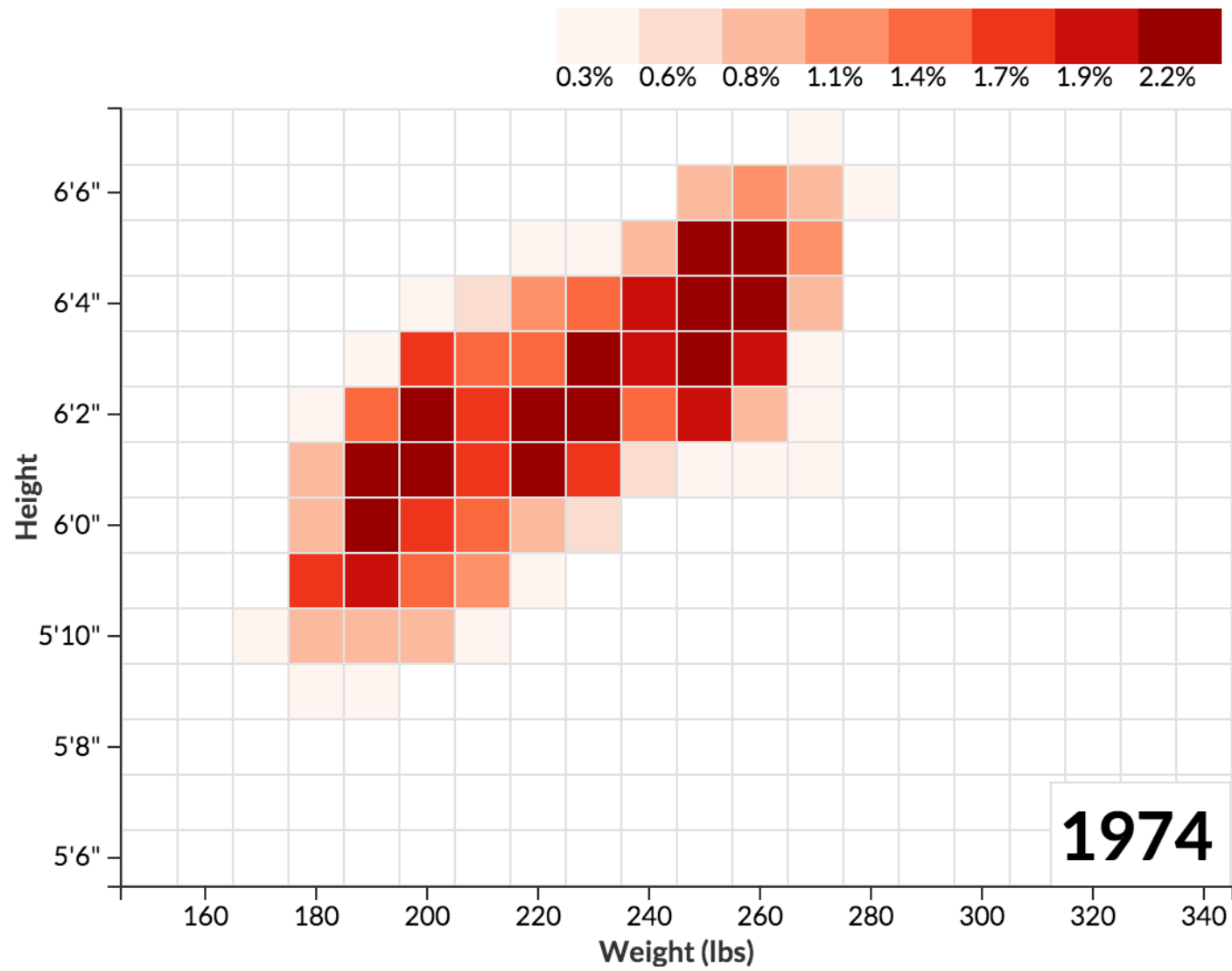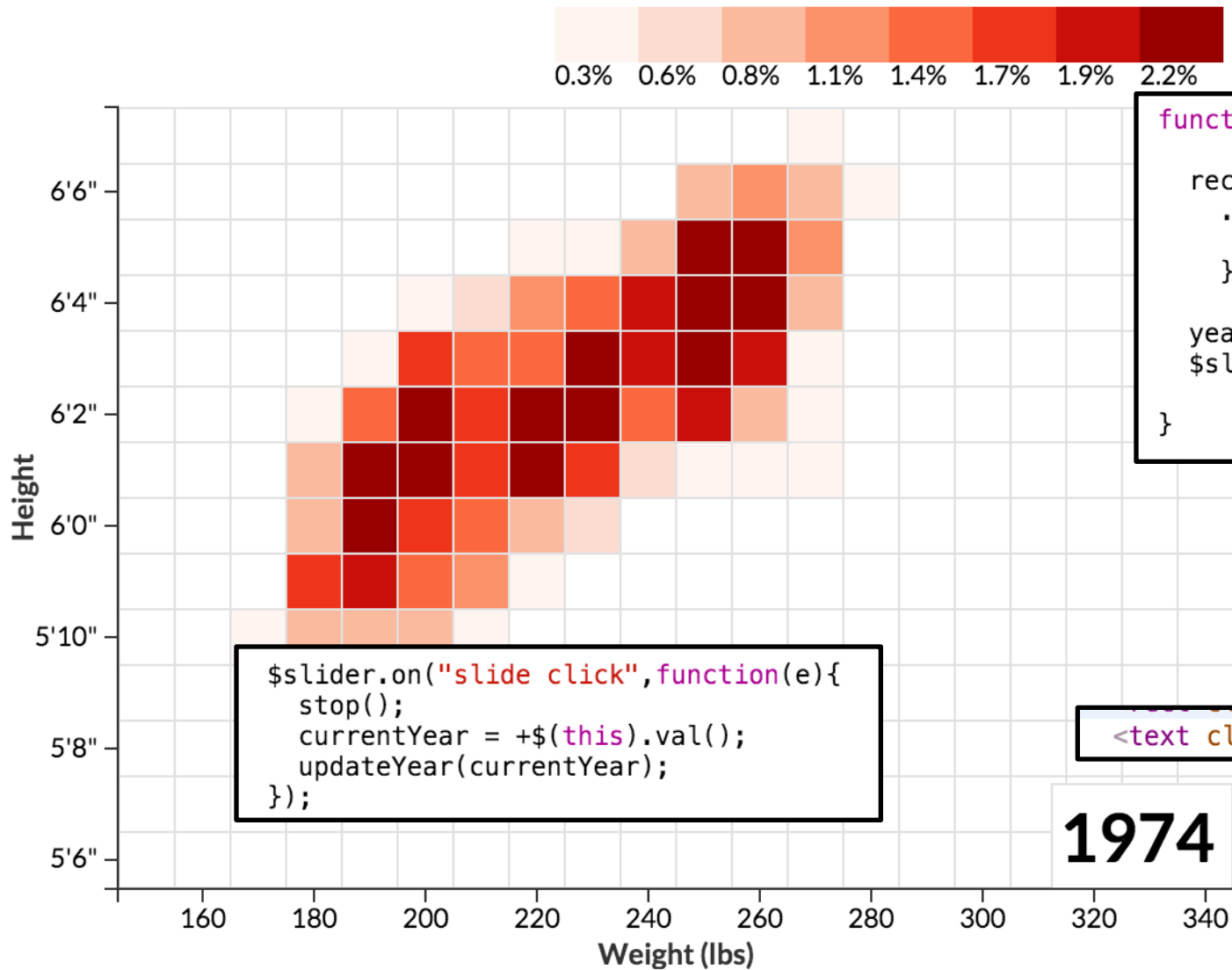
```
function updateYear(yr) {

    rects.data(bins[yr])
      .style("fill",function(d){
          return color(d.pct);
      });

    year.text(yr);
    $slider.val(yr);

}
```

Height (y-axis): 6'6", 6'4", 6'2", 6'0", 5'10", 5'8", 5'6"

Weight (lbs) (x-axis): 160, 180, 200, 220, 240, 260, 280, 300, 320, 340

```
$slider.on("slide click",function(e){
    stop();
    currentYear = +$(this).val();
    updateYear(currentYear);
});
```

```
<text class="chart-title year" x
```

1974

PLAY

1974

```
::before
<span class="year">1974</span>
::after
```

References

- [1] A graph-based approach to API usage adaptation. Hoan Anh Nguyen etal. (OOPSLA '10)