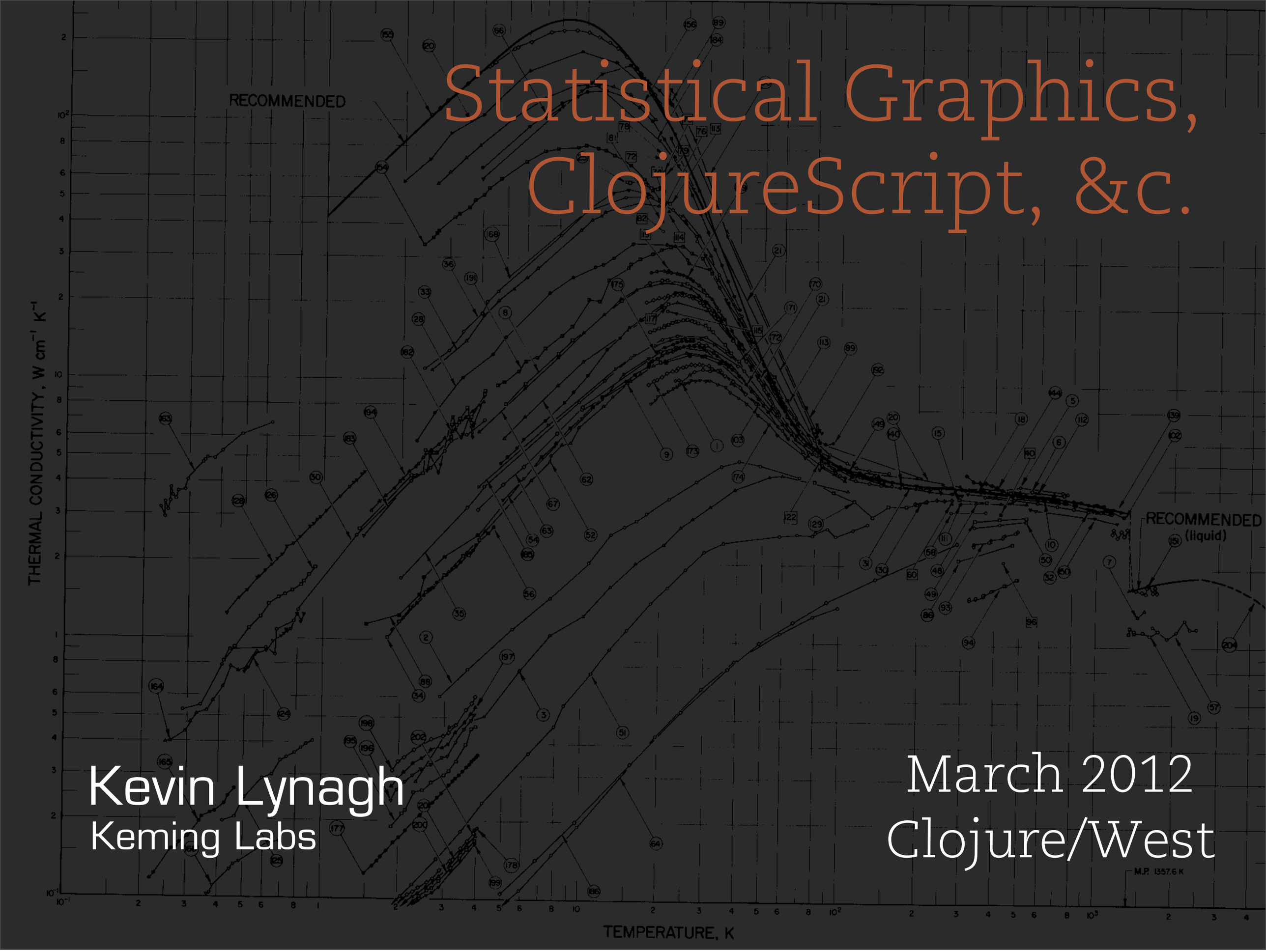
Statistical Graphics, ClojureScript, &c.



Kevin Lynagh
Keming Labs

March 2012
Clojure/West

Agenda

Agenda

I. Info. design

We are terrible.

Agenda

I. Info. design

We are terrible.

II. In practice

On the Internets, with Cljs.

I.

Information Design

Example.

ORBITZ

Welcome to Orbitz. Sign in | Register now

My Trips | My Account | Traveler Update | Customer Support

Quick Search | Vacation Packages | Hotels | Flights (NO FEES) | Cars | Cruises | Activities | Deals

Like Orbitz on Facebook Like 311k

ORBITZ MATRIX DISPLAY℠

My Search

Find flights by: Airline / Stops / Price

Stops	Alaska Airlines	American Airlines	Multiple Carriers	Delta Air Lines
Non-stop	$257 total $278	$257 total $278	$257 total $285	$287 total $308
1 stop	The lowest fare is shown above. Still want more options? Show all flights			

Fares are per person in US dollars, using e-tickets. Total fare includes airfare taxes and fees .
Additional baggage charges may apply.
Some itineraries require paper tickets with an additional charge. Changes after booking are subject to change fees.

Need a hotel too?
Book flight + hotel together for a discount
Find Flight + Hotel

Sort flights by: **Lowest price** | Departure time | Shortest flight

Airport codes

Showing **lowest priced flights** (45 flights out of 180 total) See all 180 flights

Select **$257** + $21 taxes & fees = $278 USD per person NO BOOKING FEES PRICE ASSURANCE

Leave **Thu, Mar 15** — **Alaska Airlines 400**
Depart:**6:10am** Portland, OR (PDX)
Arrive:**7:55am** San Jose, CA (SJC)
Non-stop Economy | 1hr 45min | Boeing 737 | View seats
Choose this departure

Return **Sun, Mar 18** — **Alaska Airlines 2546**
operated by HORIZON AIR DBA ALASKA HORIZON
Depart:**6:15am** San Jose, CA (SJC)
Arrive:**8:19am** Portland, OR (PDX)
Choose this return

Change Search

From City name or airport
PDX
incl. nearby airports

To City name or airport
SJC
incl. nearby airports

Leave
03/15/12
Depart
Anytime

Sponsored Links

Fly Portland to San Jose
Enjoy every day low fares on Alaska Air. Book today!
www.alaskaair.com

Costa Rica from $149
Costa Rica Fares Just Dropped! Book Now to Lock In the Best Deals.
www.lowfares.com/Costa-Rica-Fares

San Jose Cheap Deals
Food, Spas & More up to 90% Off. Limited Availability- Act Now!
www.groupon.com

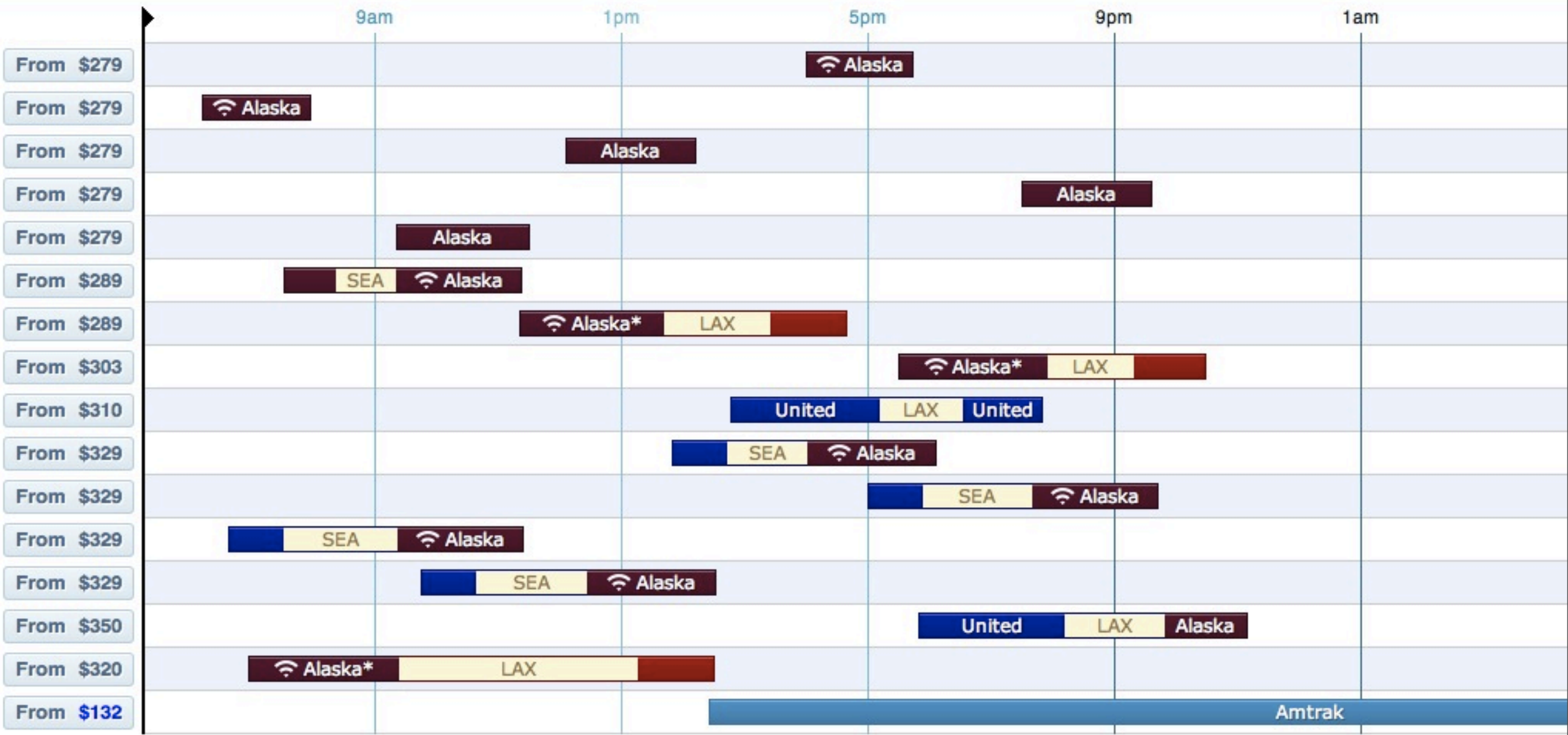
Cheapest San Jose Flights
Up to 65% off **San Jose Flights**. No Service Fees on all **Flights**!

hipmunk

PDX ⟷ SJC, Mar 15 ⟷ Mar 18 | Search for hotels in San Jose, CA | New Search

1 PDX → SJC on Thu, Mar 15 | 2 SJC → PDX on Sun, Mar 18

Sort By: Agony | Price | Duration | Departure | Arrival | Airlines | Non-stops



why did this take 10 years?

```
<div id="dashboard">
  <div class="news">
    <div class="alert issues_opened"><div class="body"><div
class="title">
      <a href="/mbostock">mbostock</a> <span>opened</span>
<a href="/mbostock/d3/pull/482">pull request 482</a> on <a
href="/mbostock/d3">mbostock/d3</a>
      <time class="js-relative-date"
datetime="2012-01-30T03:25:22Z" title="2012-01-30
03:25:22">January 30, 2012</time>
    </div>
    <div class="details">
      <div class="gravatar"><img height="30" src="https://
secure.gravatar.com/avatar/
005a27e09fe946ebef64bf4d134efc0a?s=140&amp;d=https://
a248.e.akamai.net/assets.github.com%2Fimages%2Fgravatars
%2Fgravatar-140.png" width="30" /></div>
      <div class="message">
        <blockquote>Fix polygon.centroid for open
polygons.</blockquote>
        1 commit with
        51 additions and
        25 deletions
      </div>
    </div>
```

```
- (id) initWithFrame:(CGRect)frame
{
    self = [super initWithDocument:self name:@"svg"];
   if (self) {
        _width = CGRectGetWidth(frame);
        _height = CGRectGetHeight(frame);
    }
   return self;
}

- (void)dealloc {
   [_version release];
   [super dealloc];
}

- (BOOL)parseFileAtPath:(NSString *)aPath {
   NSError *error = nil;

   SVGParser *parser = [[SVGParser alloc] initWithPath:aPath
document:self];

   if (![parser parse:&error]) {
      NSLog(@"Parser error: %@", error);
      [parser release];
```

```
(defn macroexpand-1 [env form]
  (let [op (first form)]
    (if (specials op)
      form
      (if-let [mac (and (symbol? op) (get-expander op env))]
        (apply mac form env (rest form))
        (if (symbol? op)
          (let [opname (str op)]
            (cond
             (= (first opname) \.)
                (let [[target & args] (next form)]
                  (list* '. target (symbol (subs opname 1))
args))
             (= (last opname) \.)
                (list* 'new (symbol (subs opname 0 (dec
(count opname)))) (next form))
             :else form))
          form)))))
```

printf( )

```
for (int i=0; i < flights.length; i++) {
  Flight f = flights[i]
  printf("%s $%d %s %s",
    f.flight_number, f.price,
    f.departure_time, f.arrival_time);
}
```

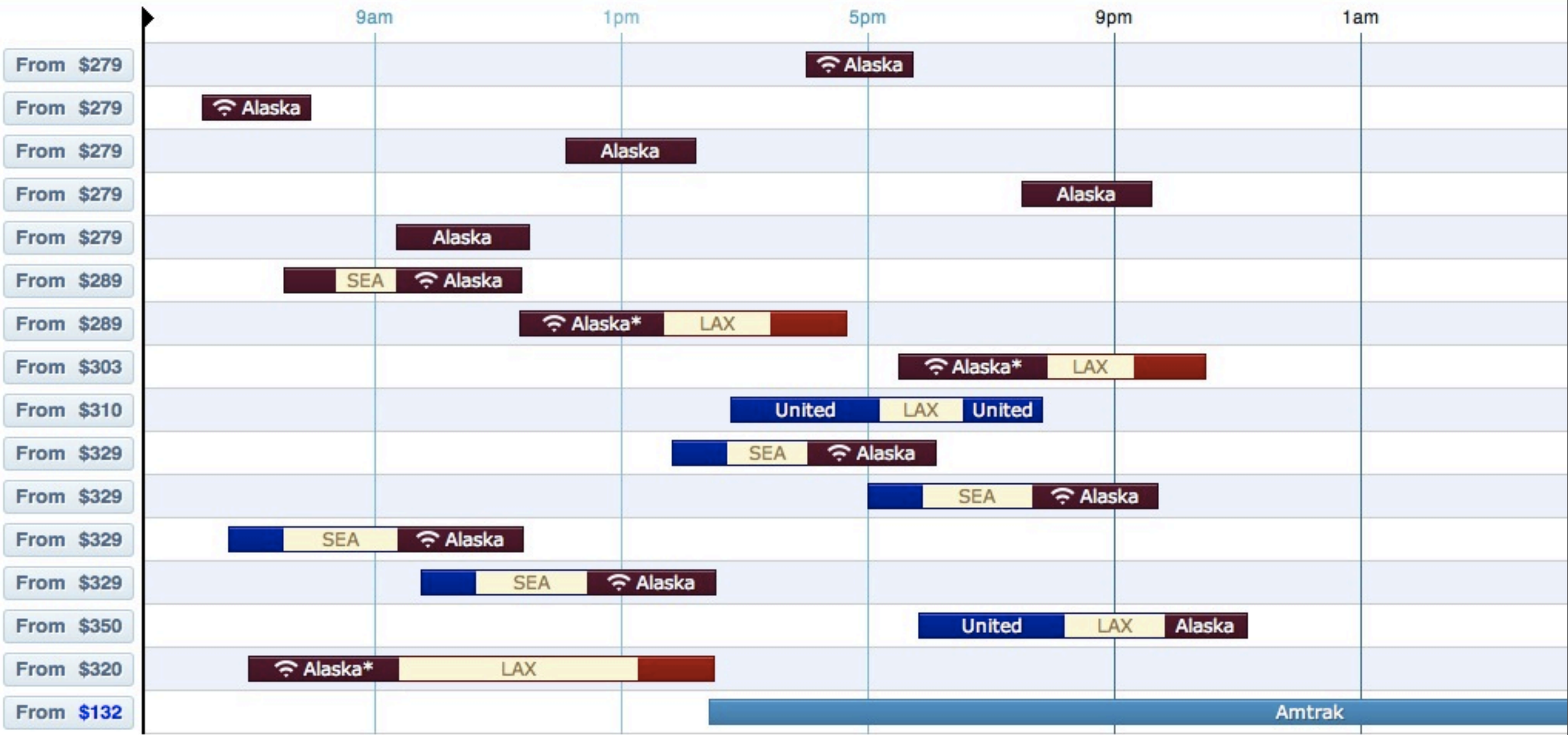
FNUM	PRIC	DEP	ARR
1120	$120	410p	545p
1010	$220	1110a	1234p
2017	$139	600a	811a
1118	$193	1239p	120p

hipmunk

PDX ⟷ SJC, Mar 15 ⟷ Mar 18 | Search for hotels in San Jose, CA | New Search

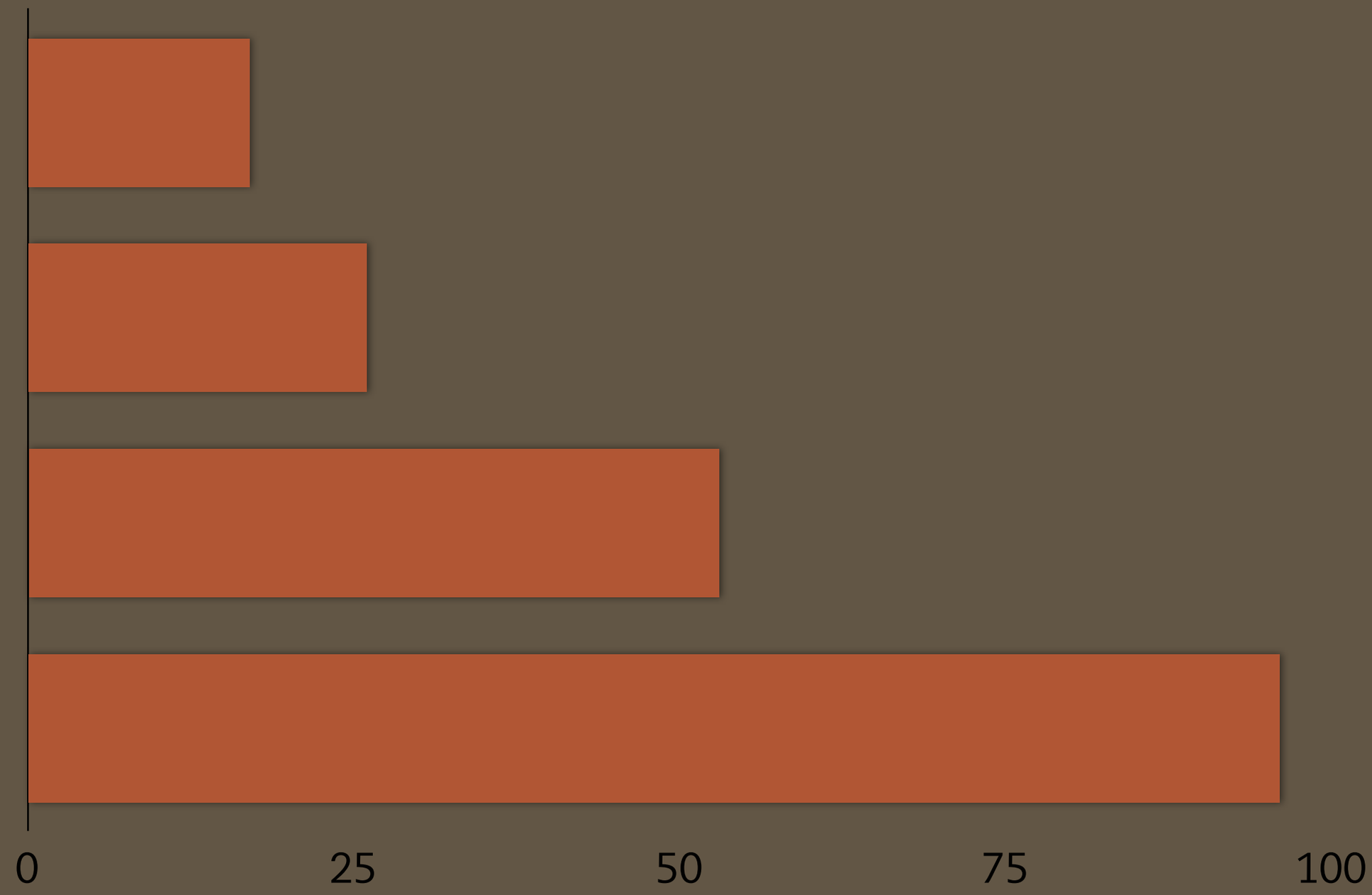
1 PDX → SJC on Thu, Mar 15 | 2 SJC → PDX on Sun, Mar 18

Sort By: Agony | Price | Duration | Departure | Arrival | Airlines | Non-stops

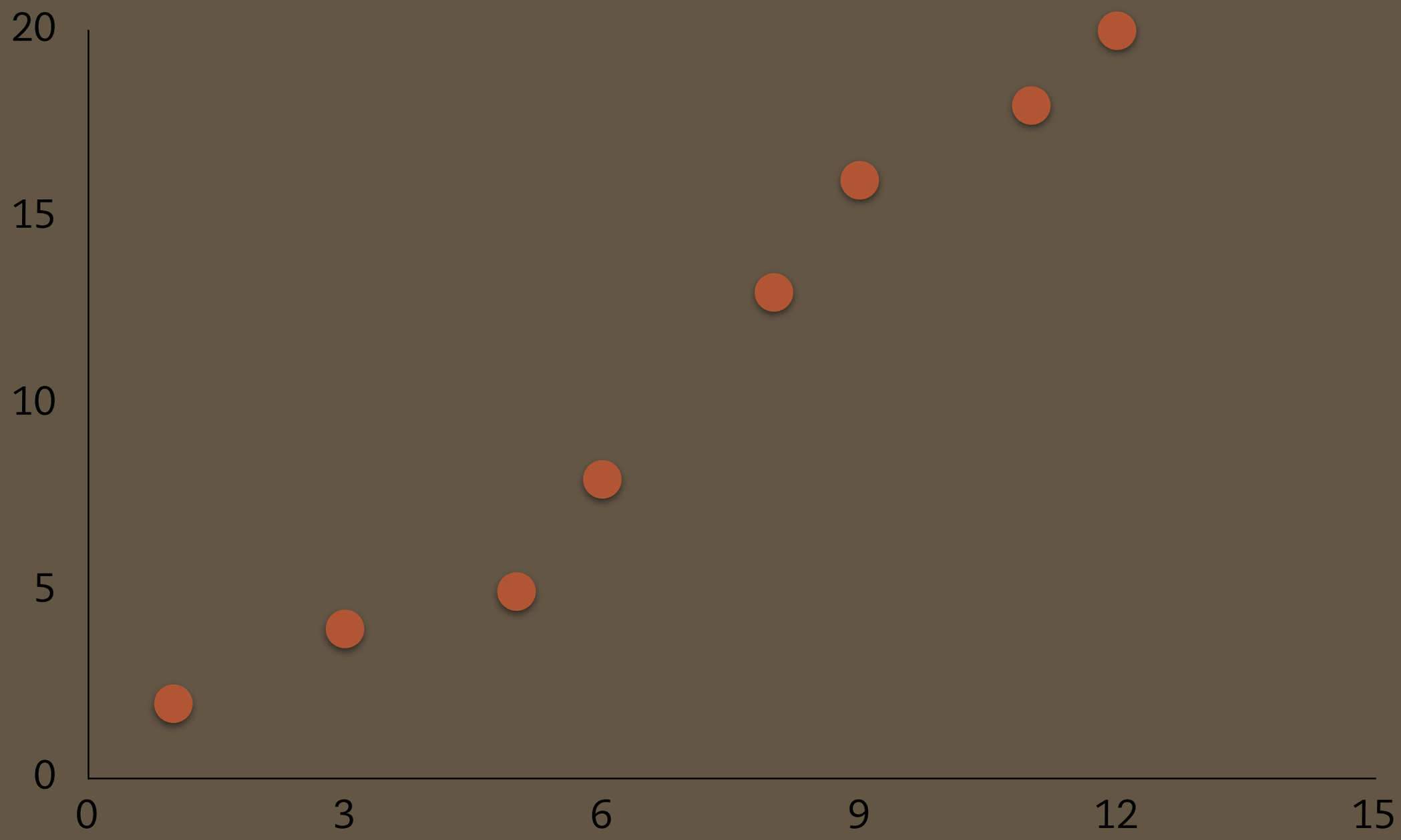


II. *Making* Pictures *of* Data

Data: [17, 26, 53, 96]



Data: [[1, 2] [3, 4] [5, 5] [6, 8] [8, 13] [9, 16] [11, 18] [12, 20]]

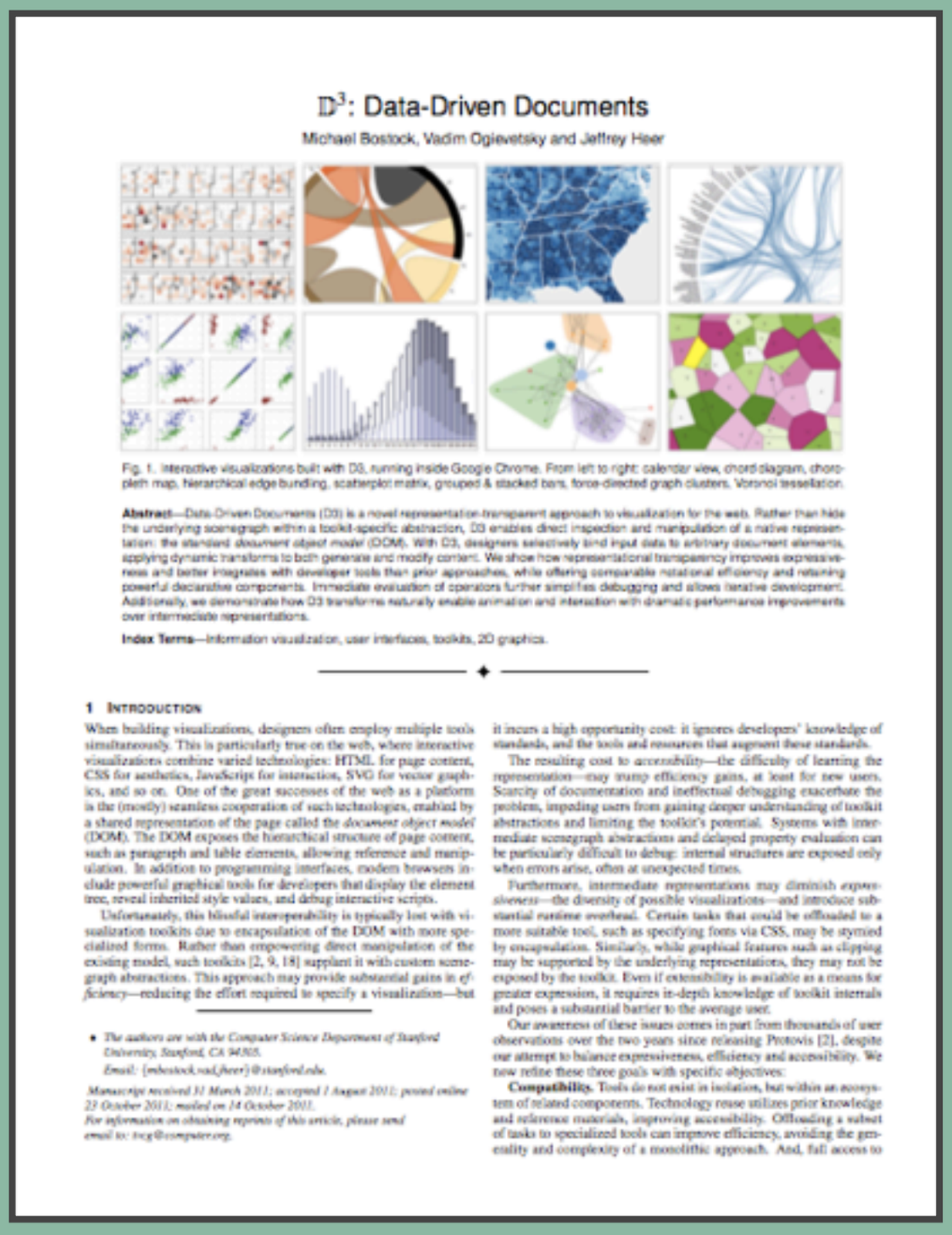


What about the

Internet?

D3: Data Driven Documents (2011)

Mike Bostock Jeffrey Heer Vadim Ogievetsky



D^3: Data-Driven Documents

Michael Bostock, Vadim Ogievetsky and Jeffrey Heer

Fig. 1. Interactive visualizations built with D3, running inside Google Chrome. From left to right: calendar view, chord diagram, choropleth map, hierarchical edge bundling, scatterplot matrix, grouped & stacked bars, force-directed graph clusters, Voronoi tessellation.

Abstract—Data-Driven Documents (D3) is a novel representation-transparent approach to visualization for the web. Rather than hide the underlying scenegraph within a toolkit-specific abstraction, D3 enables direct inspection and manipulation of a native representation: the standard *document object model* (DOM). With D3, designers selectively bind input data to arbitrary document elements, applying dynamic transforms to both generate and modify content. We show how representational transparency improves expressiveness and better integrates with developer tools than prior approaches, while offering comparable notational efficiency and retaining powerful declarative components. Immediate evaluation of operators further simplifies debugging and allows iterative development. Additionally, we demonstrate how D3 transforms naturally enable animation and interaction with dramatic performance improvements over intermediate representations.

Index Terms—Information visualization, user interfaces, toolkits, 2D graphics.

1 Introduction

When building visualizations, designers often employ multiple tools simultaneously. This is particularly true on the web, where interactive visualizations combine varied technologies: HTML for page content, CSS for aesthetics, JavaScript for interaction, SVG for vector graphics, and so on. One of the great successes of the web as a platform is the (mostly) seamless cooperation of such technologies, enabled by a shared representation of the page called the *document object model* (DOM). The DOM exposes the hierarchical structure of page content, such as paragraph and table elements, allowing reference and manipulation. In addition to programming interfaces, modern browsers include powerful graphical tools for developers that display the element tree, reveal inherited style values, and debug interactive scripts.

Unfortunately, this blissful interoperability is typically lost with visualization toolkits due to encapsulation of the DOM with more specialized forms. Rather than empowering direct manipulation of the existing model, such toolkits [2, 9, 18] supplant it with custom scenegraph abstractions. This approach may provide substantial gains in *efficiency*—reducing the effort required to specify a visualization—but it incurs a high opportunity cost: it ignores developers' knowledge of standards, and the tools and resources that augment these standards.

The resulting cost to *accessibility*—the difficulty of learning the representation—may trump efficiency gains, at least for new users. Scarcity of documentation and ineffectual debugging exacerbate the problem, impeding users from gaining deeper understanding of toolkit abstractions and limiting the toolkit's potential. Systems with intermediate scenegraph abstractions and delayed property evaluation can be particularly difficult to debug: internal structures are exposed only when errors arise, often at unexpected times.

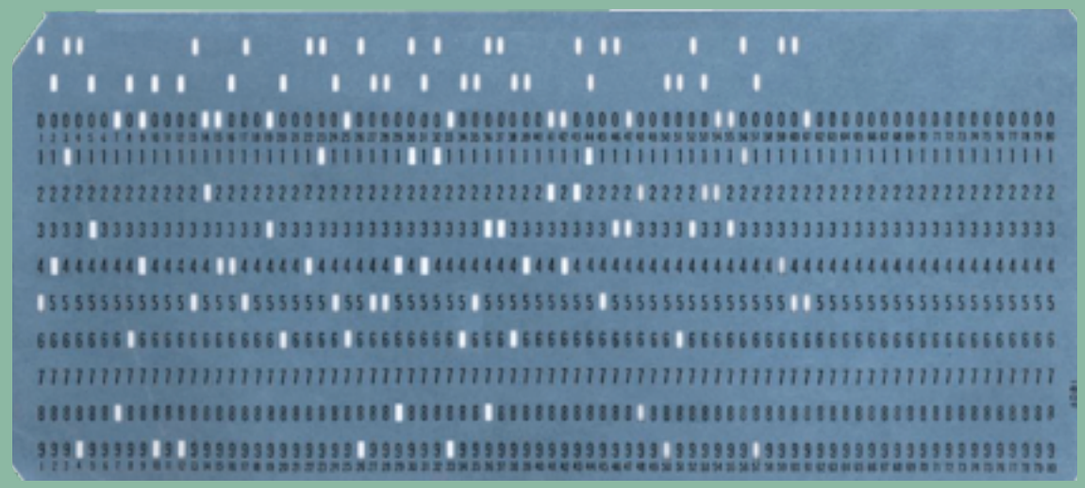
Furthermore, intermediate representations may diminish *expressiveness*—the diversity of possible visualizations—and introduce substantial runtime overhead. Certain tasks that could be offloaded to a more suitable tool, such as specifying fonts via CSS, may be stymied by encapsulation. Similarly, while graphical features such as clipping may be supported by the underlying representations, they may not be exposed by the toolkit. Even if extensibility is available as a means for greater expression, it requires in-depth knowledge of toolkit internals and poses a substantial barrier to the average user.

Our awareness of these issues comes in part from thousands of user observations over the two years since releasing Protovis [2], despite our attempt to balance expressiveness, efficiency and accessibility. We now refine these three goals with specific objectives:

Compatibility. Tools do not exist in isolation, but within an ecosystem of related components. Technology reuse utilizes prior knowledge and reference materials, improving accessibility. Offloading a subset of tasks to specialized tools can improve efficiency, avoiding the generality and complexity of a monolithic approach. And, full access to

• *The authors are with the Computer Science Department of Stanford University, Stanford, CA 94305.*
Email: {mbostock,vadi,jheer}@stanford.edu.

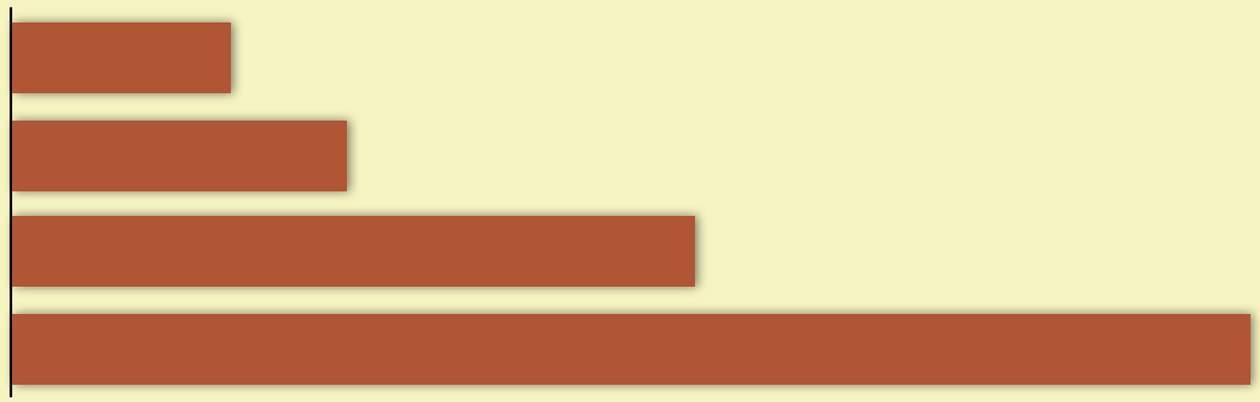
Manuscript received 31 March 2011; accepted 1 August 2011; posted online 23 October 2011; mailed on 14 October 2011.
For information on obtaining reprints of this article, please send email to: tvcg@computer.org.



D3 (JavaScript)

```
d3.select("body").selectAll("div")
  .data([17, 26, 53, 96])
  .enter().append("div")
  .style("width", function(d){d+"px";});
```

DOM



Awesome

- Declarative
  - Easy to think, explore
  - Optimizable
- Familiar representation
  - HTML, CSS, SVG; dev tools
  - No reinvented wheels

Awesomer

Clojure(Script)

Rich data structures

Deliberate state/mutation

Awesomer C2

http://keminglabs.com/c2

http://github.com/lynaghk/c2

Example.

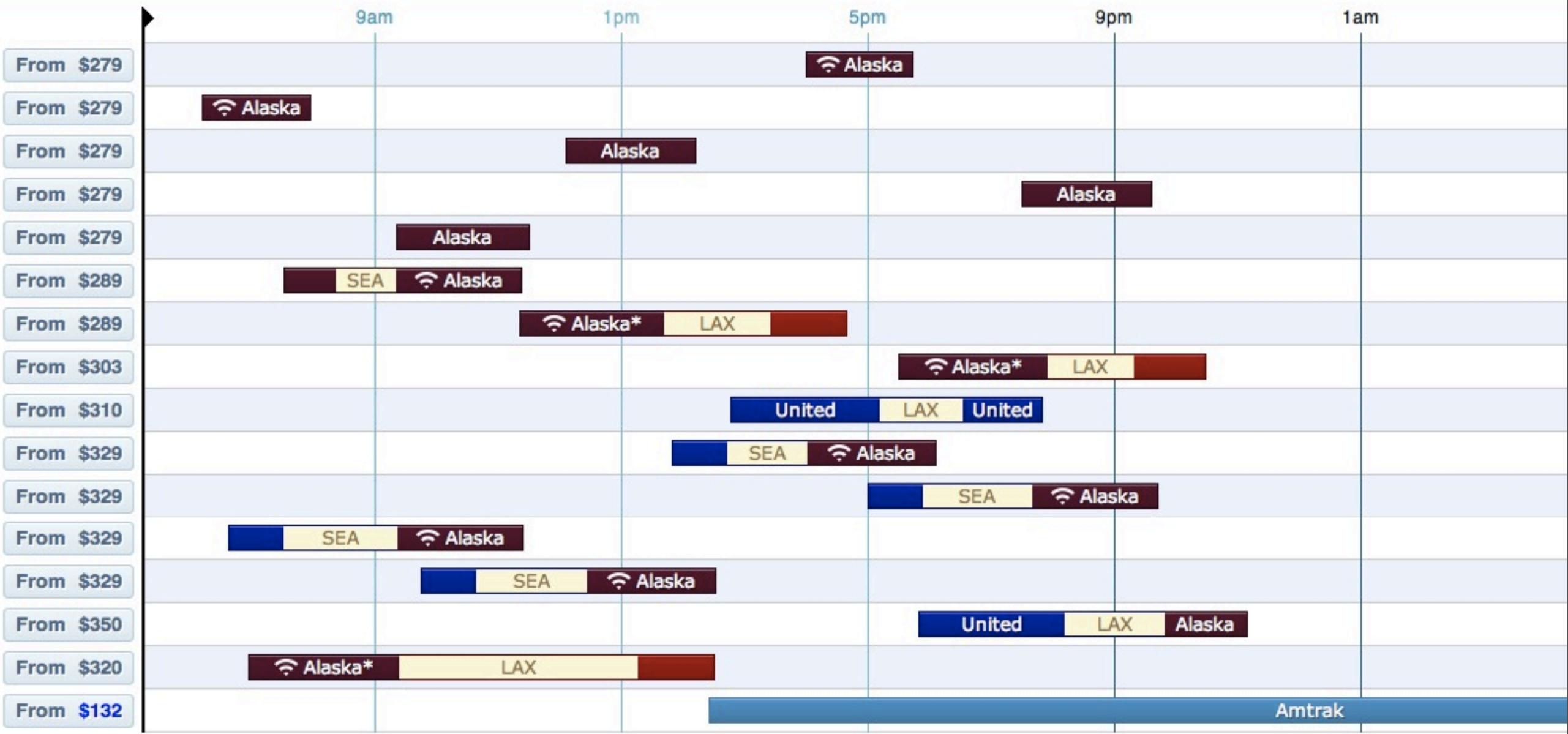
```
[{:flight-no 2, :price 106, :carrier "Alaska"
  :depart 16.91, :arrive 21.42}

 {:flight-no 1, :price 190, :carrier "United"
  :depart 6.20, :arrive 10.87}

 {:flight-no 5, :price 213, :carrier "United"
  :depart 4.73, :arrive 9.48}

{:flight-no 4, :price 221, :carrier "Alaska"
 :depart 18.39, :arrive 21.67}

                              ... ]
```

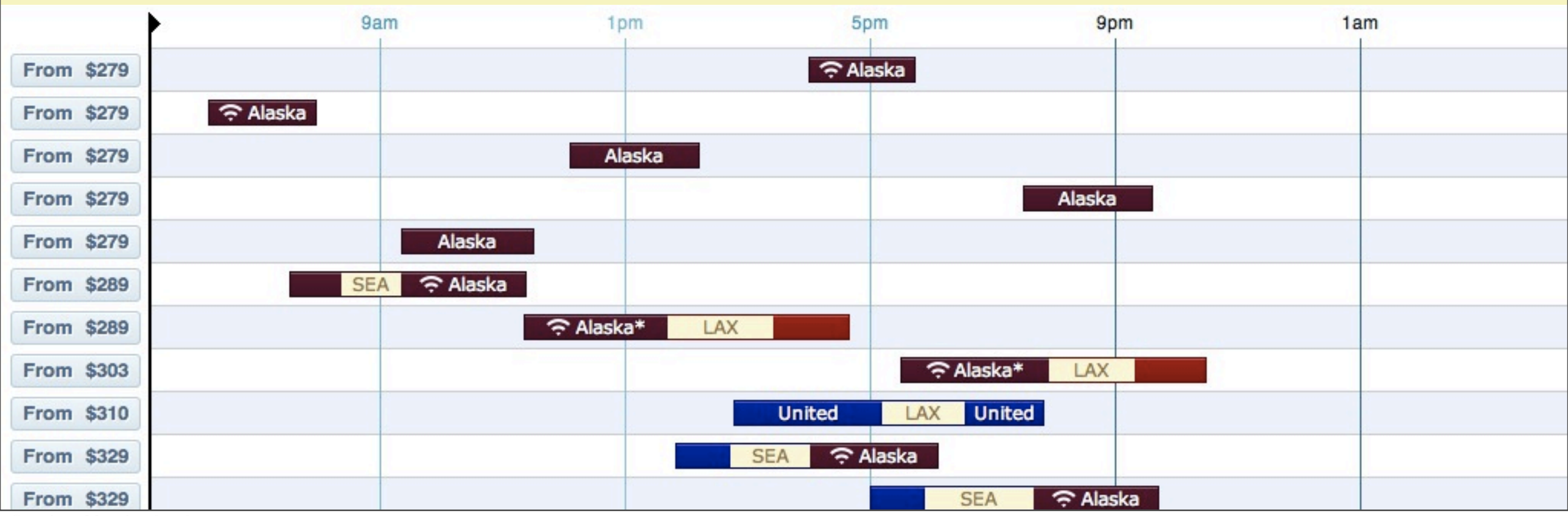
hipmunk

PDX ⟷ SJC, Mar 15 ⟷ Mar 18 | Search for hotels in San Jose, CA | New Search

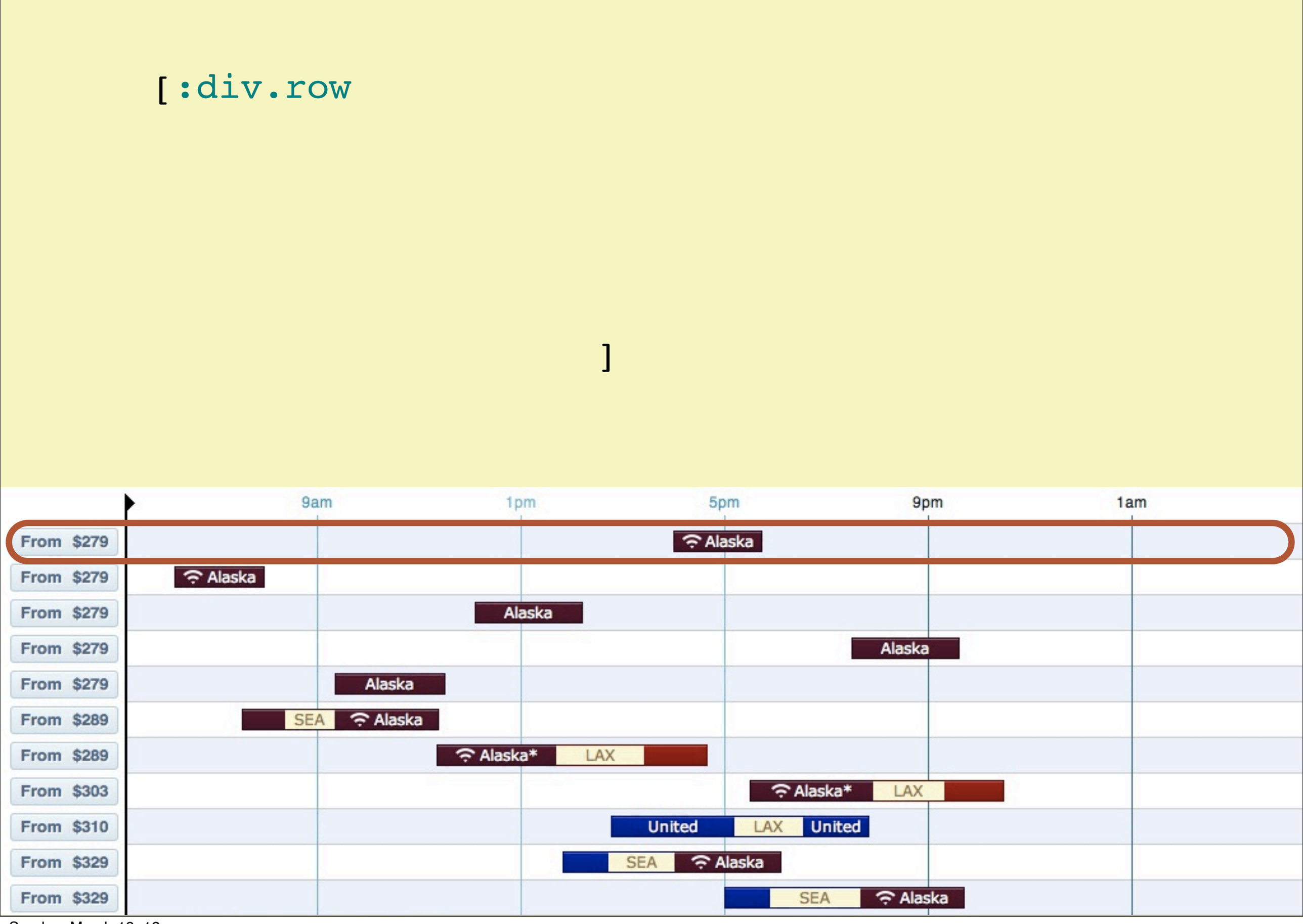
1 PDX → SJC on Thu, Mar 15 | 2 SJC → PDX on Sun, Mar 18

Sort By: Agony | Price | Duration | Departure | Arrival | Airlines | Non-stops

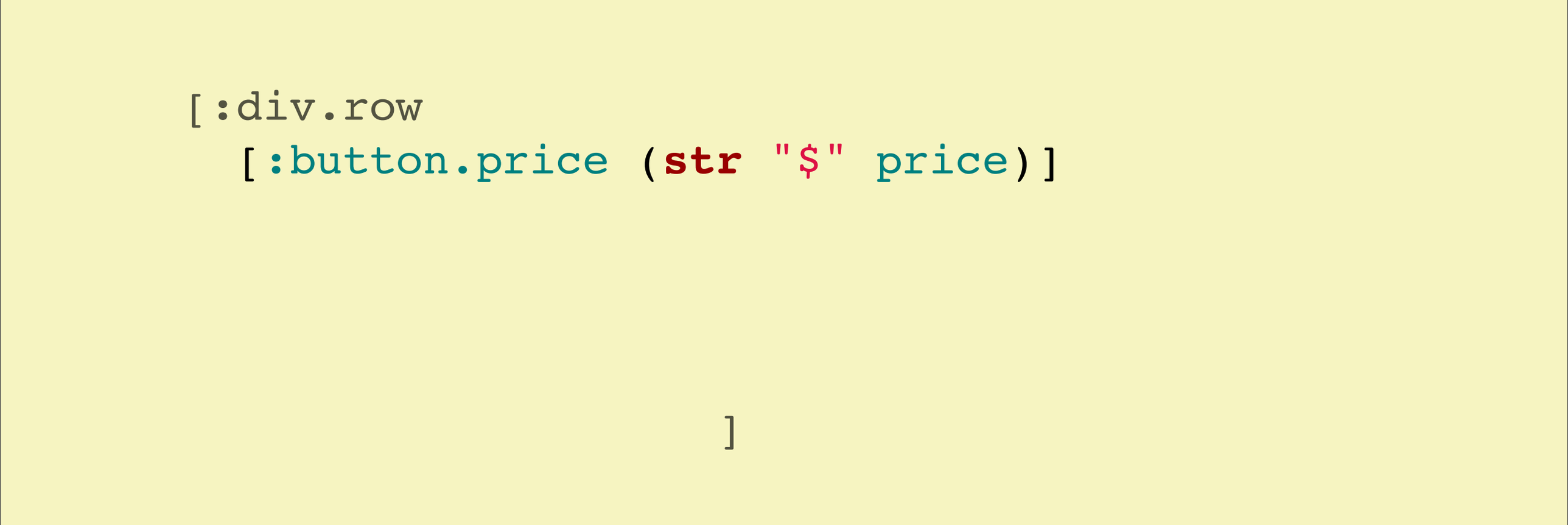



9am
1pm
5pm
9pm
1am
From $279
From $279
From $279
From $279
From $279
From $289
From $289
From $303
From $310
From $329
From $329
Alaska
Alaska
Alaska
Alaska
Alaska
SEA
Alaska
Alaska*
LAX
Alaska*
LAX
United
LAX
United
SEA
Alaska
SEA
Alaska

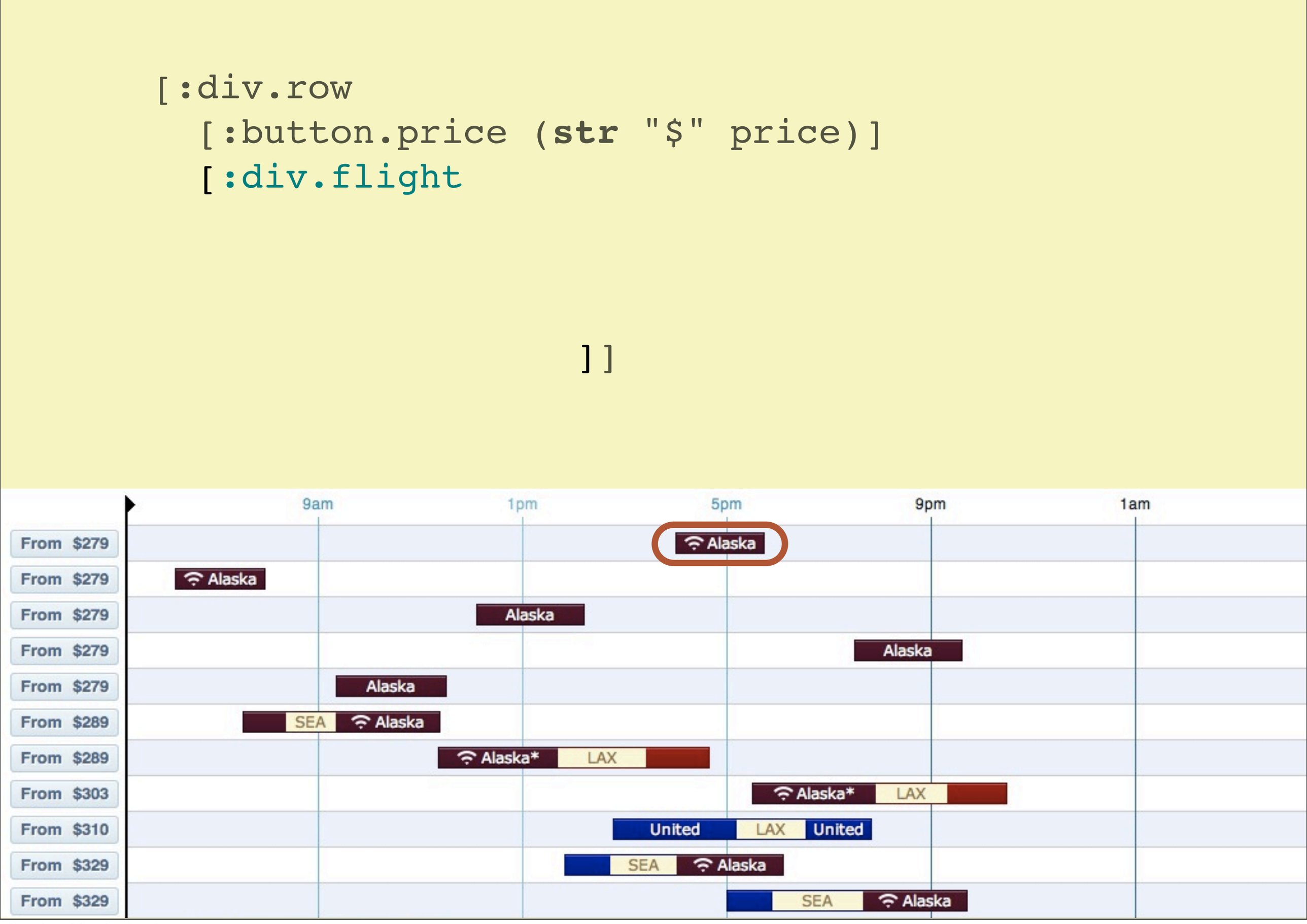
```
[:div.row

                                ]
```


9am
1pm
5pm
9pm
1am
From $279
Alaska
From $279
Alaska
From $279
Alaska
From $279
Alaska
From $279
Alaska
From $289
SEA
Alaska
From $289
Alaska*
LAX
From $303
Alaska*
LAX
From $310
United
LAX
United
From $329
SEA
Alaska
From $329
SEA
Alaska

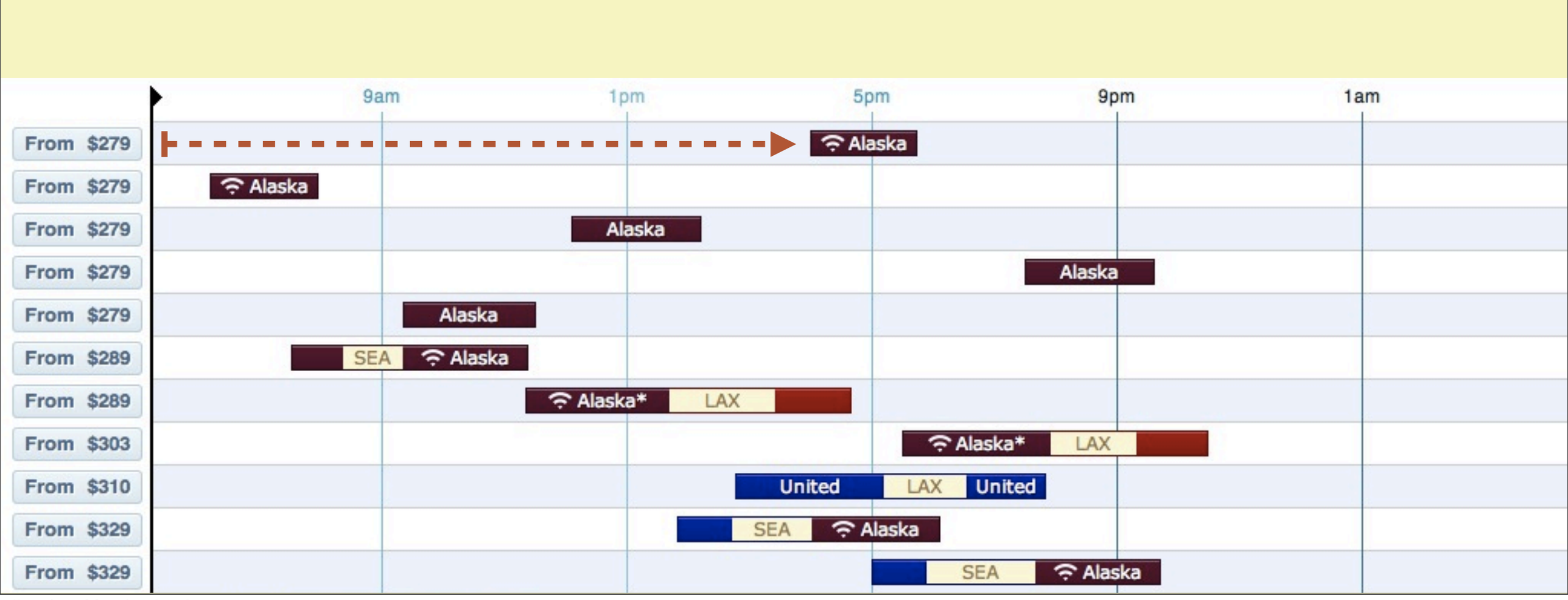
```
[:div.row
  [:button.price (str "$" price)]

                                ]
```



```
[:div.row
  [:button.price (str "$" price)]
  [:div.flight

                          ]]
```

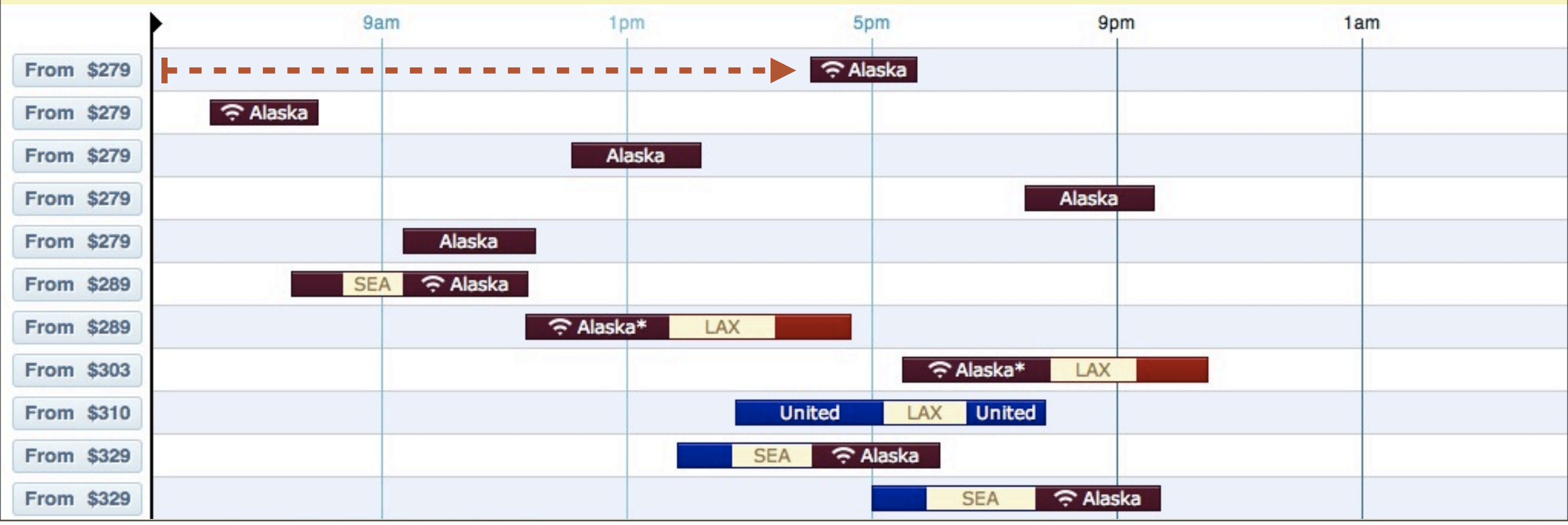


```
[:div.row
  [:button.price (str "$" price)]
  [:div.flight
    {:style {:left  (time-scale depart)
                                        }
                      }
                    ]]
```

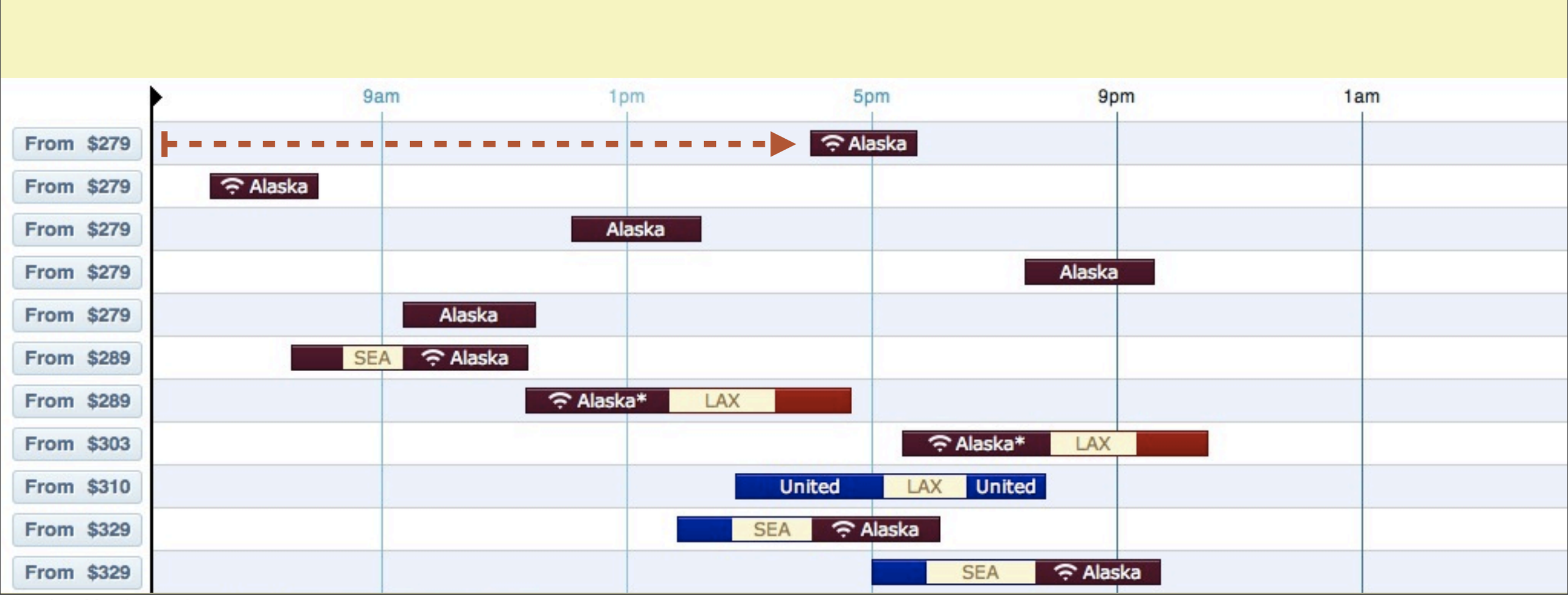


```
(let [time-scale (scale/linear :domain [0 24]
                               :range :percent)]

  (time-scale 0) ;=> “0%”
  (time-scale 12) ;=> “50%”
  (time-scale 24) ;=> “100%”
)
```

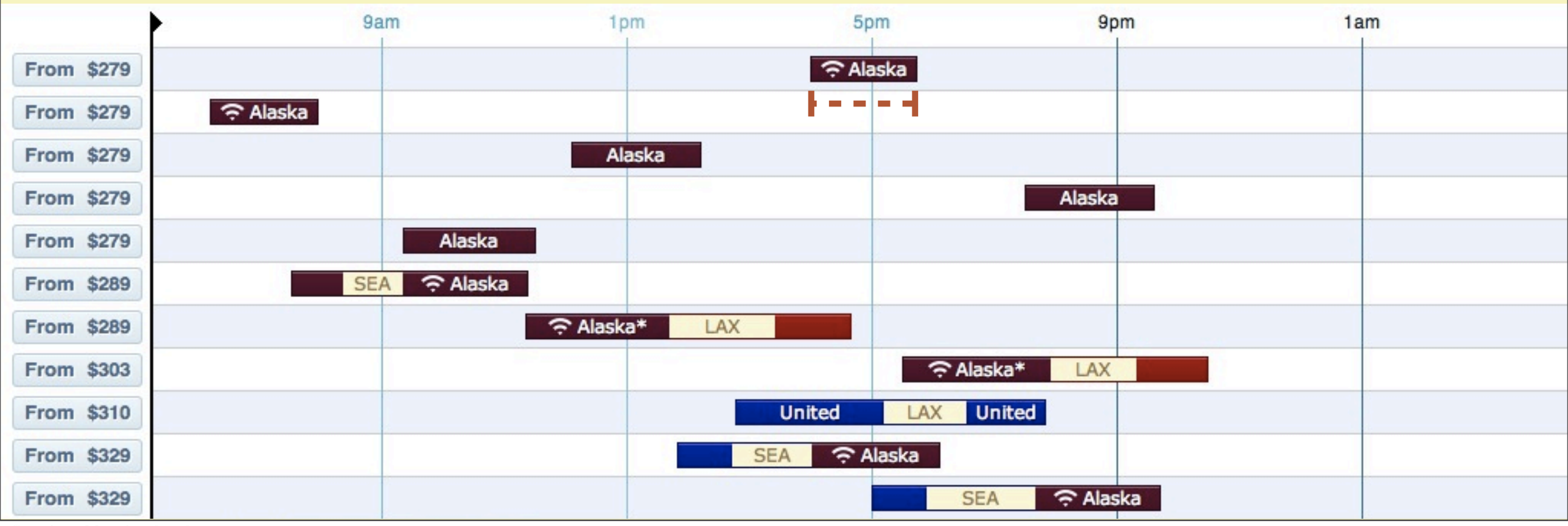
}



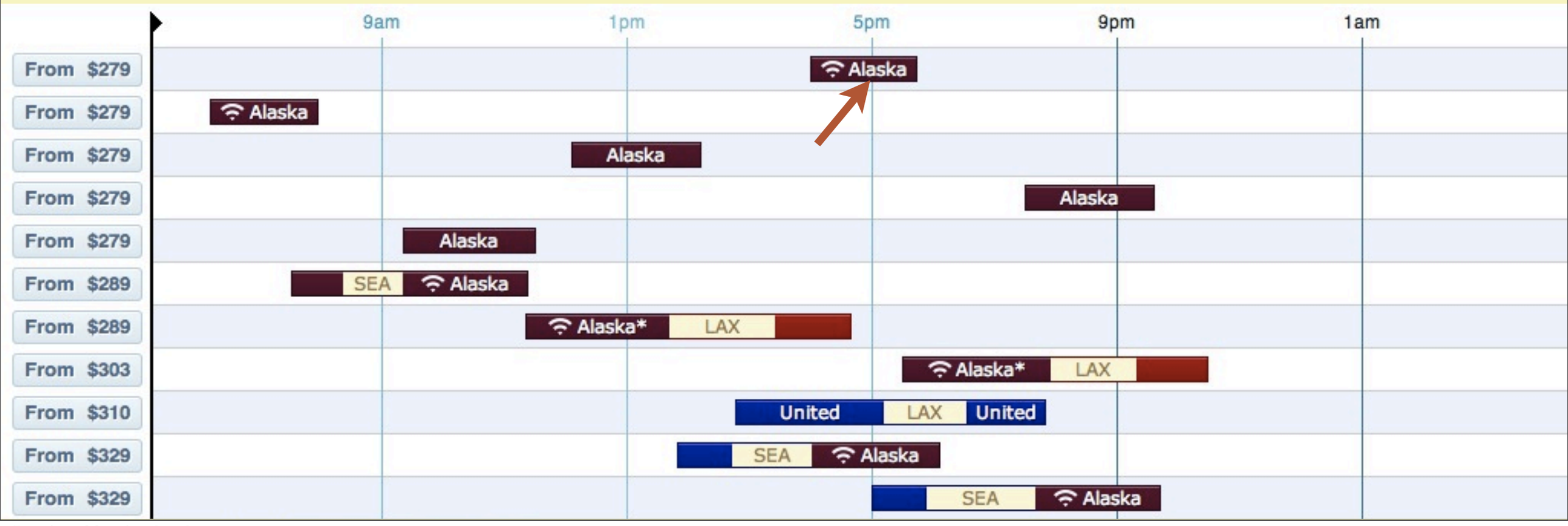
```
[:div.row
  [:button.price (str "$" price)]
  [:div.flight
    {:style {:left  (time-scale depart)
                                        }
                      }
                    ]]
```



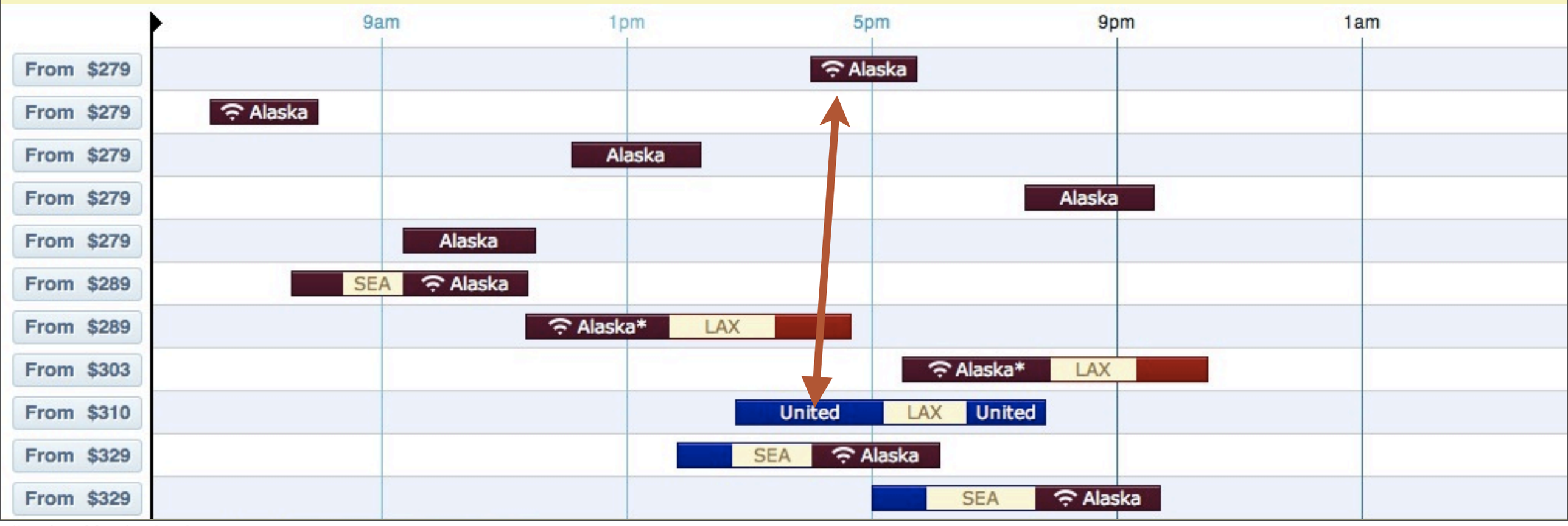
```
[:div.row
  [:button.price (str "$" price)]
  [:div.flight
    {:style {:left  (time-scale depart)
             :width (time-scale (- arrive depart))}
                     }
                   ]]
```



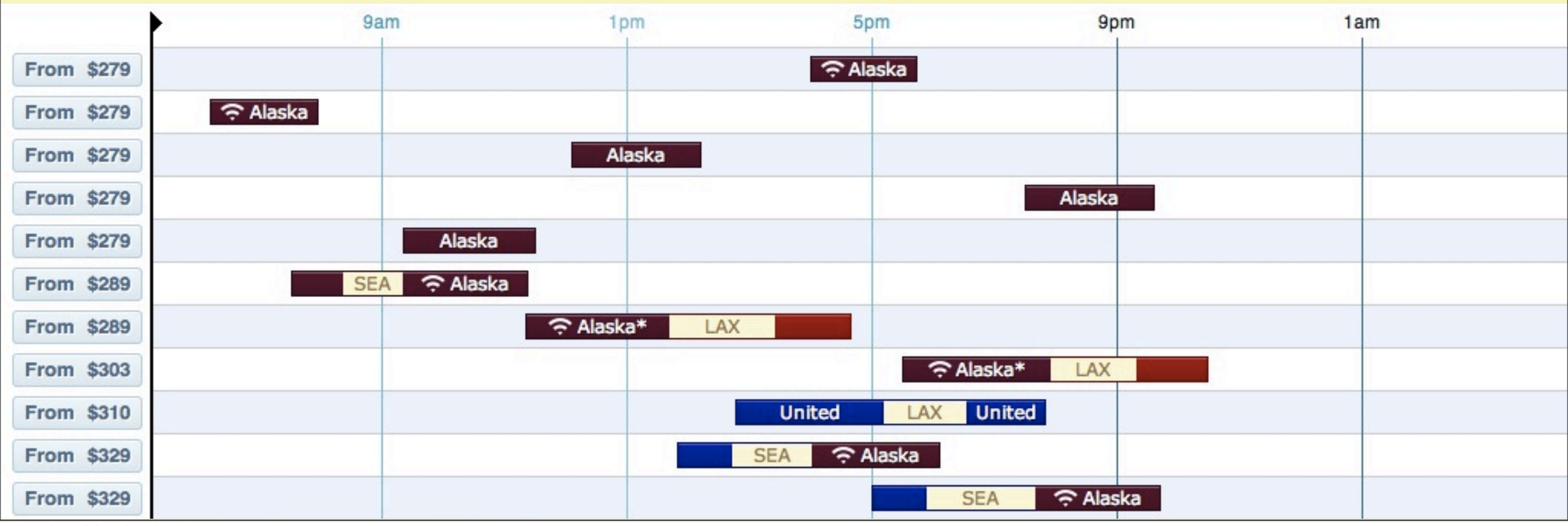
```
[:div.row
  [:button.price (str "$" price)]
  [:div.flight
    {:style {:left  (time-scale depart)
             :width (time-scale (- arrive depart))}
                     }
    [:span carrier]]]
```



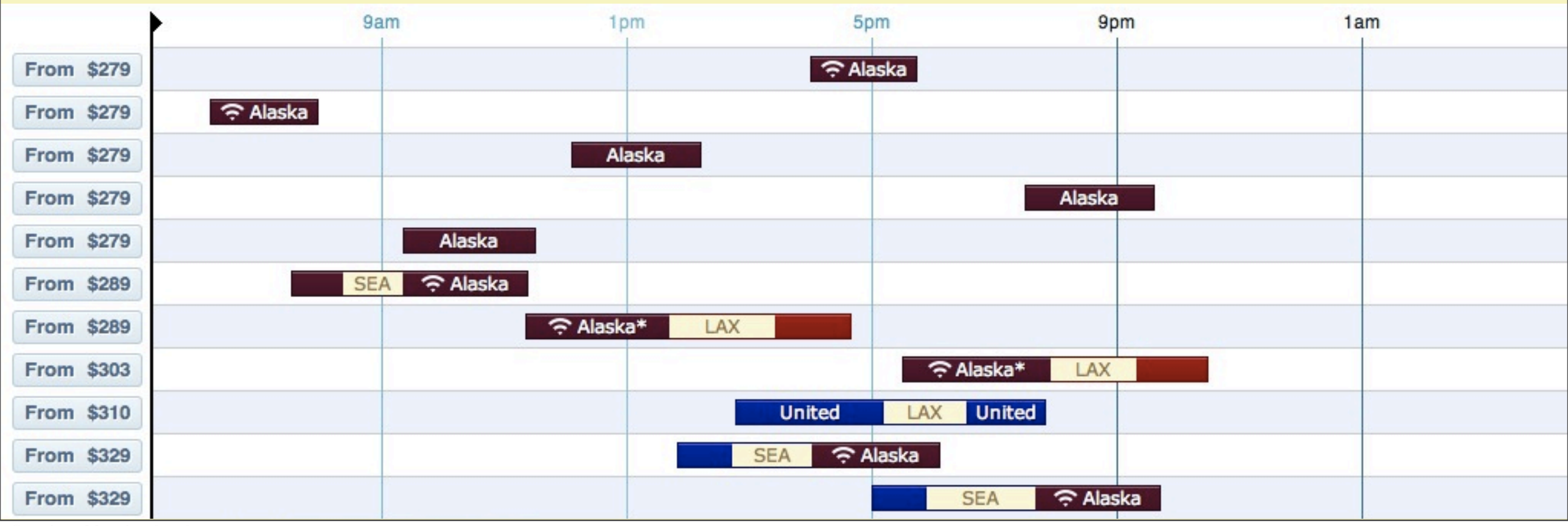
```
[:div.row
  [:button.price (str "$" price)]
  [:div.flight
    {:style {:left  (time-scale depart)
             :width (time-scale (- arrive depart))}
     :carrier carrier}
    [:span carrier]]]
```



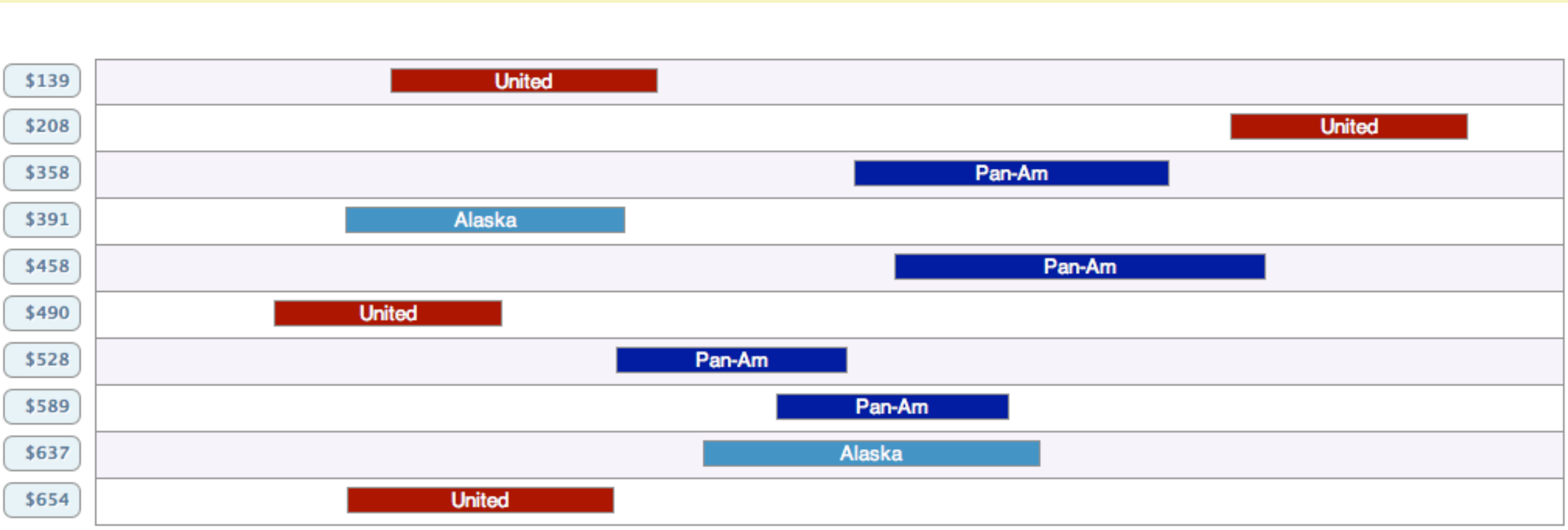
```clojure
(fn [{:keys [price carrier depart arrive]}]
    [:div.row
      [:button.price (str "$" price)]
      [:div.flight
        {:style {:left  (time-scale depart)
                 :width (time-scale (- arrive depart))}
         :carrier carrier}
        [:span carrier]]])
```



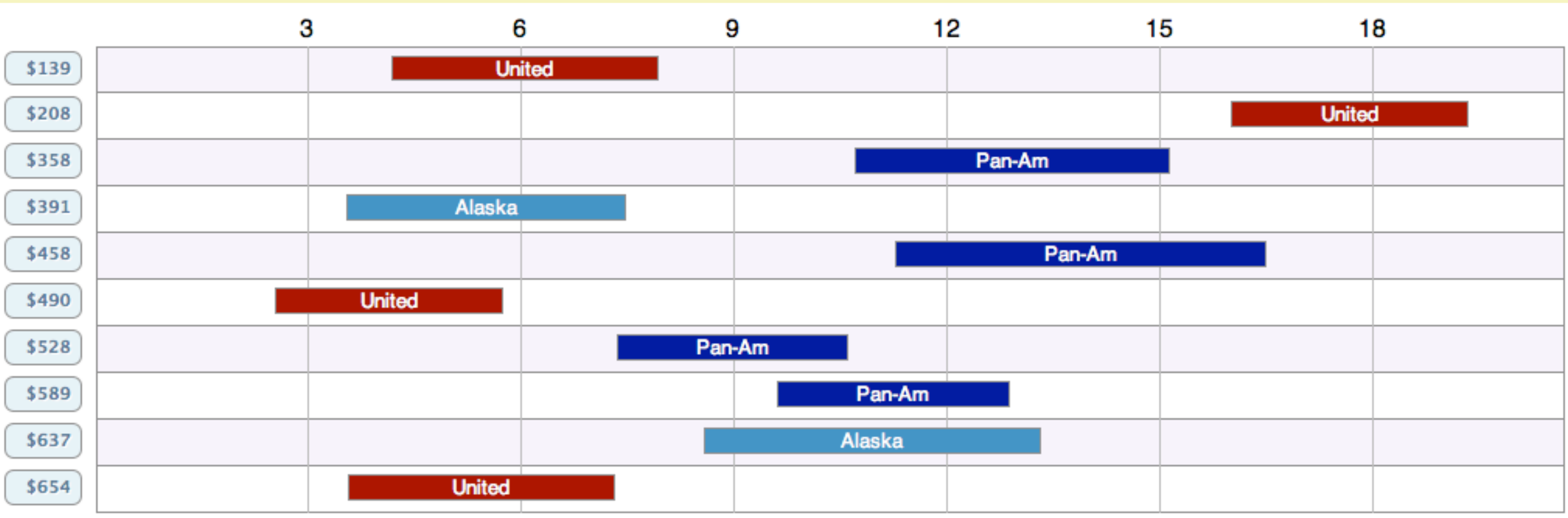
```
(unify! "#chart" flight-data
  (fn [{:keys [price carrier depart arrive]}]
      [:div.row
        [:button.price (str "$" price)]
        [:div.flight
          {:style {:left  (time-scale depart)
                   :width (time-scale (- arrive depart))}
           :carrier carrier}
          [:span carrier]]]))
```



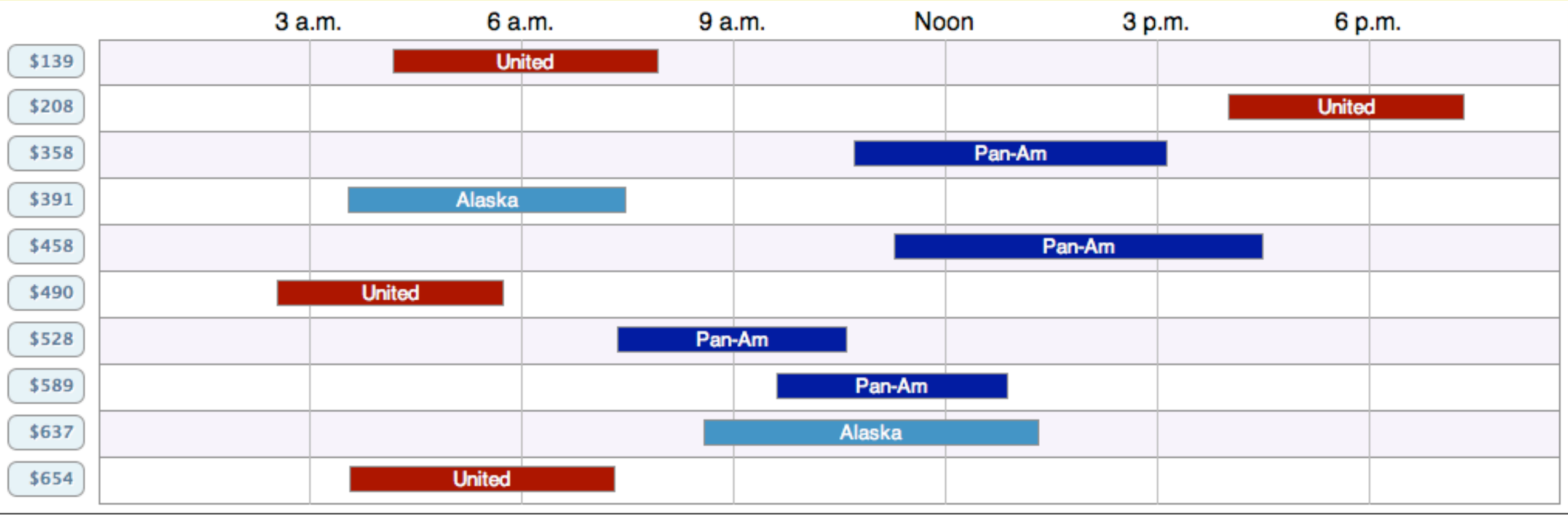
```
(unify! "#chart" flight-data
  (fn [{:keys [price carrier depart arrive]}]
      [:div.row
        [:button.price (str "$" price)]
        [:div.flight
          {:style {:left  (time-scale depart)
                   :width (time-scale (- arrive depart))}
           :carrier carrier}
          [:span carrier]]]))
```



```clojure
(unify! "#axis" [3 6 9 12 15 18]
          (fn [t]
             [:div.tick {:style {:left (time-scale t)}}
               [:div.grid-line]
               [:span.label (str t)]]))
```



```clojure
(unify! "#axis" [[3 "3 a.m."] [6 "6 a.m."] [9 "9 a.m."]
                 [12 "Noon"] [15 "3 p.m."] [18 "6 p.m."]]
        (fn [[t label]]
           [:div.tick {:style {:left (time-scale t)}}
             [:div.grid-line]
             [:span.label label]]))
```



Awesomer

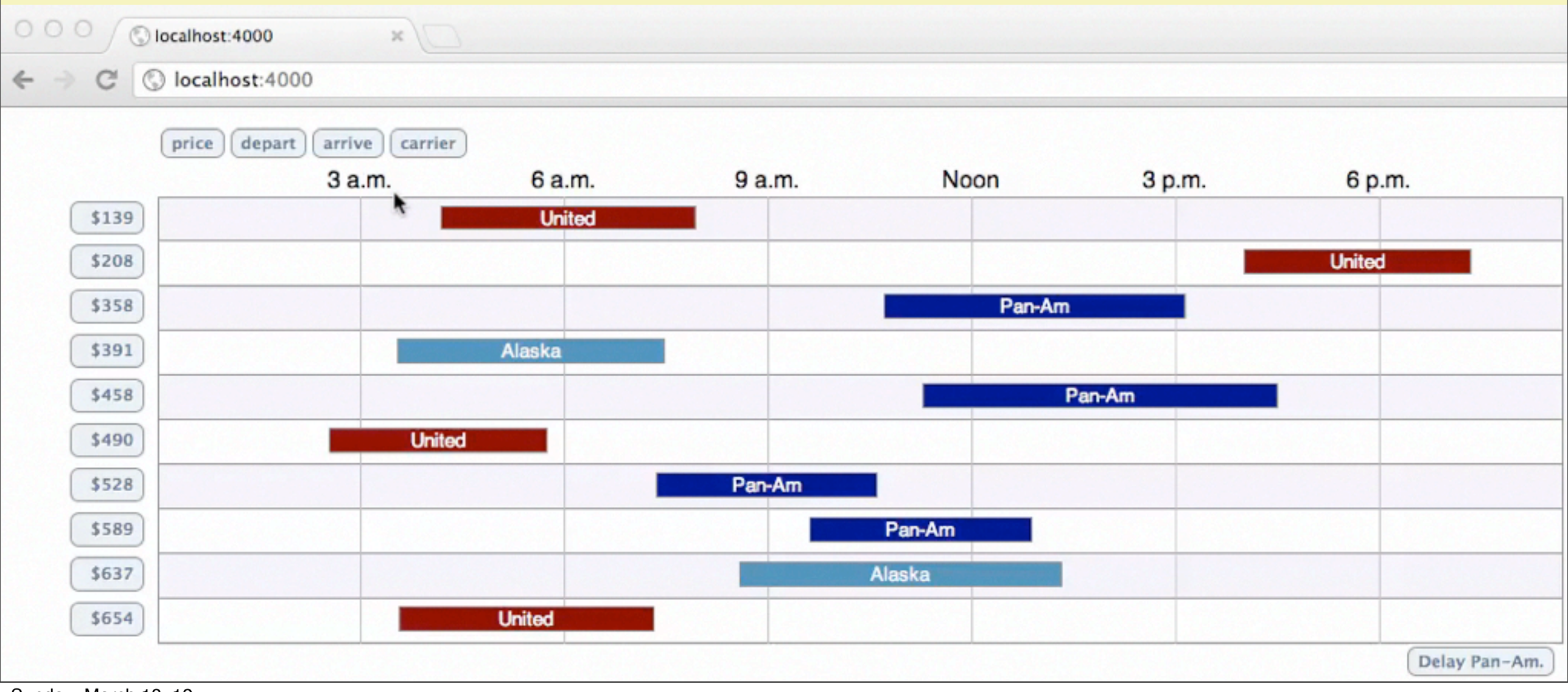
Clojure(Script)

Rich data structures

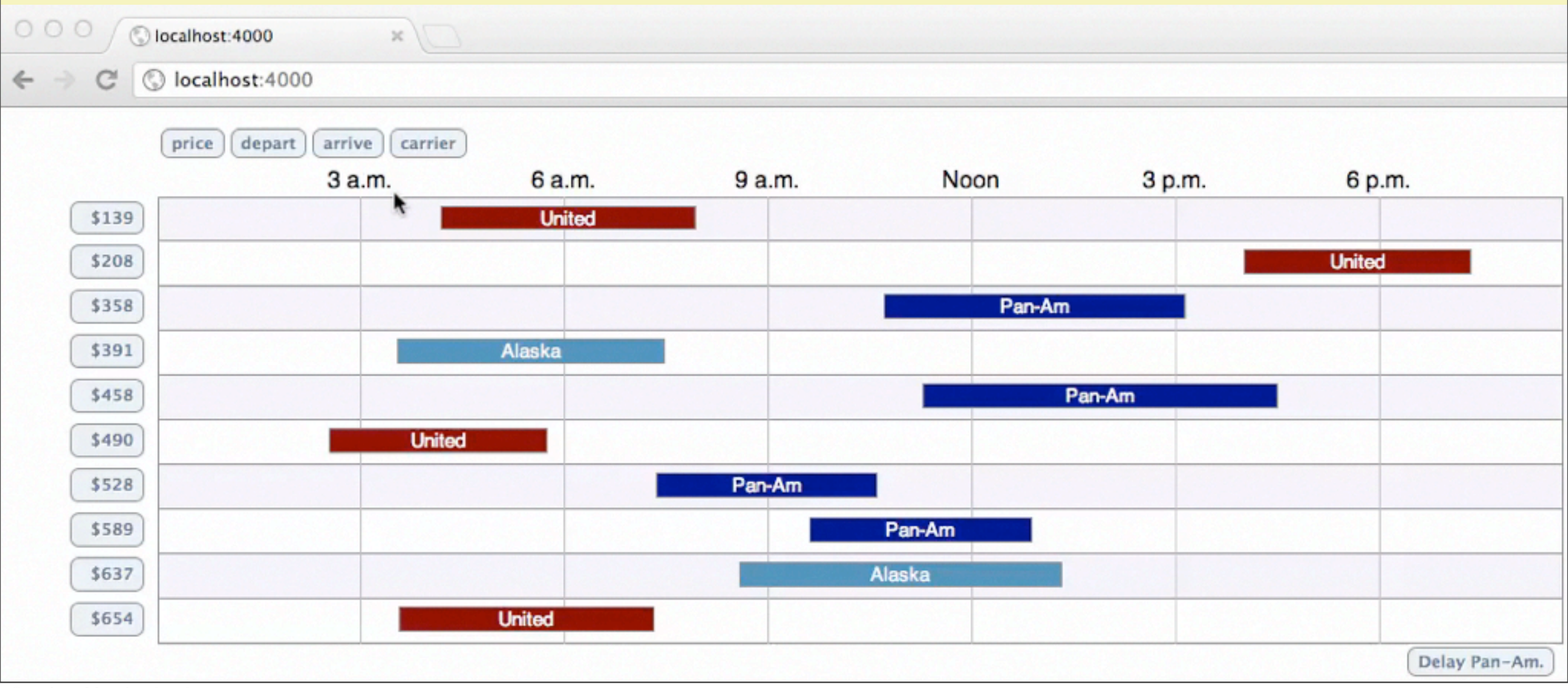
Deliberate state/mutation

```
(unify! "#buttons" [:price :depart :arrive :carrier]
        (fn [field idx] [:button (name field)]))

(event/on "#buttons" :click
  (fn [field] (swap! flight-data #(sort-by field %))))
```

```clojure
(unify! "#buttons" [:price :depart :arrive :carrier]
        (fn [field idx] [:button (name field)]))

(event/on "#buttons" :click
  (fn [field] (swap! flight-data #(sort-by field %))))
```



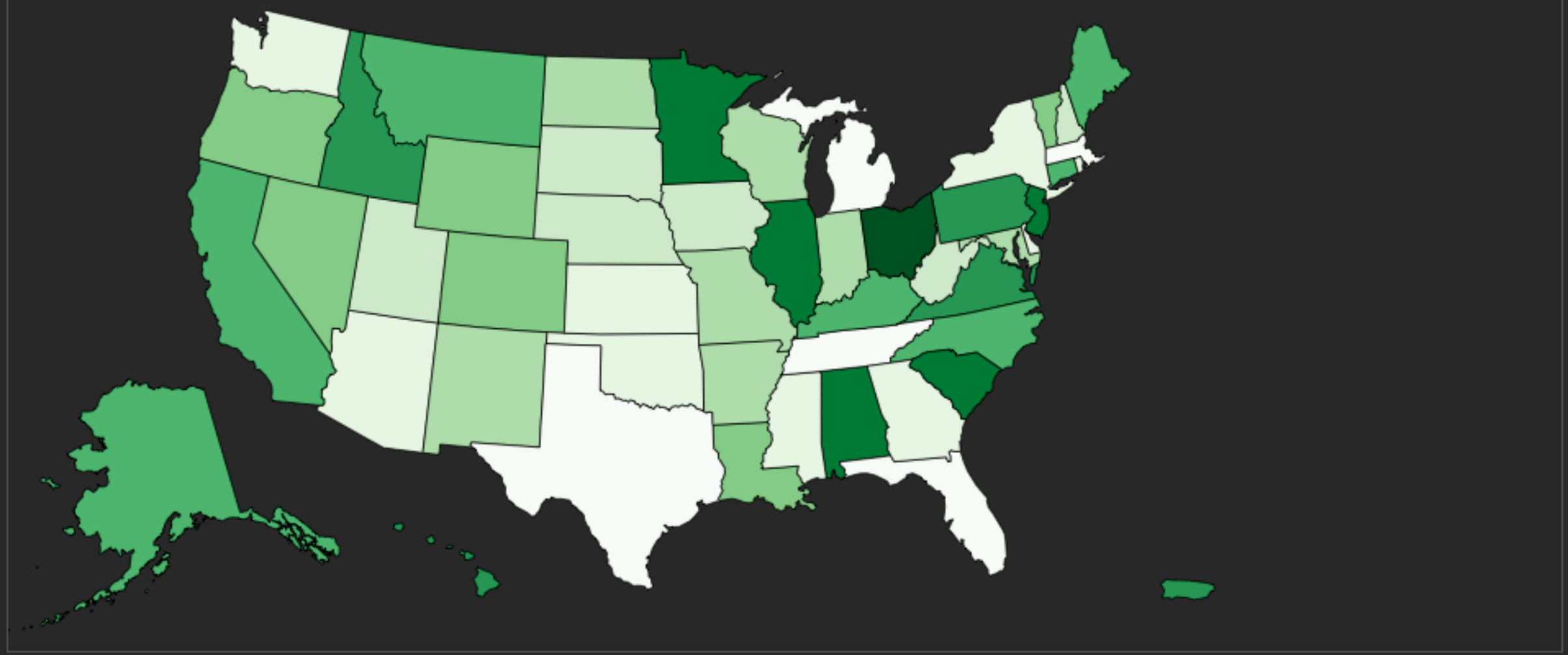
```
#flights .flight {
   z-index: 2;
   -webkit-transition: all 1s ease-out; }
```



Awesomer

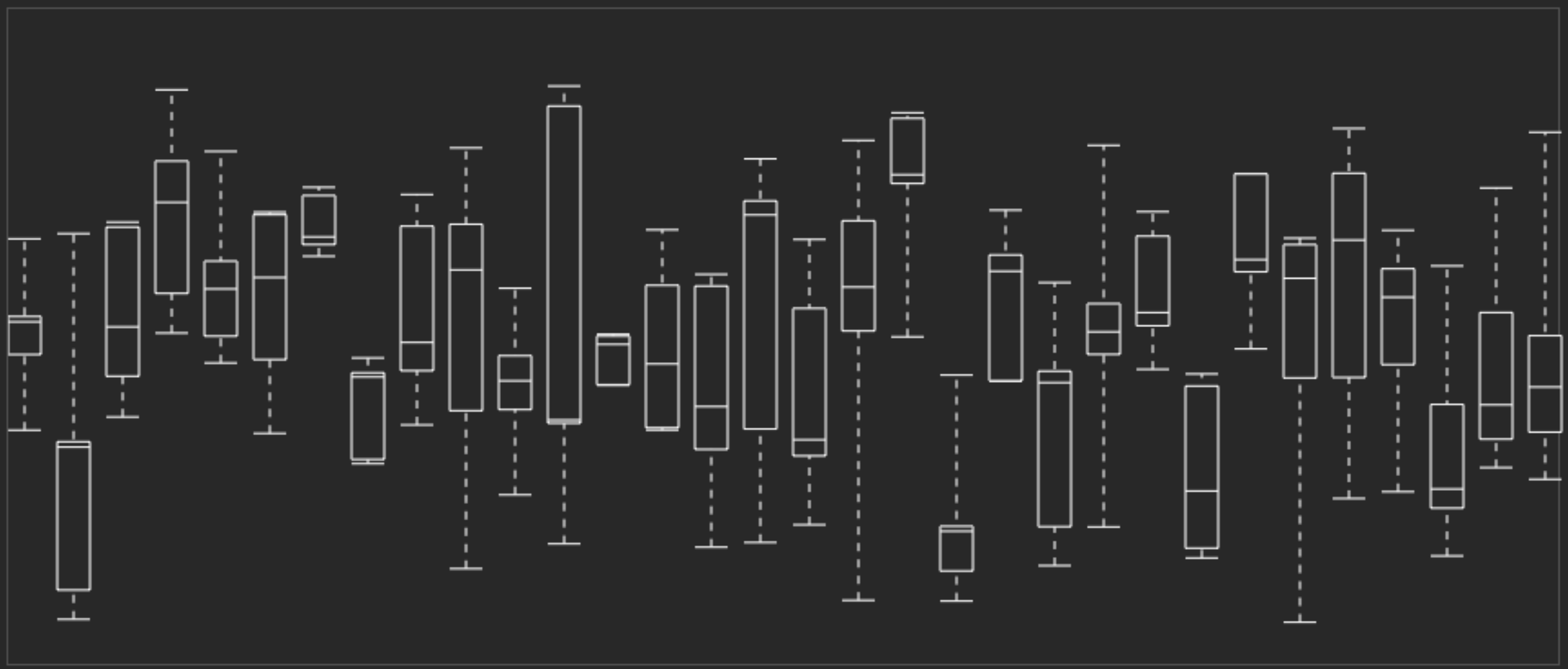
- Clojure(Script)
- ✓ Rich data structures
- ✓ Deliberate state/mutation

Choropleth



```
(ns choropleth
  (:use [c2.core :only [unify style]]
        [c2.maths :only [extent floor]]
        [c2.geo.core :only [geo->svg]]
        [c2.geo.projection :only [albers-usa]]
        [vomnibus.geo.us :only [states]])
  (:require [c2.scale :as scale]
            [vomnibus.color-brewer :as color-brewer]))
```

Boxplot

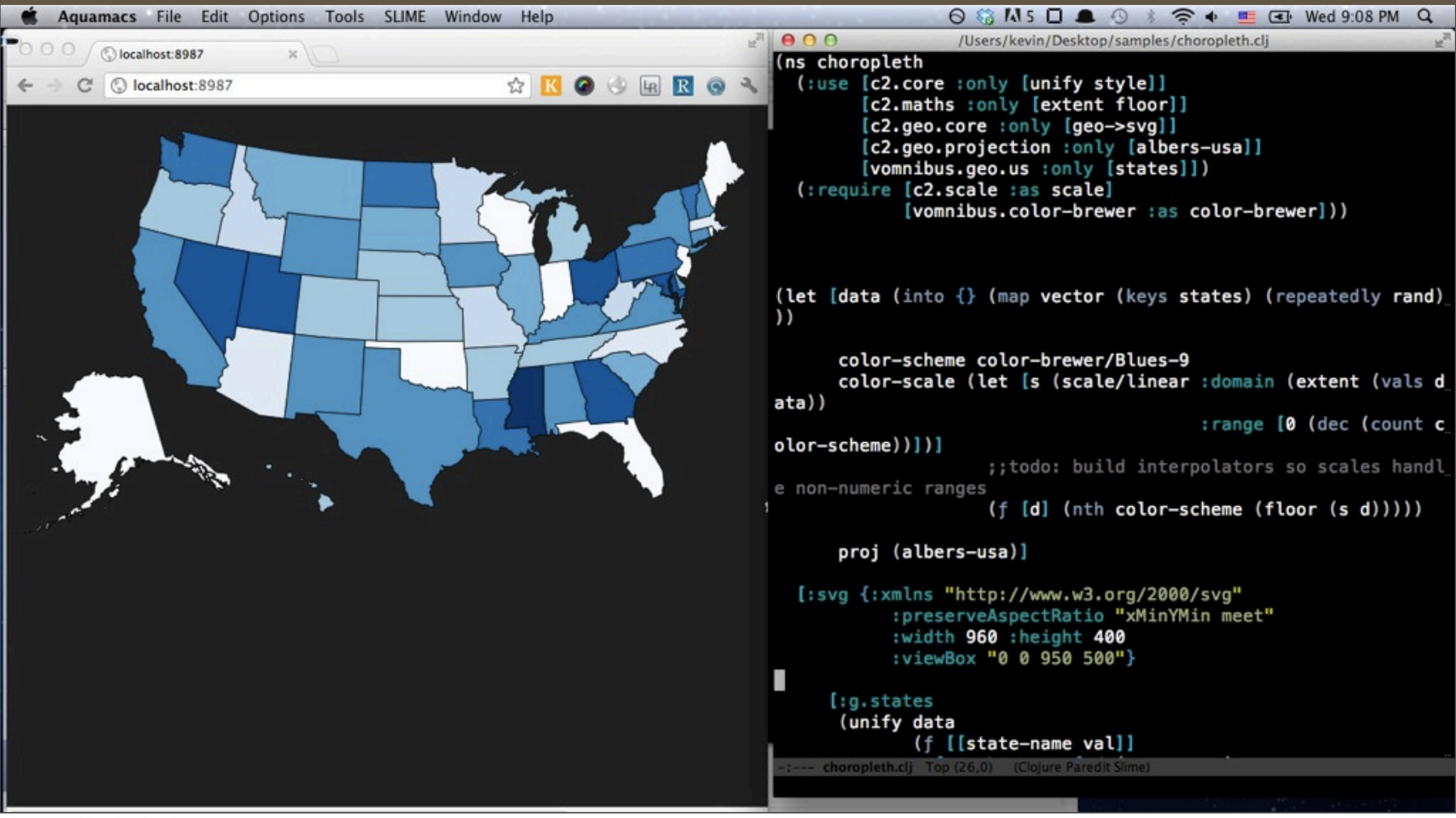


```
(ns boxplot
  (:use [c2.core :only [unify style]]
        [c2.maths :only [sin cos Tau extent]]
        [c2.util :only [half]]
        [c2.svg :only [translate]]
        [clojure.string :only [join]])
  (:require [c2.scale :as scale]))
```

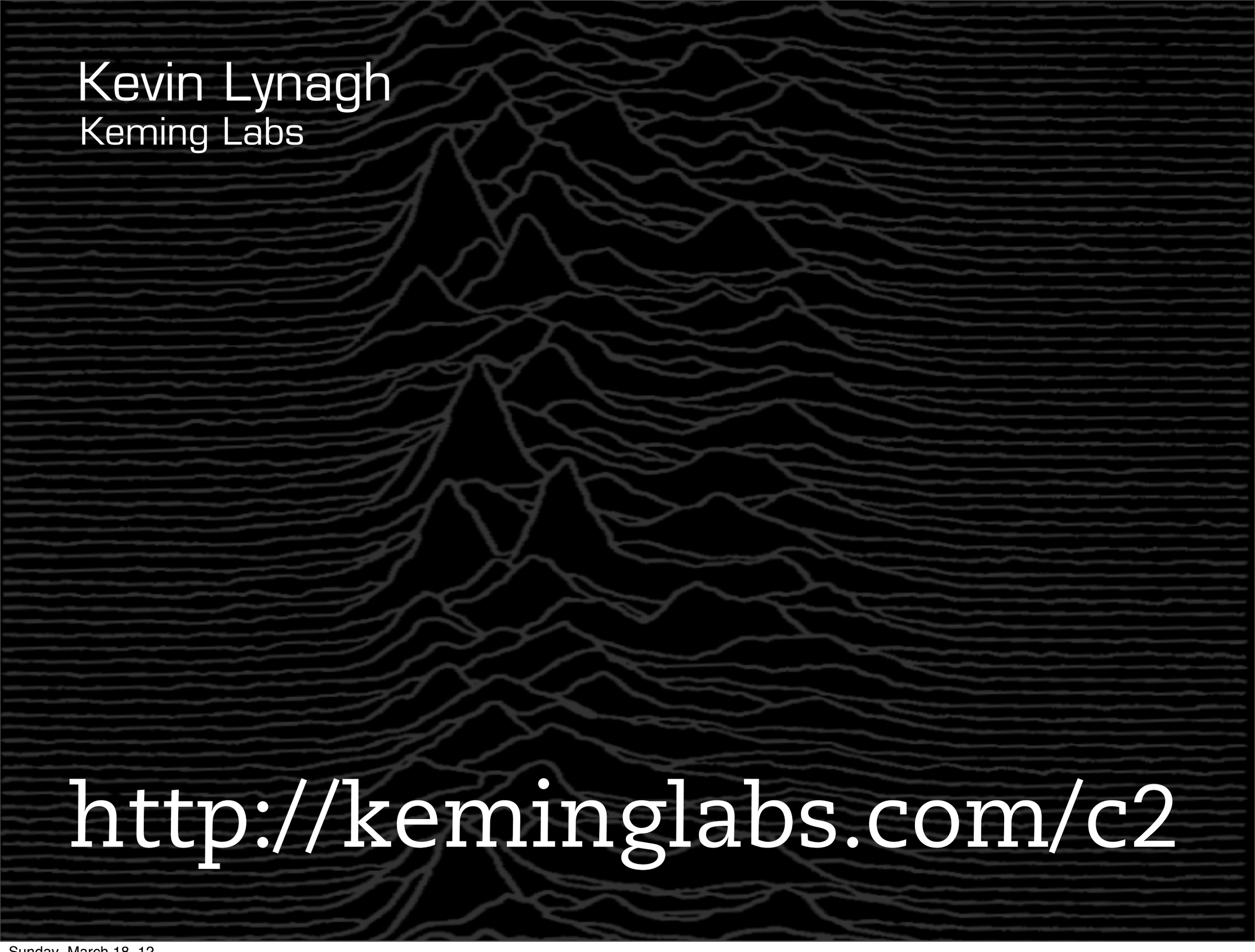
Bullet


```
(ns bullet
  (:use [c2.core :only [unify style]]
        [c2.util :only [half]])
  (:require [c2.scale :as scale]
            [vomnibus.color-brewer :as color-brewer]))


(def css "
.bullet { font: 10px sans-serif; }
.bullet .labels { fill: white; text-anchor: end; }
.bullet .marker { stroke: #000; stroke-width: 2px; }
.bullet .tick line { stroke: #666; stroke-width: .5px; }
.bullet .range.s0 { fill: #eee; }
.bullet .range.s1 { fill: #ddd; }
.bullet .range.s2 { fill: #ccc; }
```

Visual REPL


Aquamacs File Edit Options Tools SLIME Window Help
Wed 9:08 PM
localhost:8987
localhost:8987
/Users/kevin/Desktop/samples/choropleth.clj
(ns choropleth
(:use [c2.core :only [unify style]]
[c2.maths :only [extent floor]]
[c2.geo.core :only [geo->svg]]
[c2.geo.projection :only [albers-usa]]
[vomnibus.geo.us :only [states]])
(:require [c2.scale :as scale]
[vomnibus.color-brewer :as color-brewer]))
(let [data (into {} (map vector (keys states) (repeatedly rand)
))
color-scheme color-brewer/Blues-9
color-scale (let [s (scale/linear :domain (extent (vals d
ata))
:range [0 (dec (count c
olor-scheme))])]
;;todo: build interpolators so scales handl
e non-numeric ranges
(f [d] (nth color-scheme (floor (s d)))))
proj (albers-usa)]
[:svg {:xmlns "http://www.w3.org/2000/svg"
:preserveAspectRatio "xMinYMin meet"
:width 960 :height 400
:viewBox "0 0 950 500"}
[:g.states
(unify data
(f [[state-name val]]
-:--- choropleth.clj Top (26,0) (Clojure Paredit Slime)

fin.


Kevin Lynagh
Keming Labs
http://keminglabs.com/c2