Ben Welsh • Dec 18, 2018
Editor, @datadesk. Organizer, @california-civic-data-coalition. Archivist, @pastpages.

9

Listed in California fire zones, Maps, and Journalism

California buildings in severe fire hazard zones

The Los Angeles Times conducted an analysis of California buildings within fire hazard zones for the Dec. 18, 2018, story "A million California buildings face wildfire risk. 'Extraordinary steps' are needed to protect them." It found that at least 1.1 million structures, roughly 1 in 10 in the state, are within the highest risk zones.

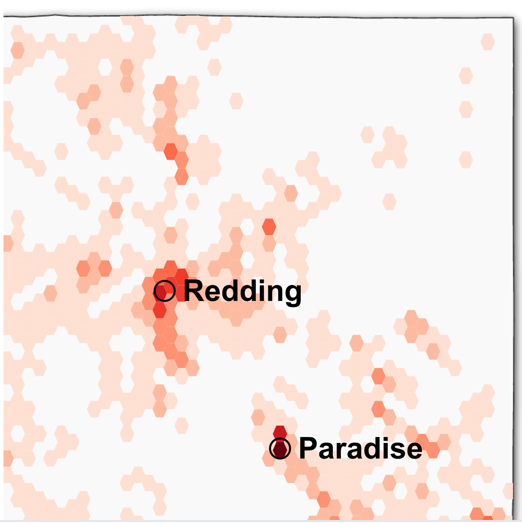
Here's one of several maps presented as part of the story.

Pick a hexagon size

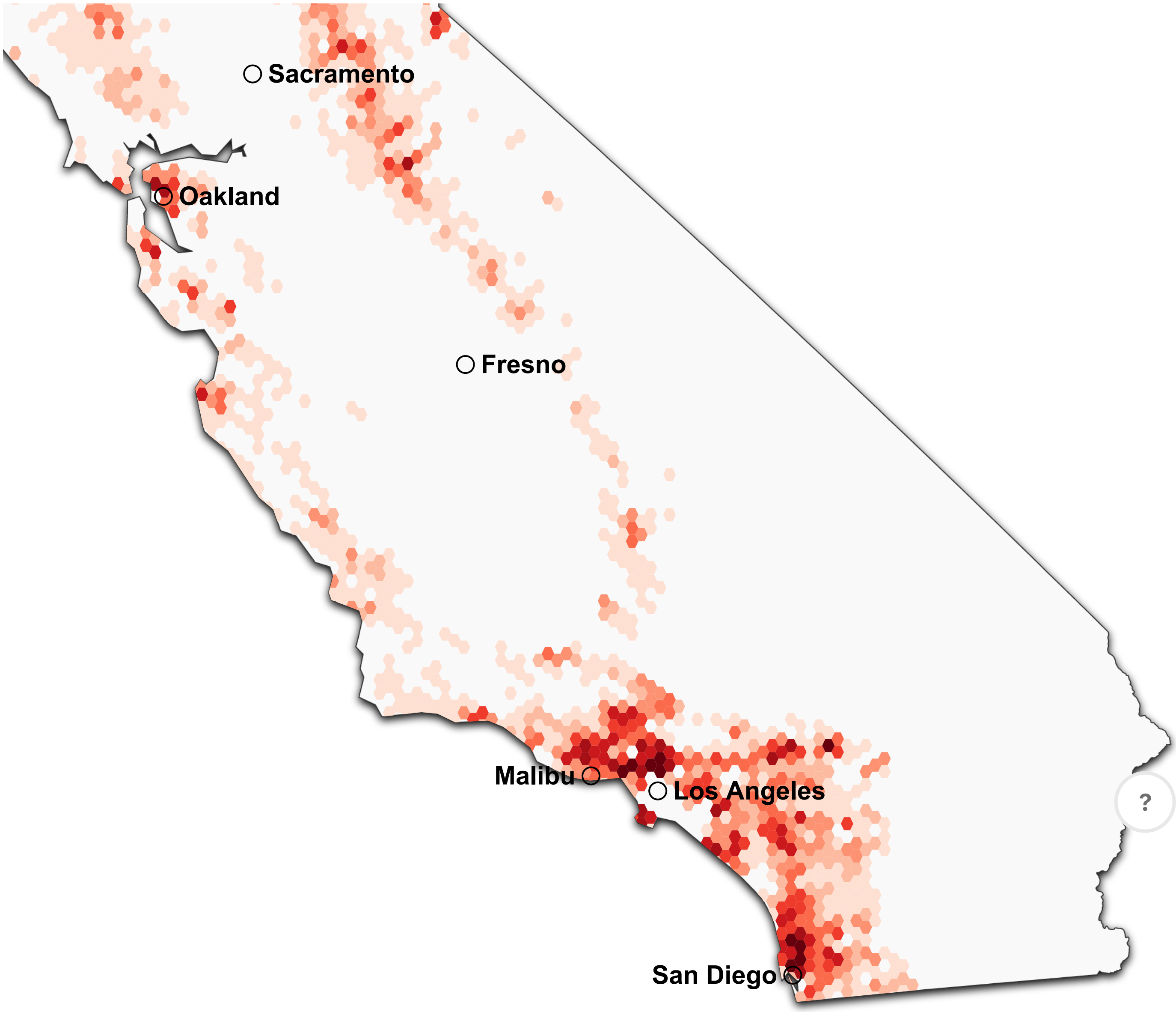
Big ▼

Pick a metric

buildings in a risk zone ▼



You have 1 unsaved change. Fork this notebook to save.



<style>

SVG exports

Download map

Download legend

Configuration

```
formatTick = ƒ(v)

formatTick = v =>  metric == 'in_pct' ? percent(v) : intcomma(roundHundred(v))
```

You have 1 unsaved change. Fork this notebook to save.

```
percent = v => (v*100).toFixed(1) + "%"
```

```
intcomma = ƒ(t)

intcomma = d3.format(",")
```

```
roundHundred = ƒ(v)

roundHundred = v => Math.round(v/100)*100
```

```
legendBreaks = ▸ Array(9) [Array(2), Array(2), Array(2), Array(2), Array(2), Array(2), Arr

legendBreaks = color.range().map(c => {
    var d = color.invertExtent(c);
    if (d[0] == null) d[0] = 1;
    return d;
})
```

```
legendAxis = ƒ(l)

legendAxis = d3.axisBottom(legendScale)
    .tickSize(13)
    .tickValues(color.domain())
    .ticks(breaks)
```

```
legendScale = ƒ(n)

legendScale = d3.scaleLinear()
    .domain(d3.extent(hexes.features.map(d => d.properties[metric])))
    .range([0, width*0.66]);
```

```
color = ƒ(t)

color = d3.scaleThreshold()
  .domain(ckMeansGroups.map(d => d3.min(d)))
  .range(d3.schemeReds[breaks])
```

```
ckMeansGroups = ▸ Array(9) [Array(1061), Array(256), Array(146), Array(78), Array(55), Arr

ckMeansGroups = ss.ckmeans(hexes.features.map(d => d.properties[metric]), breaks)
```

```
breaks = 9
```

```
path = ƒ(t)
```

You have 1 unsaved change. Fork this notebook to save.

```
projection = ƒ(t)
```

```
projection = d3.geoMercator()
  .center([-118.75, 37.5])
  .scale((1 << 19) / (40 * Math.PI))
  .translate([width / 2, height / 2])
```

```
height = 900
```

```
height = 900
```

Data

```
cities = ▸ Array(8) [Object, Object, Object, Object, Object, Object, Object, Object]
```

```
cities = [
  {
    name: 'Los Angeles',
    coordinates: [-118.2437, 33.95],
  },
  {
    name: "Paradise",
    coordinates: [-121.60, 39.7596]
  },
  {
    name: "Oakland",
    coordinates: [-122.21, 37.81]
  },
  {
    name: "Redding",
    coordinates: [-122.3917, 40.5865]
  },
  {
    name: "Sacramento",
    coordinates: [-121.4944, 38.5816]
  },
  {
    name: "Malibu",
    coordinates: [-118.7798, 34.05],
    x: "-9",
    anchor: "end"
  },
  {
    name: "San Diego",
```

You have 1 unsaved change. Fork this notebook to save.

```
    anchor: "end"
  },
  {
    name: "Fresno",
    coordinates: [-119.7871, 36.7378]
  }
]
```

```
state = ▸Object {type: "Feature", properties: Object, geometry: Object}
```

```
state = {
  const url =
"https://gist.githubusercontent.com/palewire/e001c971f2cab1664168658caa7536da/raw/state.j
son";
  const r = await d3.json(url);
  return topojson.feature(r, r.objects["state"]).features[0];
}
```

```
hexes = {
  const r = await d3.json(hexConfig.url);
  const layer = topojson.feature(r, r.objects[hexConfig.name]);
```

Fork

```
    features: layer.features.filter(d => d.properties[metric] >= minimum)
  };
}
```

```
hexConfig = {
  return {
    Big: {
      name: "big-hexes-with-analysis",
      url:
"https://gist.githubusercontent.com/palewire/a57662d364a131bc4d17ca436cc4e20b/raw/big-
hexes-with-analysis.json"
    },
    Small: {
      name: "small-hexes-with-analysis",
      url:
"https://gist.githubusercontent.com/palewire/37354699bc3855ccf5e1b48eff6fc146/raw/small-
hexes-with-analysis.json"
    }
```

You have 1 unsaved change. Fork this notebook to save.

```
}
```

```
minimum = {
  return {
    in_zone: 10,
    in_pct: 0.01,
    total: 1
  }[metric]
}
```

```
topojson = ▸ Object {bbox: ƒ(topology), feature: ƒ(topology, o), mesh: ƒ(topology), meshAr
```

```
d3 = ▸ Object {event: null, format: ƒ(t), formatPrefix: ƒ(t, n), timeFormat: ƒ(t), timePar
```

```
import {serialize} from @mbostock/saving-svg
```

```
import {select, slider} from @jashkenas/inputs
```

Continue reading Journalism

PREVIOUS

Regional maps of California buildin…

NEXT

The Morris Mistake